



## Харківський національний університет радіоелектроніки

Факультет інформаційних радіотехнологій та технічного захисту інформаціїКафедра медіаінженерії та інформаційних радіоелектронних системРівень вищої освіти другий (магістерський)Спеціальність 171 Електроніка

(код і повна назва)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма Системи, технології і комп'ютерні засоби мультимедіа

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Пархомову Антону Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Моделювання польотів БПЛА у зоні критично важливого об'єкта інфраструктури

затверджена наказом по університету від " 08 " \_\_\_\_\_ 11 \_\_\_\_\_ 2021 р. № 1675 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 08.12.2021 р.

3. Вихідні дані до роботи Виконати огляд і аналіз існуючих безпілотних літальних апаратів, функцій, які вони використовують, відомих методів і засобів виявлення БПЛА. Виконати моделювання динамічної повітряної обстановки у зоні критично важливих об'єктів інфраструктури. Модель обстановки повинна включати: модель випадкового потоку літальних апаратів, що перетинають зовнішню межу зони, що стосується критично важливого об'єкта; модель випадкового потоку локаційних позначок, відбитих від літальних апаратів, що у зоні критично важливого об'єкта (з урахуванням ймовірності їх виявлення та помилок визначення координат); модель потоку хибних позначок. Реалізувати математичні моделі в середовище Visual Studio.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

Вступ1. Методи виявлення і спостереження БПЛА2. Математичне моделювання польотів БПЛА3. Розробка програми моделювання динамічної обстановки

ВисновкиПерелік посиланьДодатки

5. Перелік графічного матеріалу із зазначенням обов'язкових креслеників, схем, плакатів, комп'ютерних ілюстрацій

1. Види БПЛА – 3 арк.

2. Радіолокаційний метод виявлення БПЛА - 2 арк.

3. Математичні моделі – 1 арк.

4. Середовище моделювання – 1 арк.

5. Програма моделювання – 4 арк.

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи (проекту)	Термін виконання етапів роботи	Примітка
1	Аналітичний огляд літератури	08.11.21–09.11.21	
2	Аналіз методів дослідження	10.11.21–11.11.21	
3	Розробка математичних моделей	12.11.21–13.11.21	
4	Розробка алгоритму моделювання	14.11.21–15.11.21	
5	Розробка програмної частини	16.11.21–19.11.21	
6	Комп'ютерне моделювання	20.11.21-21.11.21	
7	Розробка пояснювальної записки	22.11.21-02.12.21	
8	Графічна частина роботи	30.11.21-02.12.21	
9	Перевірка керівником	03.12.21-04.12.21	
10	Перевірка нормоконтролю	05.12.21-06.12.21	
	Перевірка на академічний плагіат	07.12.21	
11	Перевірка зав. кафедрою, рецензування	08.12.21	

Дата видачі завдання \_\_\_\_\_ 08.11.2021 р. \_\_\_\_\_

Студент \_\_\_\_\_  
(підпис)

Керівник роботи (проекту) \_\_\_\_\_ проф. Карташов В.м.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка випускної кваліфікаційної роботи магістра на тему: "Моделювання польотів БПЛА у зоні критично важливого об'єкта інфраструктури" 69 стор., 26 рис., 21 джерел.

БПЛА, ВИЯВЛЕННЯ, МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ, КОНТРОЛЬ КВОІ, РОЗРОБКА ПРОГРАМ НА C#, РОЗРОБКА НА UNITY, СИМУЛЯЦІЯ РОБОТИ РЛС

В роботі виконано огляд і аналіз існуючих безпілотних літальних апаратів, а також відомих методів виявлення БПЛА. Розроблено математичні моделі вторгнення БПЛА різних видів і типів в зону КВОІ, а також параметрів польотів БПЛА- висоти, швидкості, напрямку польоту. З використанням індикатора кругового огляду радіолокаційної станції створена модель динамічної повітряної обстановки в у зоні критично важливого об'єкту інфраструктури. Модель включає випадкові потоки літальних апаратів, які перетинають зовнішню межу зони, що відноситься до критично важливого об'єкту інфраструктури.

Методом виконання роботи є математичне моделювання, розробка математичних моделей, реалізація моделей з використанням мови програмування C#, в середовищі розробки Unity. Unity забезпечує широкі можливості для створення моделей різних видів для різних областей використання. Середовище програмування - Visual Studio.

Область використання – тестування існуючих засобів виявлення БПЛА з використанням індикатора кругового огляду РЛС шляхом симуляції польотів БПЛА.

## РЕФЕРАТ

Пояснительная записка выпускной квалификационной работы магистра по теме: "Моделирование полетов БПЛА в зоне критически важного объекта инфраструктуры" 69 стр., 26 рис., 21 источников.

БПЛА, ОБЪЯВЛЕНИЕ, МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ, КОНТРОЛЬ КВОИ, РАЗРАБОТКА ПРОГРАММ НА C#, РАЗРАБОТКА НА UNITY, СИМУЛЯЦИЯ РАБОТЫ РЛС

В работе выполнено обзор и анализ существующих беспилотных летательных аппаратов, а также известные методы и обнаружения БПЛА. Приведены основные математические модели полетов БПЛА. На основе радиолокационной станции была создана модель динамической воздушной обстановки в зоне критически важного объекта инфраструктуры. Модель включает случайные потоки летательных аппаратов, пересекающих внешнюю границу зоны, касающуюся критически важного объекта инфраструктуры.

Методом выполнения этой работы является программирование на языке C#, в среде разработки Unity. Unity предоставляет широкие возможности для создания моделей разных типов и для разных областей использования. Средой программирования для модели будет Visual Studio.

Область использования – тестирование существующих средств обнаружения БПЛА с использованием индикатора кругового обзора РЛС путём симуляции полетов БПЛА.

## ABSTRACT

Explanatory note of the master's final qualification work on the topic: "Simulation of UAV flights in the zone of a critical infrastructure object" 69 pages, 26 gob., 21 sources.

UAV, ANNOUNCEMENT, MATHEMATICAL SIMULATION, QUOI CONTROL, C # PROGRAM DEVELOPMENT, UNITY DEVELOPMENT, RADAR OPERATION SIMULATION

The paper reviews and analyzes existing unmanned aerial vehicles, as well as known methods and UAV detection. The main mathematical models of UAV flights are presented. Based on the radar station, a dynamic air situation model was created in the area of a critical infrastructure facility. The model includes random streams of aircraft crossing the outer edge of the zone, touching a critical infrastructure.

The method of doing this work is programming in C #, in the Unity development environment. Unity provides ample opportunities for creating models of different types and for different areas of use. The programming environment for the model will be Visual Studio.

Area of use - testing of existing UAV detection equipment using the radar circular view indicator by simulating UAV flights.

## ЗМІСТ

Перелік умовних позначень, символів, скорочень і термінів.....	8
ВСТУП.....	9
1 МЕТОДИ ВИЯВЛЕННЯ І СПОСТЕРЕЖЕННЯ БПЛА.....	11
1.1. Види БПЛА та їх функції.....	11
1.2. Захист КВОІ від БПЛА.....	19
1.3. Методи та системи виявлення БПЛА.....	21
2 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПОЛЬОТІВ БПЛА.....	28
2.1. Інформація про моделювання систем.....	28
2.2. Математичні моделі польотів БПЛА.....	35
3. РОЗРОБКА ПРОГРАМИ МОДЕЛЮВАННЯ ДИНАМІЧНОЇ ОБСТАНОВКИ.....	41
3.1. Вибір середовища моделювання.....	41
3.2. Моделювання роботи індикатора кругового огляду.....	47
3.3. Моделювання польотів БПЛА.....	58
ВИСНОВКИ.....	65
ПЕРЕЛІК ПОСИЛАНЬ.....	67
ДОДАТОК А. ГРАФІЧНА ЧАСТИНА.....	70
ДОДАТОК Б. ВІДОМІСТЬ КВАЛІФІКАЦІЙНОЇ РОБОТИ.....	82

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І  
ТЕРМІНІВ

- БПЛА - безпілотний літальний апарат;  
ЗПС - злітно-посадкова смуга;  
РЛС - радіолокаційна станція;  
КВОІ – критично важливий об’єкт інфраструктури;  
ЕОМ - електронна обчислювальна машина;  
АЕС - атомна електростанція;  
ППО – протиповітряна оборона;  
ЕМВ - електромагнітне випромінювання;  
КРУ - командні радіолінії управління;  
CLR - загальномовне виконуюче середовище;  
ЛА - літальний апарат.

## ВСТУП

В даний час безпілотні літальні апарати (БПЛА) знаходять широке застосування в різних сферах діяльності людства. Наприклад, у сільському господарстві БПЛА з GPS-навігацією застосовуються для запилення рослин на полях. При цьому досягається значна економія хімікатів і більш ретельна обробка посівів у порівнянні з авіацією, що пілотується. БПЛА [1] використовуються для доставки медикаментів та гуманітарних вантажів у важкодоступні райони, можуть застосовуватись для перевірки ліній електропередач та трубопроводів. Дрони, інша назва БПЛА, можуть використовуватись і державною службою з надзвичайних ситуацій для моніторингу та прогнозування, а також при контролі небезпечних об'єктів.

Водночас поява і розвиток безпілотних літальних апаратів значно збільшило можливості правопорушників і організованих груп в області тероризму, організованої злочинності, транспортування і розповсюдження наркотиків, контрабанди, стеження за окремими особами й організаціями, несанкціонованим моніторингом об'єктів і територій державного значення (АЕС, закритих підприємств). Значні проблеми породжує неадекватна поведінка деяких власників БПЛА, що призводить, наприклад, до погроз повітряному транспорту в районах аеропортів (може привести до зіткнення БПЛА з літаками цивільної авіації при зльоті та посадці), іншим видам транспорту, відомі випадки психічних розладів, викликаних польотами БПЛА.

З'явилася необхідність якимось чином виявляти БПЛА для захисту людей та підприємств. Почали розроблятися різні системи для швидкого та ефективного виявлення дронів. Метод моделювання в свою чергу поліпшує розробку систем виявлення та ідентифікації БПЛА. Математична модель допомагає отримати інформацію про БПЛА, шляхом проведення експериментів, які імітують та аналізують поведінку дронів. Тим самим

знаходячи необхідні патерни які ввійдуть в основу систем виявлення та ідентифікації БПЛА, удосконалюючи процес розробки.

Метою цієї роботи є захист об'єктів інфраструктури від злочинців, для чого нам і потрібно моделювати політ БПЛА, для більш ефективного спостереження та протидії БПЛА.

## 1. МЕТОДИ ВИЯВЛЕННЯ І СПОСТЕРЕЖЕННЯ БПЛА.

### 1.1. Види БПЛА та їх функції

Для кращого розуміння дронів треба спочатку розібратися в їхньому влаштуванні. БПЛА зазвичай складається з таких основних елементів [2].

Рама. Є основою літаючого апарату. Саме на неї встановлюються усі елементи. У більшості випадків її роблять із полімерів та різних сплавів металів.

Польотний контролер. Тобто сам дрон. Саме на нього приходять сигнали від пульта керування. У контролер входять процесор, барометр, який визначає висоту, акселерометр, гіроскоп, GPS-навігатор, оперативний пристрій, пристрій прийому сигналу.

Двигун, регулятори та пропелери відповідають за політ безпілота. За допомогою регулятора задається швидкість БПЛА;

Акумуляторна батарея. Використовується як джерело енергії для двигуна, а також інших елементів дрону;

Пульт керування. Комерційні та споживчі безпілоти управляються за допомогою них. Деякі військові агрегати теж управляються за допомогою пульта, але можливі й супутникові системи.

Але незважаючи на схожі елементи, всі БПЛА сильно відрізняються за рахунок власної конструкції.

За різноманітністю конструкції існує безліч типів безпілотних літальних апаратів, кожен з яких має свої особливості та створений для своїх завдань.

#### 1.1.1. Мультикоптери



Рисунок 1.1 - Мультикоптер DJI MATRICE 200

Мультикоптерні дрони (Рис. 1.1.) є платформою, що літає, з безколекторними двигунами та пропелерами. Вони мають можливості для вертикальних зльотів або зависання над об'єктом і є найдешевшою та найдоступнішою версією БПЛА.

Коптери [2] різняться між собою числом пропелерів - їх може бути 2, 3, 4, 6, 8 і більше. Найбільш масовими є на 2015-2019 роки квадрокоптери з 4 пропелерами, але також зустрічаються, як правило, більш вантажопідйомні, гекса- та октакоптери. Є, зрозуміло, і екзотичні двох-і три-гвинтові коптери.

Інша можлива опціональна конструктивна відмінність мультикоптерів - складність конструкції. Випускається кілька різних моделей коптерів, які легко складаються для зручності перенесення і легко готуються до використання.

Переваги:

- простота у використанні (не вимагає для зльоту катапульти або ЗПС);
- здатність зависати у заданій точці;
- не вимагає для посадки парашута або ЗПС (злітно-посадкової смуги).

Недоліки:

- малий час польоту;
- низька швидкість;

- спрощене пілотування, якщо порівнювати з апаратами з фіксованим крилом.

### 1.1.2. Літаки (безпілотники з фіксованим крилом)



Рисунок 1.2 - БПЛА літакового типу X8 Flying Wing

БПЛА з нерухомим крилом (Рис. 1.2.) реалізують структуру літака та ідеально підходять для далеких операцій. Здатний літати завдяки підйомній силі, створюваної аеродинамічної формою фіксованого крила під час руху вперед з певною швидкістю, розвиток якої досягається різними способами (частіше рахунок двигуна або завдяки раціональному використанню висхідних повітряних потоків).

#### Переваги:

- висока швидкість польоту;
- висока дальність польоту.

#### Недоліки

- старт з катапульті;
- посадка на парашуті або з ЗПС;
- потрібні навички пілотування.

### 1.1.3. Гелікоптери (БПЛА вертолітного типу)



Рисунок 1.3 - БПЛА ТБ-29В Тайбер

Можна розрізнати вертолітний БПЛА (Рис. 1.3.) - автономні та телекеровані, що допускають опціональне керування пілотом, що знаходяться на борту або не передбачають екіпажу на борту.

Переваги:

- не вимагає для зльоту катапульти або ЗПС;
- енергоефективність - краще, ніж у мультикоптера за рахунок використання гвинта більшого діаметра (більше питоме навантаження на потужність);
- хороші аеродинамічні якості;
- здатність зависати у заданій точці;
- не вимагає для посадки парашута чи ЗПС;
- найчастіше більша вантажопідйомність, ніж у мультикоптерів;
- легше масштабуються за необхідності роботи з великими вантажами.

Недоліки:

- малий час польоту порівняно з БПЛА літакового типу;
- низька швидкість польоту;
- порівняно складне пілотування;
- висока вартість літньої години щодо мультикоптерів;
- складна механіка.

#### 1.1.4. Конвертоплани



Рисунок 1.4 - Конвертоплан ЭРА-54Д

Конвертоплан (Рис. 1.4.) - це безпілотник, який сідає і злітає "вертолітним" способом, за рахунок повороту його двигунів, а в польоті рухається як літак з опорою на фіксоване крило. Корпус безпілотника залишається у горизонтальному положенні. Як варіант, двигуни можуть залишатися у фіксованому положенні, а напрям тяги задає відхилення жалюзі.

Переваги:

- здатність зльоту як у коптера (не вимагає катапульти або ЗПС);
- здатність польоту по-літаковому;
- здатність до зависання у заданій точці;
- простота у використанні.

Недоліки:

- більш складна конструкція, ніж у літака;
- При повороті двигунів в іншу площину під час їхньої роботи доводиться боротися з моментом інерції, що знижує надійність конвертопланів.
- Знижений час роботи в повітрі при інших рівних, якщо порівнювати з літаком з тими самими двигунами та запасом енергії.

#### 1.1.5. Глайдери (планери)

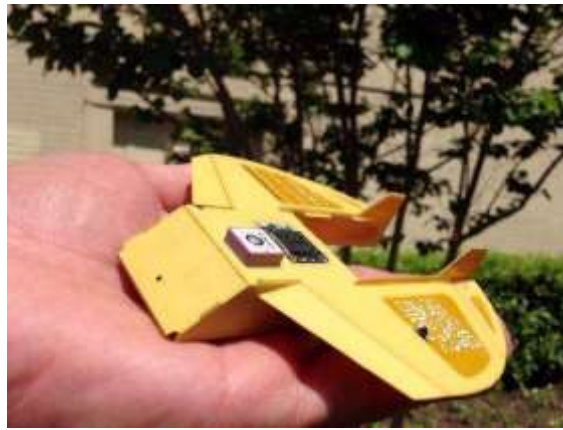


Рисунок 1.5 - Мікродрон глайдер Cicada

Глайдери (Рис 1.5) - це БПЛА без двигуна або з двигуном, потужність якого недостатня для забезпечення постійного утримання машини в повітрі, але достатня для коригування курсу БПЛА для того, щоб забезпечити його приземлення в точці із заданими координатами або, наприклад, тривале знаходження машини в повітрі з використанням висхідних потоків . Можуть не мати двигуна, але використовувати, наприклад, керма курсу та висоти, керовані бортовим процесором, для досягнення описаних вище цілей [2].

Як правило, призначені для розвідки. Зібрана інформація передається по радіо до пункту керування через літак або дрон супроводу або через супутник.

#### 1.1.6. Тейлсіттери



Рисунок 1.6 - Тейлсітер від компанії Neurobotics

Тейлсітери (Рис. 1.6.) – це безпілотник вертикального зльоту, який опинившись у повітрі, повертається горизонтально і летить, як дрон літакового типу. Для посадки такий безпілотник знову повертається у вертикальне положення і приземляється на спеціальні "ребра", що відходять від крил і хвоста, які є опорою. Від конвертопланів таку конструкцію відрізняє відсутність поворотних елементів.

#### 1.1.7. Прив'язні безпілотники



Рисунок 1.7 - Прив'язний мультикоптер DragonFly Pictures

Ідея цієї конструкції - прибрати з безпілотника найважчу його частину - батарею живлення (натомість, щоправда, з'являється кабель живлення, який, по-перше, навантажує дрон, також суттєво обмежуючи можливості його переміщення). Прив'язний дрон (Рис. 1.7.) підключається до наземного джерела живлення - блоку живлення, з'єднаного з мережею промислового струму або потужного акумулятора. Після чого такий показник, як час безперервного польоту, стає дуже великим - години, доба, тижні. Такий дрон, як правило, не літає туди-сюди, а залишається на місці, над точкою, куди веде його провід живлення.

Подібні безпілотники можуть стати в нагоді для спостереження. БПЛА забезпечує спостереження за допомогою бортової відеокамери (камер) з

можливістю огляду до 360 градусів. Так само дрон може виконувати функції антени, ретранслятора або базової станції стільникового зв'язку, яка дозволить вести сеанс зв'язку навіть у ситуаціях, коли з землі цього зробити не вийшло б через кривизну поверхні нашої планети.

#### 1.1.8. Мініатюрні безпілотники



Рисунок 1.8 - Мініатюрний БПЛА Black Hornet

БПЛА подібного типу (Рис. 1.8.) можуть мати різну конструкцію від глайдерів до гелікоптерів. Але основне, що об'єднує всі ці безпілотники, це "розмах крил", який у таких моделях до 10 см [2].

Такі дрони використовуються в основному для розвідки або зйомки, і через невеликі розміри керувати ними в закритих приміщеннях набагато простіше ніж більшими мультикоптерами.

#### 1.1.9. Модульні безпілотники



Рисунок 1.9 - Модульний БПЛА Alenia Aermacchi Sky-Y

Модульні БПЛА (Рис. 1.9.) – це безпілотники, які можуть збиратися у різній конфігурації залежно від завдання із уніфікованих модулів. Ці модулі можуть мати різні завдання від перенесення вантажів до швидшого польоту за допомогою додаткових пропелерів тощо. БПЛА такої конструкції є найуніверсальнішим із дронів, проте додаткові модулі можуть займати багато місця і приєднання їх до безпілотника може займати багато часу.

## 1.2. Захист КВОІ від БПЛА

Як видно, за досить недовгий час існування з'явилося безліч видів БПЛА, які допомагали в безлічі різних ситуацій і були корисні для багатьох професій. Але, з іншого боку, БПЛА здатні нести потенційну загрозу для різних сфер діяльності людини – господарської, повсякденної та військової.

Порівняно невисока вартість безпілотних апаратів, а також труднощі їх виявлення та спостереження за допомогою сучасної техніки призвело до підвищення масовості протиправних дій із використанням БПЛА та їх безкарності. Зважаючи на це - завдання захисту різних об'єктів від впливу БПЛА, які можуть використовуватися як окремими правопорушниками, так і організованими злочинними угрупованнями та арміями ймовірних супротивників, є одним із актуальних завдань сучасності.

Найважливішим є завдання захисту від атак БПЛА критично важливих об'єктів інфраструктури (КВОІ), до яких відносять, атомні електростанції (АЕС), аеропорти, та ключові об'єкти інформаційної інфраструктури, очисні споруди, лабораторії.

13 лютого 2017 р. під головуванням Міністра закордонних справ України Павла Клімкіна, Рада Безпеки Організації Об'єднаних Націй ухвалила резолюцію 2341 про захист КВОІ та розширення можливостей держав щодо запобігання нападам на критично важливі об'єкти інфраструктури, ініціатором якої виступила Україна. В цій резолюції закликають держави протистояти небезпеці терористичних атак на КВОІ. Резолюція пропонує всім державам розробити національні стратегії, а також вжити можливих превентивних заходів та конкретних дій у цьому напрямку.

Об'єкти інфраструктури є системами життєзабезпечення повсякденного існування спільнот, які підтримуються комплексною та складною мережею різних споруд та інфраструктурних систем, це було наголошено у резолюції. Громадяни покладаються на функціонування установ та служб щодо свого здоров'я, безпеки, охорони економічного та екологічного благополуччя.

До основних видів життєво важливої матеріальної інфраструктури відносяться [3] медичні послуги, повітряні та наземні перевезення, банківська сфера, виробництво та розподілення електроенергії, а також державні аварійно-рятувальні служби. Основні види матеріальної інфраструктури що неспроможні нині повноцінно працювати без комп'ютерів і мереж, представлених насамперед системами диспетчерського управління та збору даних (SCADA), взаємозв'язок яких дозволяє їм обмінюватися інформацією та виконувати аналіз з усіх критично важливих функцій.

А головне що ці атаки можуть мати далекосяжні наслідки, тому що існує взаємозалежність інфраструктури на рівні секторів та галузей, між кібер та фізичними областями, а також міжнаціональними кордонами. Одна атака на єдину точку збою може призвести до порушення кількох життєво важливих систем безпосередньо в країні, або навіть за ланцюговою реакцією,

у всьому світі. Це створює манливу мету для злочинців і тих, хто має намір завдати збитки та шкоди. У той час як населені пункти та інфраструктура по всьому світу розвиваються, те саме вдосконалення відбувається і зі зброєю нападу, одним із видів якого можуть виступати БПЛА.

Для виявлення БПЛА використовують [3] комплексну обробку сигналів різної фізичної природи та різних діапазонів хвиль, оскільки БПЛА є надзвичайно складним для виявлення і спостереження об'єкт, здатний виконувати обширний клас завдань. Комплексна обробка сигналів дозволяє підвищити ефективність спостереження БПЛА при різних станах зовнішнього середовища та при різних несприятливих ситуаціях та дозволить забезпечити захист критично важливих об'єктів інфраструктури з достатнім ступенем надійності.

Саме тому надзвичайно важливо мати математичні та програмно реалізовані моделі, що формують динамічну повітряну обстановку в зоні КВОІ, які допоможуть при розробці технічних засобів захисту критично важливих об'єктів інфраструктури від атак БПЛА. Це дозволить моделювати обстановку різного ступеня складності, що включає використання різних типів та видів БПЛА, а також виявляти режими інтенсивності атак, близькі до критичних.

### 1.3. Методи та системи виявлення БПЛА

Існують декілька основних методів виявлення БПЛА, для початку розберемо кожен з них [4].

Оптичний метод – це метод якій використовує для виявлення БПЛА відеокамери які знімають на території та аналізують відео за допомогою алгоритму та ідентифікують ціль.

Апаратура оптичного виявлення безпілотних літальних апаратів легко вбудовується у вже існуючу систему безпеки об'єкта, оскільки є камерами високої роздільної здатності. Спеціальне програмне забезпечення налаштовує

їх на малорозмірні повітряні цілі та дозволяє записувати відео наближення дрону до об'єкта. Оптична детекція найкраще підходить для щільної міської забудови, оскільки має порівняно невисокий радіус дії та допускає будь-який рівень шуму. Її мінус у тому, що дрони, зібрані любителями, часто мають силуети, надто відмінні від тих, що містяться в базі даних системи, і, отже, вона може їх не впізнати

Оптичне виявлення БПЛА [5] дуже залежить від чинників довкілля. Збільшення дальності виявлення досягається за рахунок звуження поля зору, зменшення зони огляду та збільшення часу пошуку. Тому візуальні сенсори є неефективними пристроями щодо пошуку. Оскільки безпілотники є значно меншими за розмірами порівняно з пілотованими засобами, це ускладнює їх виявлення за допомогою оптичних засобів. Порівняно з літаком контрастність БПЛА щодо фону є меншою через відсутність світлових маяків, зменшений факел малого двигуна та меншу поверхню відображення.

Акустичний метод – це метод який використовує звукові датчики які реагують на спеціальні звукові характеристики що асоціюються з дронами.

Апаратура акустичного виявлення БПЛА [6] проста в установці та експлуатації, не вимагає дозволу спецслужб і однаково ефективна як у денний, так і в нічний час. Звукові характеристики (сигнатура) дрону, що летить, передаються з акустичного датчика на сервер, де порівнюються з сигнатурами всіх безпілотників, зведеними в особливу базу даних. При збігу з відповідною сигнатурою (пізнання об'єкта як дрона) формується зворотна команда на оповіщення. Акустична детекція може мати досить високу дальність дії та несприйнятливості до перешкод типу дерев, дротів та антен. Однак у міському середовищі високий рівень фонового шуму представлятиме певну складність для системи, яка працює на такому принципі. Отже, ці системи найбільше підходять для замських об'єктів.

Сумарний спектр акустичного випромінювання [7] тактичного БПЛА обумовлений гармонійними та широкосмуговими складовими. Він включає в себе гармонійні які залишаються після випромінювання двигуна, шуму

обертів гвинта, випромінювання механічної природи, а також високочастотну і низькочастотну складові шуму двигуна з безперервними по частоті спектрами. У шумі силової установки БПЛА, що має поршневий двигун повітряного охолодження, за відсутності в його вихлопному тракті глушника визначальним джерелом зовнішнього шуму є поршневий двигун.

Радіотехнічний метод [4] – це метод який використовує радіолокаційну станцію. Формувати динамічну повітряну обстановку в зоні КВОІ ми будемо на прикладі індикатора радіолокаційної станції. РЛС буде нашою станцією спостереження, від якої ми отримуватимемо сигнал відображення БПЛІ на території КВОІ. Тому розберемо цей метод більш детально.

Спочатку розберемося у тому, як працює РЛС. Радіолокаційна станція (РЛС) - радіотехнічна система для виявлення повітряних, морських та наземних об'єктів, а також для визначення їхньої дальності, швидкості та геометричних параметрів. Використовує метод радіолокації, заснований на випромінюванні радіохвиль та реєстрації їх відбиття від об'єктів.

Пошук БПЛА за допомогою активних станцій радіолокації досить продуктивний, так як вони мають відносно великий імпульсний обсяг пошуку і значну дальність виявлення.

Тепер думаю, слід уважно врахувати такі основні аспекти, через які виявлення БПЛА набагато складніше, ніж, наприклад, виявлення традиційних літаків:

-БПЛА можна правильно ідентифікувати лише на дуже коротких дистанціях. Ефективні системи спостереження повинні мати можливість швидко реагувати та приймати відповідні контрзаходи навіть у несприятливих умовах експлуатації, таких як низька видимість, навколишнє середовище поширення, повне перешкод тощо.

-Серйозні загрози безпеці можуть походити від безпілотних літальних апаратів, що наближаються зграями. Прийняті технології повинні мати можливість виявляти кілька цілей одночасно і відстежувати їх траєкторії як реального часу.

-БПЛА нелегко відрізнити від інших невеликих літаючих об'єктів, наприклад, птахів. У цьому випадку необхідні удосконалені алгоритми обробки сигналів, щоб знизити ймовірність помилкової тривоги та підвищити точність виявлення.

Завдання ідентифікації цілі можна перетворити на (складне) макро-завдання, яке включає три різні етапи [8].

Перший, названий виявленням, полягає у використанні статистичної теорії перевірки гіпотез для ухвалення рішення про можливу присутність цілі.

Після того, як спрацювало виявлення, може наслідувати етап перевірки, щоб перевірити, чи є ціль насправді чи ні. Цей етап важливий для зменшення кількості помилкових тривог, які можуть бути зроблені на першому етапі, і до нього можна підійти різними способами, у тому числі з використанням зовнішньої інформації, додаткових систем або навіть ручного втручання.

Якщо виявлення підтверджується, на заключному етапі класифікації ціль призначається певної категорії з урахуванням деяких відмітних атрибутів (наприклад, розміру, кількості роторів, максимальної корисної навантаження тощо. буд.). Весь процес зображено на рисунку 1.10.

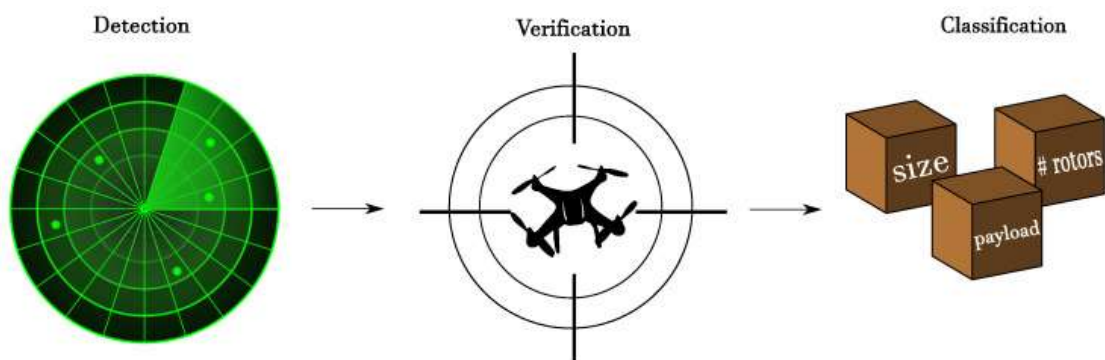


Рисунок 1.10 - Процес ідентифікації цілі

З цих трьох етапів для нас найважливішим буде перший виявлення. Радіолокаційні датчики залишаються важливим компонентом систем спостереження та виявлення дронів. Як відомо, цей підхід, по суті,

ґрунтується на електромагнітному принципі зворотного розсіювання, яке відбувається, коли об'єкт висвітлюється променем радара. Основні труднощі для невеликих безпілотних літальних апаратів полягають в тому, що ймовірність виявлення сильно залежить від радіолокаційного перерізу цілей, яке досить мале для безпілотних літальних апаратів, які здебільшого є пустотними і зроблені з пластику. З цієї причини були розроблені нові радіолокаційні установки, які намагаються використовувати зворотне розсіювання від частин, що обертаються, таких як пропелери та ротори, для оцінки мікродоплерівських сигнатур.

Ще однією проблемою є те, що через невеликий розмір безпілотних літальних апаратів, з урахуванням обмежень у смузі пропускання і, що важливо, через високу вартість, радіолокаційні радари спостереження не можуть використовуватися часто, особливо коли необхідно розгорнути мережу радарів, щоб гарантувати повне покриття контрольованої зони. Радіолокатори безперервного (CW) і безперервного випромінювання з частотною модуляцією (FMCW) в даний час є найбільш привабливим і економічним рішенням цієї проблеми.

FMCW-сигнал, [9] також відомий як лінійно-частотно-модульований (LFM) сигнал, складається з лінійно-модульованої безперервної радіохвилі, що передається в бажаному напрямку. Ці види сигналів часто називають щебетанням і відрізняються від CW, тому що в останньому робоча частота не змінюється під час передачі, як показано на рис. 1.11(a, b).

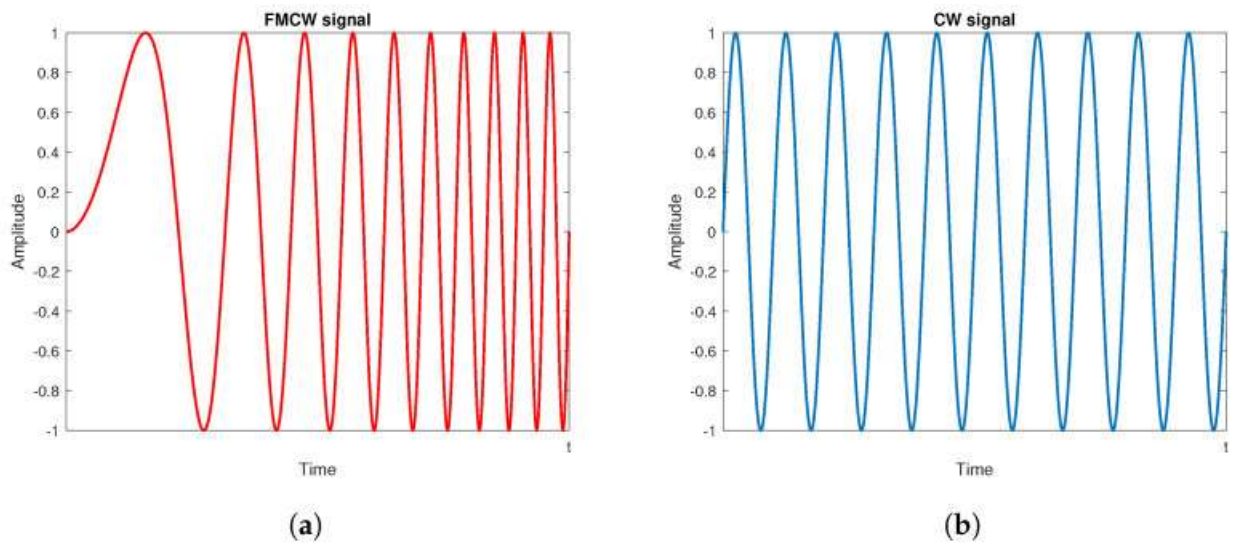


Рисунок 1.11 - (a) Частотно-модульовані сигнали безперервної хвилі (FMCW) та (b) безперервні хвилі (CW)

Радари, що використовують такі форми сигналів, стали дуже популярними, особливо в автомобільній області, через низьку вартість апаратних компонентів і завдяки їхній здатності надавати інформацію як про дальність, так і про режим Доплера. Передані та прийняті сигнали FMCW схематично зображені на рисунку 1.12; з цих сигналів можна отримати інформацію про затримку  $\tau$  і фазі  $\phi$ , які корисні для одночасного отримання інформації про відстань і швидкість однієї або декількох цілей.

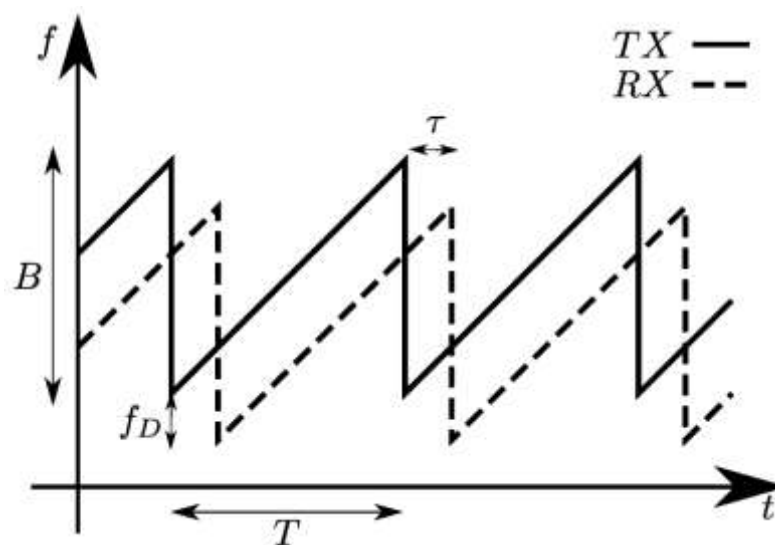


Рисунок 1.12 - FMCW сигнали, що передаються і приймаються

Основна обробка прийнятого сигналу виконується за допомогою I/Q-демодуляції (як показано на рисунку 1.13), яка забезпечує синфазні та квадратурні компоненти комплексного сигналу основної смуги частот, що називається сигналом биття або сигналом проміжної частоти (ПЧ). Перетворення зі зниженням частоти введено, щоб значно спростити реалізацію схем обробки, дозволяючи схемі працювати на набагато більш низькій частоті порівняно з сигналом, що передається. Істотна різниця між ланцюжками обробки FMCW і CW, див. рис. 1.13, полягає в генераторі сигналу, що управляє, який забезпечує опорний сигнал (у разі CW він постійний). Об'єднання цього сигналу з генератором, керованим напругою (ГУН), дає результуючий радіочастотний сигнал, який передаватиметься радаром [9].

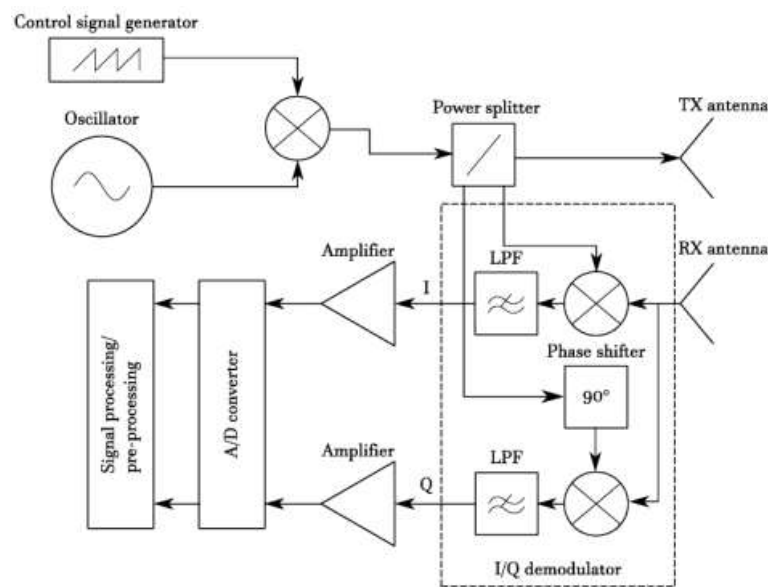


Рисунок 1.13 - Архітектура радара FMCW

Сучасні системи спостереження [9], що складаються з мережі просторово-розподілених датчиків, є найбільш перспективним підходом для забезпечення повного охоплення спостережуваної території та використання переваг різних технологій, впораючись із їхніми окремими недоліками. Ця робота зосередилась на радарних датчиках, які є ключовою технологією, також через низьку вартість і здатність працювати на відносно великих відстанях, високу стійкість до освітлення та погодних умов.

## 2. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПОЛЬОТІВ БПЛА

### 2.1. Інформація про моделювання систем

Моделювання – це [10] процес заміщення об'єкта, що вивчається, іншим з метою отримання інформації про найважливіші властивості об'єкта-оригіналу за допомогою об'єкта-моделі, тобто моделювання, може бути ще визначено як подання об'єкта моделлю для отримання інформації про цей об'єкт шляхом проведення експериментів з його моделлю.

В основі моделювання лежить теорія подібності, яка стверджує, що абсолютна подоба може мати місце лише при заміні одного об'єкта іншим таким самим. При моделюванні абсолютна подоба немає місця, тому необхідно прагнути до того, щоб модель досить добре відображала досліджувану сторону функціонування об'єкта.

Визначення же системи - це повний, цілісний набір елементів (компонентів), взаємопов'язаних та взаємодіючих між собою так, щоб могла реалізуватися функція системи. Рівні в системі можуть бути підпорядковані один одному, утворюючи ієрархію зв'язків.

Класифікація видів моделювання може бути проведена з різних підстав. Один із варіантів класифікації в своїй основі буде мати такі способи моделювання:

- наочне (гіпотетичне);
- символічне (мовне або знакове);
- натуральне (науковий експеримент та комплексні випробування);
- фізичне (у реальному часі або у модельному часі);
- математичне (аналітичне та імітаційне).

Ми будемо працювати з математичною моделлю тому далі будемо розмовляти про неї.

Математична модель виражає суттєві риси об'єкта чи процесу мовою рівнянь та інших математичних засобів. Власне кажучи, сама математика

зобов'язана своїм існуванням тому, що вона намагається відобразити, тобто промодельовати своєю специфічною мовою закономірності навколишнього світу. Шлях математичного моделювання в наш час набагато більш всеосяжний, ніж моделювання натурного. Математичне моделювання набуло інтенсивного розвитку з появою ЕОМ, проте метод зародився одночасно з математикою тисячі років тому.

Математичне моделювання не завжди супроводжується комп'ютерною підтримкою. Перш ніж створити математичну модель роблять все можливе для аналітичного дослідження моделі. Аналітичні рішення (тобто представлені формулами, що виражають результати дослідження через вихідні дані) зазвичай зручніші та інформативніші за чисельні. Можливості аналітичних методів вирішення складних математичних завдань, однак, дуже обмежені і, як правило, ці методи набагато складніші за чисельні. У цій моделі домінують чисельні способи, реалізовані на комп'ютерах. Це з тим, що моделювання тут розглядається під кутом зору комп'ютерних (інформаційних) технологій. Такий підхід дещо звужує можливості методу загалом; його гідність - деяке зниження бар'єра необхідної математичної підготовки (хоча, зрозуміло, і в чисельні методи при професійному занятті математичним моделюванням доводиться заглиблюватись настільки, що при цьому потрібна значна математична освіта). Нарешті, відзначимо, що поняття «аналітичне рішення» і «комп'ютерне рішення» аж ніяк не протистоять один одному, оскільки комп'ютери при математичному моделюванні використовуються як для чисельних розрахунків, так і для аналітичних перетворень; результат аналітичного дослідження математичної моделі часто виражений дуже складними рівняннями, при цьому диференціально-інтегральні рівняння призводять до чисельної реалізації.

Математична модель являє собою формалізований опис системи (або операції) деякою абстрактною мовою, наприклад, у вигляді сукупності математичних співвідношень або схеми алгоритму, тобто такий математичний опис, який забезпечує імітацію роботи систем або пристроїв

на рівні, досить близькому до їхньої реальної поведінки, одержуваному при натурних випробуваннях систем чи пристроїв. Будь-яка математична модель описує реальний об'єкт, явище чи процес із деяким ступенем наближення до дійсності. Вид математичної моделі залежить від природи реального об'єкта, і від завдань дослідження.

Математичне моделювання суспільних, економічних, біологічних та фізичних явищ, об'єктів, систем та різних пристроїв є одним із найважливіших засобів пізнання природи та проектування найрізноманітніших систем та пристроїв, прогнозування системи в цілому.

Метою математичного моделювання є аналіз реальних процесів математичними методами. Математичне моделювання для дослідження характеристик систем можна поділити на аналітичне, імітаційне та комбіноване. У свою чергу, математичні моделі поділяються на імітаційні та аналітичні.

Аналітичне моделювання [10] — це функціонування елементів системи, підпорядкованих певним законам, які записуються як деяких функціональних співвідношень (алгебраїчних, інтегральних, диференціальних, тощо.) чи логічних умов.

Аналітична модель може бути досліджена такими методами: аналітичним, явні залежності для шуканих характеристик; чисельним, рівняння призводять до чисельної реалізації, щоб одержати чисельних результатів при конкретних початкових даних; якісним, не маючи рішення у явному вигляді, можна знайти деякі властивості рішення.

Повне дослідження процесу функціонування системи можна провести, коли відомі явні залежності, що пов'язують шукані характеристики з початковими умовами, параметрами та змінними. Такі залежності вдається отримати лише порівняно простих систем. При ускладненні систем дослідження їх аналітичним методом наштовхується на значні труднощі, які часто бувають непереборними. Тому, бажаючи використовувати аналітичний метод, у разі йдуть істотне спрощення початкової моделі, щоб мати

можливість вивчити хоча б загальні властивості системи. Таке дослідження на спрощеній моделі аналітичним методом допомагає отримати орієнтовні результати визначення більш точних оцінок іншими методами.

Чисельний метод дозволяє досліджувати (порівняно з аналітичним методом) ширший клас систем, але при цьому отримані рішення мають приватний характер. Чисельний метод особливо ефективний при використанні ЕОМ. В окремих випадках дослідження системи можуть задовольнити ті висновки, які можна зробити при використанні якісного методу аналізу математичної моделі. Такі якісні методи широко використовуються, наприклад, теорії автоматичного управління для оцінки ефективності різних варіантів систем управління.

При імітаційному моделюванні [10] алгоритм, що реалізує, відтворює процес функціонування системи в часі, причому імітуються елементарні явища, що становлять процес, зі збереженням їх логічної структури і послідовності перебігу в часі, що дозволяє за вихідними даними отримати відомості про стани процесу в певні моменти часу, що дають можливість оцінити показники системи. Перевагою імітаційного моделювання (проти аналітичним) є можливість розв'язання складніших завдань.

Імітаційні моделі дозволяють досить просто враховувати такі фактори, як наявність дискретних та безперервних елементів, нелінійні характеристики елементів системи, численні випадкові впливи та ін., які часто створюють труднощі під час аналітичних досліджень. В даний час імітаційне моделювання - найбільш ефективний метод дослідження великих систем, а часто і єдиний, практично доступний метод отримання інформації про поведінку системи, особливо на етапі її проектування. Якщо результати, отримані при відтворенні на імітаційній моделі процесу функціонування системи, є випадковими величинами і функціями, то визначення характеристик процесу потрібно багаторазове відтворення з наступної статистичної обробкою даних і як методу машинної реалізації імітаційної моделі слід використовувати метод статистичного моделювання.

Також існує комбіноване (аналітико-імітаційне) моделювання при аналізі та синтезі систем дозволяє об'єднати переваги аналітичного та імітаційного моделювання. При побудові комбінованих моделей проводиться попередня декомпозиція процесу функціонування об'єкта на складові підпроцеси і тих, де це можливо, використовуються аналітичні моделі, а інших підпроцесів будуються імітаційні моделі.

Сутність побудови математичної моделі полягає в тому, що реальна система спрощується, схематизується та описується за допомогою того чи іншого математичного апарату. Можна виділити такі основні етапи побудови моделей [10].

#### 2.1.1. Змістовний опис модельованого об'єкта

Об'єкти моделювання описуються з позицій системного підходу. Виходячи з мети дослідження встановлюються: сукупність елементів, взаємозв'язки між елементами, можливі стани кожного елемента, суттєві характеристики станів та відносини між ними. Наприклад, фіксується, що якщо значення одного параметра зростає, то значення іншого — меншає і т. д. Питання, пов'язані з повнотою і єдністю вибору характеристик, не розглядаються. Звісно, у такому словесному описі можливі логічні протиріччя, невизначеності. Це вихідна природничо-наукова концепція досліджуваного об'єкта. Таке попереднє, наближене уявлення системи називають концептуальною моделлю. Для того щоб змістовний опис служив гарною основою для подальшої формалізації, потрібно докладно вивчити об'єкт, що моделюється. Нерідко природне прагнення прискорити розробку моделі відводить дослідника від цього етапу безпосередньо до вирішення формальних питань. У результаті побудована без достатнього змістовного базису модель виявляється непридатною для використання. На цьому етапі моделювання широко застосовуються якісні методи опису систем, знакові та мовні моделі.

### 2.1.2. Формалізація операцій

На основі змістовного опису визначається вихідна множина характеристик системи. Для виділення суттєвих характеристик необхідний хоча б наближений аналіз кожної їх. Під час проведення аналізу спираються на постановку завдання та розуміння природи досліджуваної системи. Після виключення несуттєвих характеристик виділяють керовані та некеровані параметри та виробляють символізацію. Потім визначається система обмежень значення керованих параметрів. Якщо обмеження не мають принципового характеру, то ними нехтують.

Подальші дії пов'язані з формуванням цільової функції моделі. Відповідно до відомих положень вибираються показники результату операції та визначається приблизний вид функції корисності на результатах. Якщо функція корисності близька до граничної (або монотонної), то оцінка ефективності рішень можлива безпосередньо за показниками результату операції. У цьому випадку необхідно вибрати спосіб згортки показників (спосіб переходу від безлічі показників до одного узагальненого показника) і зробити саму згортку. По пачунку показників формуються критерій ефективності та цільова функція.

Якщо при якісному аналізі виду функції корисності виявиться, що її не можна вважати граничною (монотонною), пряма оцінка ефективності рішень через показники результату операції неправомірна. Необхідно визначати функцію корисності і вже на її основі вести формування критерію ефективності та цільової функції.

Загалом заміна змістовного опису формальним — це ітеративний процес.

### 2.1.3. Перевірка адекватності моделі

Вимога адекватності перебуває у протиріччі з вимогою простоти, і це треба враховувати під час перевірки моделі на адекватність. Вихідний варіант моделі попередньо перевіряється за такими основними аспектами:

- включення всіх суттєвих параметрів до моделі;
- присутність у моделі несуттєвих параметрів;
- правильність відображення функціональних зв'язків між параметрами;
- правильність визначення обмеження на значення параметрів.

Для перевірки рекомендується залучати фахівців, які не брали участі у розробці моделі. Вони можуть об'єктивніше розглянути модель і помітити її слабкі сторони, ніж її розробники. Така попередня перевірка моделі дає змогу виявити грубі помилки. Після цього приступають до реалізації моделі та проведення досліджень. Отримані результати моделювання піддаються аналізу відповідності відомим властивостям досліджуваного об'єкта. Для встановлення відповідності створюваної моделі оригіналу використовуються такі шляхи:

- порівняння результатів моделювання з окремими експериментальними результатами, отриманими за однакових умов;
- використання інших близьких моделей;
- зіставлення структури та функціонування моделі з прототипом.

Головним шляхом перевірки адекватності моделі об'єкту, що досліджується, виступає практика. Однак вона вимагає накопичення статистики, яка далеко не завжди буває достатньою для отримання надійних даних.

Для багатьох моделей перші два прийняті меншою мірою. У цьому випадку залишається один шлях: висновок про подібність моделі та прототипу робити на основі зіставлення їх структур та функцій, що реалізуються. Такі висновки не мають формального характеру, оскільки ґрунтуються на досвіді та інтуїції дослідника.

За результатами перевірки моделі на адекватність приймається рішення про можливість її практичного використання або про проведення коригування.

При коригуванні моделі можуть уточнюватися суттєві параметри, обмеження значення керованих параметрів, показники результату операції, зв'язку показників результату операції з суттєвими параметрами, критерій ефективності. Після внесення змін до моделі знову виконується оцінка адекватності.

Оптимізація моделі. Сутність оптимізації моделей полягає у їх спрощенні при заданому рівні адекватності. Основними показниками, за якими можлива оптимізація моделі, є час та витрати коштів для проведення досліджень на ній. В основі оптимізації лежить можливість перетворення моделей з однієї форми на іншу. Перетворення може виконуватися з використанням математичних методів, або евристичним шляхом.

## 2.2. Математичні моделі польотів БПЛА

Спершу потрібно дізнатися що математичний метод моделювання реалізує для систем виявлення БПЛА. По перше, він здійснює оцінку характеристик та рішень, які будуть використані в цих системах або в їх інформаційних каналах на стадії проектування. По друге, підправляє та оптимізує параметри системи. И насамперед, детально вивчає те, як різні фактори впливають на роботу станції [11].

Застосування математичних моделей для моделювання систем має наступні переваги [3]:

- дослідження більш складних процесів з зваженням багатьох факторів;
- знижена вартість та час на виконання експериментів;
- більш проста організація експериментів;
- контроль результатів;
- виконання залежних один від одного випробувань.

Тепер розберемо основні елементи моделі яку будемо моделювати та розташування їх в просторі. У нас є КВОІ та поряд з ним система спостереження та виявлення БПЛА, у нашому випадку радіолокаційна станція. Їх оточує коло радіусом  $r$ , що є межею досяжності виявлення дронів нашою РЛС. Відповідно радіолокаційна станція знаходиться в середині кола разом з КВОІ (Рис. 2.1).

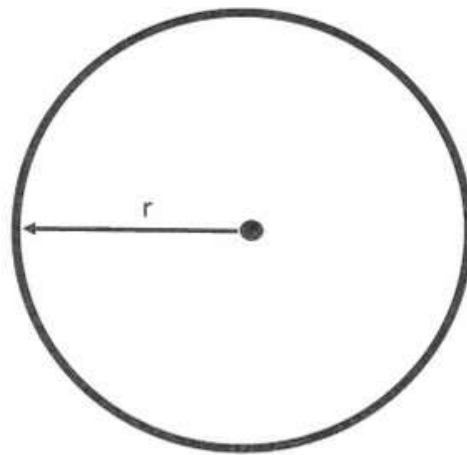


Рисунок 2.1 - Просторове розташування КВОІ разом з станцією спостереження за БПЛА

Коли в зону дії РЛС влітає БПЛА, то точка вльоту дрона зумовлюється часом вльоту та координатами точки початкової траєкторії.

В зону спостереження можуть вторгтися БПЛА різних видів та типів, вони будуть відрізнятися швидкістю висотою та іншими характеристиками. Саме тому не на останньому місці є формування класифікацій дронів, номенклатура яких постійно збільшується. Для більшого розуміння треба описати основні сформовані класифікації які використовує весь світ.

Для протидії БПЛА їх слід класифікувати за принципом використання, призначенням, габаритами та швидкістю.

Перша по важливості класифікація це принцип польоту дронів. Відповідно до схеми та принципу польоту БПЛА, слід розрізняти [3]:

- БПЛА літакового типу;
- БПЛА вертолітного типу.

За швидкістю польоту, що теж є одним з найважливіших параметрів в БПЛА, розрізняють такі класи:

- мало швидкісні БПЛА, що мають швидкості польоту до 150 км/год;
- середньо швидкісні БПЛА, що мають швидкості польоту від 150 до 450 км/год ;
- високошвидкісні БПЛА, що мають швидкості польоту від 450 до 880 км/год.

За призначенням прийнято поділяти БПЛА на такі класи:

БПЛА багаторазового застосування:

- розвідувальні БПЛА;
- розвідувально-ударні БПЛА;
- БПЛА-носії засобів озброєння;
- транспортні БПЛА;

БПЛА одноразового застосування:

- БПЛА - хибні цілі;
- БПЛА-камікадзе;
- БПЛА - перехоплювачі.

Відповідно до кількості одночасно застосовуваних апаратів БПЛА прийнято розділяти на:

- БПЛА поодинокого застосування.
- БПЛА групового застосування.

Найскладнішими для виявлення є на сам перед малогабаритні та мало швидкісні БПЛА, в першу чергу із-за своїх розмірів, але є і інші фактори.

Такі як:

- переміщення з використанням поворотів або зупинки;
- використання композитних або пластикових матеріалів в конструкції, що розсіюють електромагнітне випромінювання (ЕМВ);
- використання мобільних операторів або точок доступу до Wi-Fi.

Малі БПЛА також поділяють за призначенням на розвідувальні та ударні. Якщо ж ми говоримо про класифікацію малих БПЛА за масою, то слід розрізняти такі класи:

- нано БПЛА, що мають масу менше 0,25 кг, висоту польоту до 300 м, радіус дії до 1 км;
- мікро БПЛА, що мають масу до 5 кг, висоту польоту до 3 км, радіус дії до 10 км;
- міні БПЛА, що мають масу до 25 кг, висоту польоту до 3 км, радіус дії до 40 км.

Тепер почнемо моделювати клас та параметри дрону, який влетів в зону КВОІ. Для початку визначимо момент часу та координати точки вльоту в зону. Після цього виконаємо моделювання за наступною послідовністю:

- моделювання типу БПЛА;
- моделювання висоти польоту БПЛА;
- моделювання швидкості польоту БПЛА;
- Формування (моделювання) траєкторії польоту БПЛА в зоні КВОІ.

Цей алгоритм повторюється для кожного безпілотного літального апарату який влетів в зону КВОІ. З визначення часу вльоту до формування траєкторії.

Маючи всі ці данні ми можемо сформуванати таблицю, в якій для всіх передбачуваних типів БПЛА, записується швидкість польоту та діапазон допустимих висот(табл. 1).

Діапазон висот (км)	$\Delta A_i = A_i - A_{i=1} (A_0 = 0)$				
	$\Delta A_1$	$\Delta A_2$	$\Delta A_3$	.....	$\Delta A_m$
Тип БПЛА	Швидкість, км / год				
1	$S_{11}$	$S_{12}$	$S_{13}$	.....	$S_{1m}$
2	$S_{21}$	$S_{22}$	$S_{23}$	.....	$S_{2m}$
...	.....	.....	.....	.....	.....
k	$S_{k1}$	$S_{k2}$	$S_{k3}$	.....	$S_{km}$

Використовуючи апріорні данні про типи та параметри БПЛА ми формуємо таблицю. Перетин зовнішньої межі зони КВОІ має імовірний характер, що означає, що аналізуючи очікувану повітряну обстановку, можливо заздалегідь визначити ймовірність появи БПЛА, які зазначені в табл. 1. Ці можливості позначимо як  $c_1, c_2, \dots, c_k$  а умову при якій сукупність гаданих нами типів дронів утворюють повну групу подуї, позначимо як  $\sum_i c_i = 1$ .

Тепер нам потрібно сформувати випадкове число  $\xi_i$ , яке треба розподілити за рівномірним законом, з інтервалом  $(0, 1)$ , для моделювання типу дрону. Це випадкове число тепер потрібно порівняти з величинами  $C_{l-1}$  та  $C_l$ , що обумовлюються наступними виразами [3]:

$$C_{l-1} = \sum_{i=1}^{l-1} c_i, \quad C_l = \sum_{i=1}^l c_i, \quad (l = \overline{1, k}).$$

Далі вибирається той тип БПЛА  $l$ , для якого виконується умова

$$C_{l-1} \leq \xi_i < C_l.$$

Коли тип дрону обраний, у табл. 1 розшукується діапазон висот цього БПЛА ( $A_2 - A_m$ ). Знову задаємо ймовірність польоту безпілотного літального апарату в різних ділянках допустимого діапазону висот, і тим же методом знаходимо під діапазон висот  $\Delta A_l$ , який вже відповідає змодельованому БПЛА

Після визначення висоти польоту, треба вибрати з табл.1 швидкість дрону даного типу. Також швидкість можливо змоделювати за допомогою принципу випадкового вибору.

Змодельовавши тип, висоту та швидкість польоту БПЛА, формується третя координата. Ця нова інформація є висотою точки початку траєкторії, в зоні спостереження, а модуль вектору швидкості переміщення дрону є параметром його руху. Ці данні нам знадобляться для формування нових точок траєкторії польоту БПЛА, в зоні виявлення дронів.

Тепер можливо створити сукупність таблиць з використанням апріорної інформації з існуючої літератури за цими принципами моделювання. Кожна з таблиць буде відповідати різним умовам динамічної обстановки в зоні КВОІ. А використовувати їх можна адаптивним чином для моделювання різного ступеня важкості повітряних ситуацій.

### 3. РОЗРОБКА ПРОГРАМИ МОДЕЛЮВАННЯ ДИНАМІЧНОЇ ОБСТАНОВКИ

#### 3.1. Вибір середовища моделювання

Моделювання польотів літальних апаратів у зоні КВОІ, ми будемо виконувати у середовищі розробки Unity, із за зручності у використанні. Також Unity має вбудований 2D шаблон, за допомогою якого буде простіше моделювати роботу радіолокаційної станції. А писати програму будемо мовою C#, середовищі програмування Visual Studio.

Unity [12] - міжплатформне середовище розробки комп'ютерних ігор, розроблене американською компанією Unity Technologies. Unity дозволяє створювати програми, що працюють на більш ніж 25 різних платформах, що включають персональні комп'ютери, ігрові консолі, мобільні пристрої, інтернет-програми та інші. Випуск Unity відбувся у 2005 році і з того часу триває постійний розвиток.

Як правило, ігровий двигун надає безліч функціональних можливостей, що дозволяють їх задіяти в різних іграх, в які входять моделювання фізичних середовищ, карти нормалей, динамічні тіні та багато іншого. На відміну від багатьох ігрових движків, у Unity є дві основні переваги: наявність візуального середовища розробки та міжплатформова підтримка. Перший фактор включає не тільки інструментарій візуального моделювання, а й інтегроване середовище, ланцюжок складання, що спрямоване на підвищення продуктивності розробників, зокрема етапів створення прототипів та тестування. Під міжплатформною підтримкою надається не тільки місця розгортання (установка на персональному комп'ютері, на мобільному пристрої, консолі тощо), але й наявність інструментарію розробки (інтегроване середовище може використовуватись під Windows та Mac OS).

Як недоліки наводяться обмеження візуального редактора під час роботи з багатокомпонентними схемами, як у складних сценах візуальна робота утруднюється. Другим недоліком називається відсутність підтримки в

Unity посилянь на зовнішні бібліотеки, роботу з якими програмістам доводиться налаштовувати самостійно, і це також ускладнює командну роботу. Ще один недолік пов'язаний із використанням шаблонів екземплярів (англ. *prefabs*). З одного боку, ця концепція Unity пропонує гнучкий підхід візуального редагування об'єктів, але з іншого боку, редагування таких шаблонів є складним. Також, WebGL-версія движка, в силу специфіки своєї архітектури (трансляція коду з C# в C++ і далі JavaScript), має ряд невирішених проблем з продуктивністю, споживанням пам'яті і працездатністю на мобільних пристроях.

Проект в Unity ділиться на сцени (рівні) [12] - окремі файли, що містять свої ігрові світи зі своїм набором об'єктів, сценаріїв та налаштувань. Сцени можуть містити як, власне, об'єкти (моделі), так і порожні ігрові об'єкти — об'єкти, які мають моделі («пустушки»). Об'єкти, у свою чергу, містять набори компонентів, з якими і взаємодіють скрипти. Також об'єкти мають назву (в Unity допускається наявність двох і більше об'єктів з однаковими назвами), може бути тег (мітка) і шар, на якому він повинен відображатися. Так, у будь-якого об'єкта на сцені обов'язково присутній компонент *Transform* — він зберігає координати розташування, повороту і розмірів об'єкта по всіх трьох осях. Об'єкти з видимою геометрією також за замовчуванням мають компонент *Mesh Renderer*, що робить модель об'єкта видимою.

Поведінка ігрових об'єктів контролюється за допомогою компонентів (*Components*), які приєднуються до них. Незважаючи на те, що вбудовані компоненти Unity можуть бути дуже різнобічними, незабаром ви знайдете, що вам потрібно вийти за межі їхніх можливостей, щоб реалізувати власні особливості геймплею. Unity дозволяє створювати свої компоненти, використовуючи скрипти (Рис. 3.1.). Вони дозволяють активувати ігрові події, змінювати параметри компонентів, і відповідати на введення користувача в будь-який спосіб.

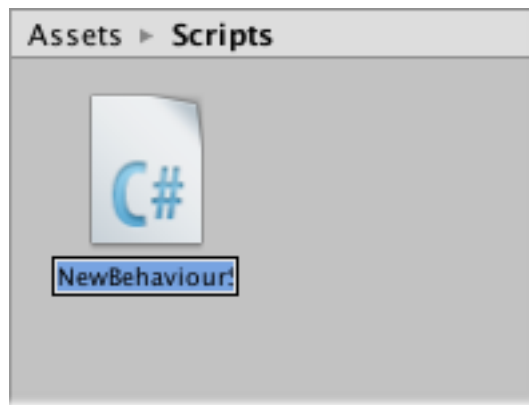


Рисунок 3.1 - Новий скрипт мовою C#.

Скрипт взаємодіє із внутрішніми механізмами Unity за рахунок створення класу, успадкованого від вбудованого класу, званого `MonoBehaviour`. Ви можете думати про клас як про свого роду план створення нового типу компонента, який може бути прикріплений до ігрового об'єкта. Щоразу, коли ви приєднуєте скриптовий компонент до ігрового об'єкта, створюється новий екземпляр об'єкта, визначений планом. Ім'я класу береться з імені, яке ви вказали під час створення файлу. Ім'я класу та ім'я файлу повинні бути однаковими, щоб скриптовий компонент міг бути приєднаний до ігрового об'єкта.

Основні речі, варті уваги, це дві функції, визначені всередині класу. Функція `Update` — це місце для розміщення коду, який оброблятиме оновлення кадру для ігрового об'єкта. Це може бути рух, спрацювання дій і реакція на введення користувача, в основному все, що повинно бути оброблено з часом в ігровому процесі. Щоб дозволити функції `Update` виконувати свою роботу, часто буває корисно ініціалізувати змінні, рахувати властивості та здійснити зв'язок з іншими ігровими об'єктами до того, як будуть здійснені будь-які дії. Функція `Start` буде викликана Unity до початку ігрового процесу (тобто до першого виклику функції `Update`) і це ідеальне місце для виконання ініціалізації.

Unity від початку підтримує мову програмування C#. C# (вимовляється як C-sharp) - це стандартна мова, схожа на Java або C++. Саме C# ми використовуватимемо для створення скриптів для нашого РЛС.

C# відноситься до сім'ї мов з C-подібним синтаксисом, їх синтаксис найбільш близький до C++ і Java. Мова має статичну типізацію, підтримує поліморфізм, навантаження операторів (у тому числі операторів явного та неявного приведення типу), делегати, атрибути, події, змінні, властивості, узагальнені типи та методи, ітератори, анонімні функції з підтримкою замикань, LINQ, винятки, коментарі у форматі XML.

XML — мова розмітки, що розширюється. Мова називається розширюваною, оскільки вона не фіксує розмітку, що використовується в документах: розробник вільний створити розмітку відповідно до потреб до конкретної області, будучи обмеженим лише синтаксичними правилами мови.

Переїнявши багато від своїх попередників - мов C++, Delphi, Модула, Smalltalk і, особливо, Java - C#, спираючись на практику їх використання, виключає деякі моделі, що зарекомендували себе проблематичні при розробці програмних систем, наприклад, C# на відміну від C++ не підтримує множинне спадкування класів (між тим допускається множинна реалізація інтерфейсів).

C# розроблявся як мова програмування прикладного рівня для CLR (загальнономовного виконуючого середовища) і, як такої, залежить, перш за все, від можливостей самої CLR. Це стосується, перш за все, системи типів C#, яка відображає BCL (стандартна бібліотека класів). Присутність або відсутність тих чи інших виразних особливостей мови диктується тим, чи конкретна мовна особливість може бути трансльована у відповідні конструкції CLR. Так, з розвитком CLR від версії 1.1 до 2.0 значно збагатився сам C#; подібної взаємодії слід очікувати і надалі (проте, ця закономірність була порушена з виходом C# 3.0, що є розширення мови, що не спираються на розширення платформи .NET). CLR надає C#, як і всім іншим .NET-

орієнтованим мовам, багато можливостей, яких позбавлені «класичні» мови програмування. Наприклад, складання сміття не реалізована в самому C#, а проводиться CLR для програм, написаних на C# так само, як це робиться для програм на VB.NET, J# та ін.

Як середовище програмування для мови C# використовуватиметься Visual Studio.

Visual Studio — це стартова площадка для написання, налагодження та складання коду, а також подальшої публікації програм. Крім стандартного редактора і відладчика, які є в більшості середовищ IDE (інтегрованих середовищах розробки), Visual Studio включає компілятори, засоби автозавершення коду, графічні конструктори і багато інших функцій для поліпшення процесу розробки.

Visual Studio [14] включає редактор вихідного коду з підтримкою технології IntelliSense і можливістю найпростішого рефакторингу коду. Вбудований налагоджувач може працювати як відладчик рівня вихідного коду, так і відладчик машинного рівня. Інші вбудовані інструменти включають редактор форм для спрощення створення графічного інтерфейсу програми, веб-редактор, дизайнер класів і дизайнер схеми бази даних. Visual Studio дозволяє створювати та підключати сторонні доповнення (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (як, наприклад, Subversion та Visual SourceSafe), додавання нових наборів інструментів (наприклад, для редагування та візуального проектування) коду предметно-орієнтованими мовами програмування) або інструментами для інших аспектів процесу розробки програмного забезпечення (наприклад, клієнт Team Explorer для роботи з Team Foundation Server).

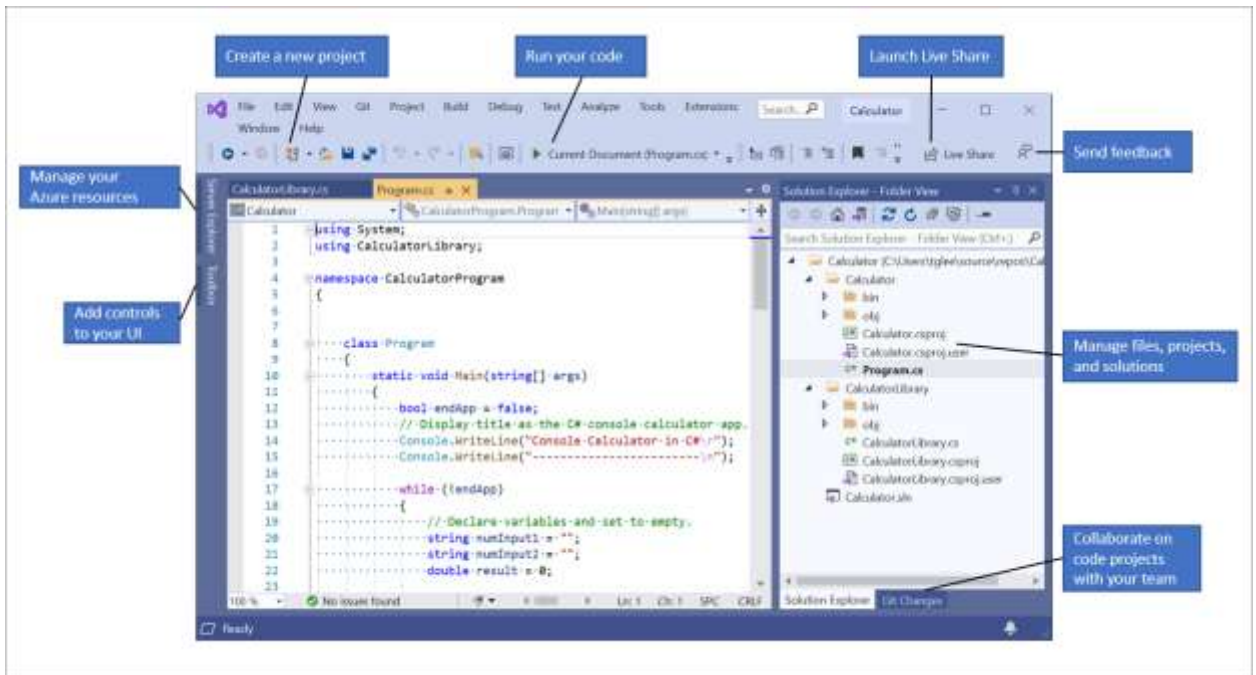


Рисунок 3.2 - Visual Studio із відкритим проектом.

На рисунку 3.2. представлено середовище Visual Studio з відкритим проектом та підказки з основних вікон та функціональних можливостей.

Праворуч у верхньому куті вікна браузера рішень можна переглядати файли коду, переміщатися по них і керувати ними. Оглядач рішень дозволяє впорядкувати код шляхом об'єднання файлів у рішення та проекти.

У центральному вікні редактора, з яким працюють найдовше, відображається вміст файлу. У вікні редактора можна вносити зміни в код або розробляти інтерфейс користувача, наприклад вікно з кнопками або текстові поля.

Вікно змін Git у нижньому куті праворуч дозволяє відстежувати робочі елементи та надавати спільний доступ до коду за допомогою Git, GitHub або інших технологій керування версіями.

### 3.2. Моделювання роботи індикатора кругового огляду

Створюємо новий 2D проект. Перед тим як моделювати польоти БПЛА нам спершу потрібно створити індикатор кругового огляду, та інші об'єкти на сцені такі як скрипти та збірники в яких будуть зображення наших майбутніх дронів.

Для початку нам потрібно створити площину, яку ми будемо використовувати як задній фон. На ній відобразимо зображення індикатора кругового огляду, яке знайдемо в інтернеті. Для цього натискаємо праву кнопку миші, після чого у віконці виберемо 3D об'єкт площини. Редакуємо розмір та місце розташування площини так щоб вона перпендикулярно камері. Після чого копіюємо зображення індикатора у папку з нашими ресурсами для проекту, тепер нам можна приєднати зображення індикатора до площини, що ми і робимо. Результат показаний на рис 3.3.

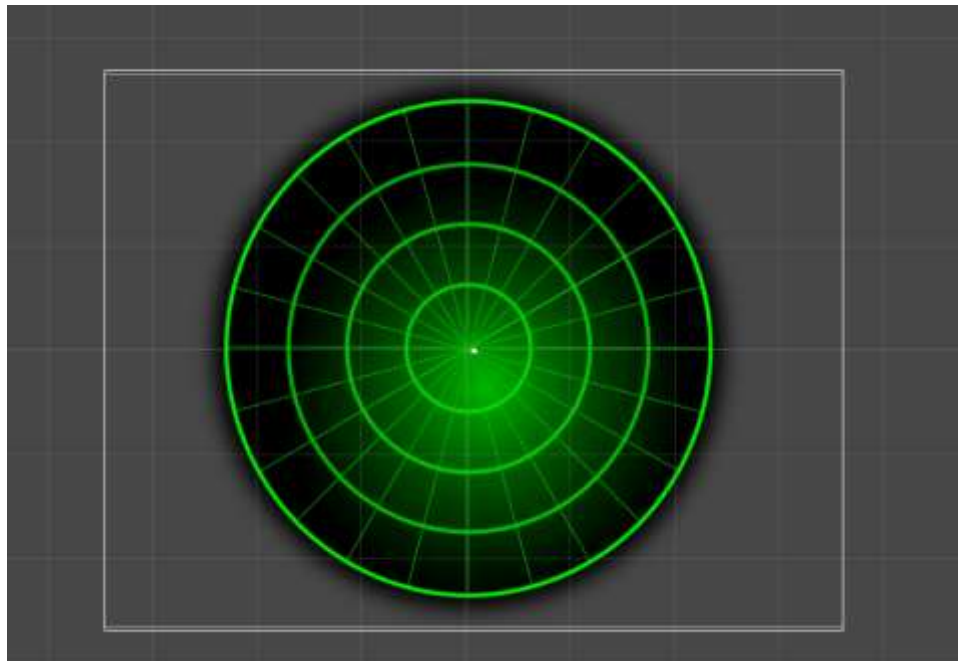


Рисунок 3.3 - Зображення індикатора кругового огляду.

Наступним предметом, який ми створимо, буде порожній об'єкт, який ми поставимо в центр нашого індикатора, він буде точкою, якої прагнуть наші дрони, з деяким розкидом, який ми поставимо в майбутньому, за

допомогою скриптів. Це нам необхідно для того, щоб деякі з наших об'єктів, а саме БПЛА літакового типу при створенні не повертали себе на 180 градусів і автоматично віддалялися. А все-таки проходили випадковою траєкторією радіолокаційної станції.

Тепер створимо ще один порожній об'єкт, він буде нашим джерелом БПЛА. Створюємо йому скрипт мовою C#, і приєднуємо його до порожнього об'єкту. Параметри спавна БПЛА, як і інші параметри, ми будемо задавати в кодї скрипту. Створимо ще один такий самий об'єкт із скриптом для другого виду БПЛА, вертолітного. Ми їх поділяємо на різні об'єкти для того, щоб у майбутньому ми могли самостійно контролювати кількість БПЛА кожного виду окремо, як і інші параметри. Всі ці об'єкти ми створюємо у віконці «ієрархія» (Рис 3.4).

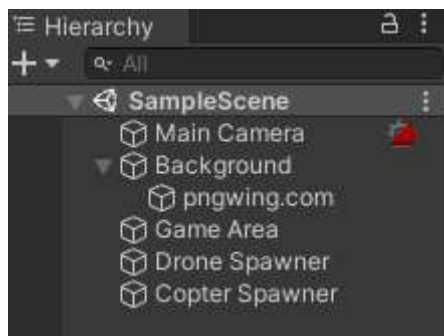


Рисунок 3.4 - Вікно «ієрархія».

Насамкінець нам потрібні ще два порожніх об'єкта, до яких ми приєднаємо зображення нашого дрона. Точніше те, як він буде виглядати на нашому індикаторі кругового огляду. У нашому випадку це буде червона точка, щоб вони не зливались із зеленим фоном нашого РЛС. Створюємо скрипт і для цих об'єктів, там будуть прописані завдання для кожного з дронів при їх появі в програмі. Ми створюємо два об'єкта тому що у дронів будуть два різні скрипта один для літакового типу а інший для вертолітного.

Починаємо ми з того що копіюємо зображення червоної точки в нашу папку з ресурсами після чого приєднуємо її до обох порожніх об'єктів. Тепер ми бачимо що точки з'явилися на головному екрані, для того щоб вони зникли нам потрібно перенести пусті об'єкти в папку з ресурсами, так вони

не будуть заважати роботі програми та при цьому будуть працювати як треба. Тепер наша папка з ресурсами буде мати 2 об'єкти для відображення дронів, 4 скрипти (2 для спавну дронів та 2 для переміщення дронів) як і 2 зображення індикатора і червоної точки (Рис 3.5).

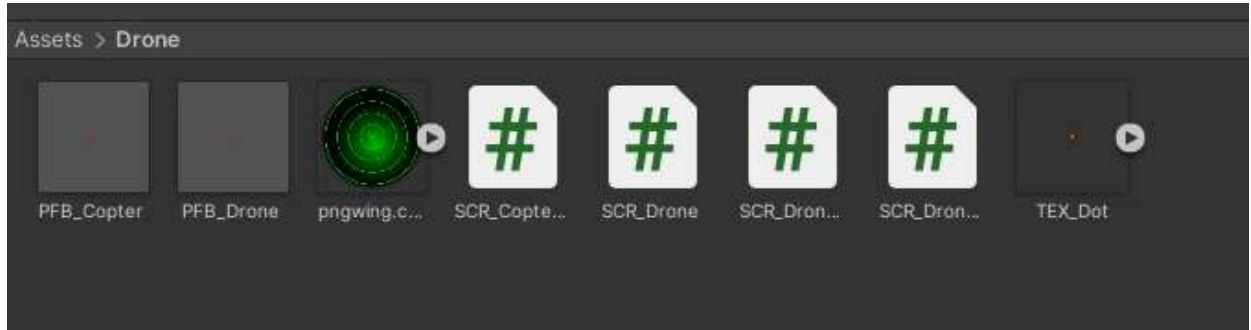


Рисунок 3.5 - Папка з ресурсами.

Коли створення проекту закінчено і файли приєднані один до одного, настав час писати код Для початку заходимо в скрипт SCR\_DroneSpawner, де ми будемо створювати нових дронів та їх параметри.

### 3.2.1 Код спавнера дронів

Спочатку напишемо ті бібліотеки які будемо використовувати у нашому коді. У нашому випадку нам нічого дописувати не потрібно, загальна (System.Collections) та основна бібліотека Unity (UnityEngine) були описані автоматично. Далі наведено фрагмент коду з описом бібліотек.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

Тепер створюймо компоненти в які ми будемо заносити інформацію про кількість дронів, їх швидкість висоту і тд. Весь подальший код ми будемо записувати в класі SCR\_DroneSpawner. Далі наведено фрагмент коду з створенням компонентів для класу SCR\_DroneSpawner.

```
public class SCR_DroneSpawner : MonoBehaviour
{
```

```
public GameObject game_area;  
public GameObject drone_prefab;  
public int drone_count1 = 0;  
public int drone_limit1 = 5;  
public int drone_per_frame = 1;  
  
public float spawn_circle_radius = 25.0f;  
public float death_circle_radius = 26.0f;  
  
public float fastest_speed = 9.8f;  
public float slowest_speed = 1.5f;  
  
public float max_altitude = 500.0f;  
public float min_altitude = 5.0f;
```

У першому рядку записуємо `game_area` в наш скрипт для того щоб він мав інформацію про місцезнаходження нашого об'єкту. У наступному рядку створюємо шаблон з якого ми будемо брати зображення дронів. `drone_count1` відповідає за кількість дронів у даний момент, цей компонент не можна редагувати під час роботи програми. `drone_limit1` відповідає за максимальну кількість дронів, саме цей компонент буде використовуватися для зміни кількості дронів. `drone_per_frame` відповідає за те скільки безпілотників будуть спавнитися кожний фрейм.

Для того щоб наші дрони спавнились по краям індикатора кругового огляду нам потрібно створити коло краї якого ми будемо використовувати як точки створення безпілотників. Для цього нам потрібні два компоненти, `spawn_circle_radius` та `death_circle_radius`. В них буде записаний радіус кіл, одне коло буде використовуватися для створення дронів а інше для видалення дронів які вийшли за рамки індикатора. Для того щоб при створенні дрони не знищувалися автоматично ми зробимо радіус другого кола трохи більше.

Редагувати швидкість БПЛА ми будемо за допомогою ще двох компонентів. В компонент `fastest_speed` ми напишемо максимальну швидкість, а в компонент `slowest_speed` ми напишемо мінімальну швидкість. В майбутньому програма буде випадково вибирати швидкість для кожного дрона індивідуально, між цими двома числами. По класифікації дронів ми знаємо що максимальна швидкість польоту може бути 980 км/год, поставимо 9.8 в максимальну швидкість. В мінімальну швидкість для дронів літакового типу ми маємо 150 до 400 км/год, що являється середньо швидкісним БПЛА, ставимо в `slowest_speed` значення 1.5.

Останні два компоненти це мінімальна і максимальна висота БПЛА (`max_altitude` та `min_altitude`). Програма таким же чином буде випадково вибирати висоту для кожного дрона. Середній робочий діапазон висот у літакових БПЛА 50-5000 м, тому ставимо 5 - 500 у відповідні компоненти. Рисунок 3.6. показує створені нами компоненти, та їх значення.

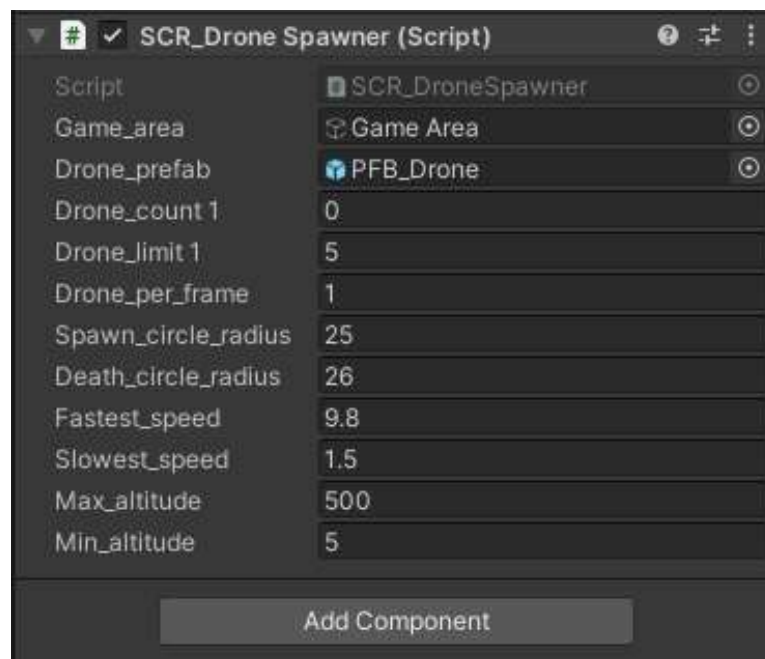


Рисунок 3.6 - Компоненти скрипта `SCR_DroneSpawner`.

Далі наведено фрагмент коду з функціями `Start()` та `Update()`.

```
void Start()
{
```

```
}
```

```
void Update()
{
    MaintainPopulation();
}
```

Функція Start() у нас буде пустою. А в функцію Update() ми напишемо майбутню функцію MaintainPopulation() в якій ми будемо створювати нові БПЛА при цьому не підвищуючи максимальний ліміт який задали раніше. Далі наведено фрагмент коду з функцією MaintainPopulation().

```
void MaintainPopulation()
{
    if (drone_count1 < drone_limit1)
    {
        for (int i = 0; i < drone_per_frame; i++)
        {
            Vector3 position = GetRandomPosition();
            SCR_Drone drone_script = AddDrone(position);
            drone_script.transform.Rotate(Vector3.forward * Random.Range(-45.0f,
45.0f));
        }
    }
}
```

В функції MaintainPopulation() ми контролюємо кількість дронів в зоні, якщо кількість БПЛА менше ніж drone\_limit1 то функція спочатку знаходить випадкову позицію в зоні, створює дрона, після чого повертає його з середини зони, куди він був повернутий до цього в ліво або вправо під випадковим кутом до 45 градусів. За допомогою Random.Range(). Це

необхідно тільки для БПЛА літакового типу, для того щоб вони летіли в зоні індикатора але не по одній і тій самій траєкторії. Далі наведено фрагмент коду з функцією `GetRandomPosition()`.

```
Vector3 GetRandomPosition()
{
    Vector3 position = Random.insideUnitCircle.normalized;

    position *= spawn_circle_radius;
    position += game_area.transform.position;

    return position;
}
```

Функція `GetRandomPosition` обирає випадкову точку для спавна дрона, використовуючи 2D коло, по краям зони індикатора. Це стане можливим за допомогою `Random.insideUnitCircle.normalized`. Якщо ми просто напишемо `Random.insideUnitCircle` то наші БПЛА будуть з'являтися не тільки по краям а і по середині кола. Але якщо ми додадим `normalized`, то дрони будуть з'являтися лише по краям кола(Рис. 3.7).

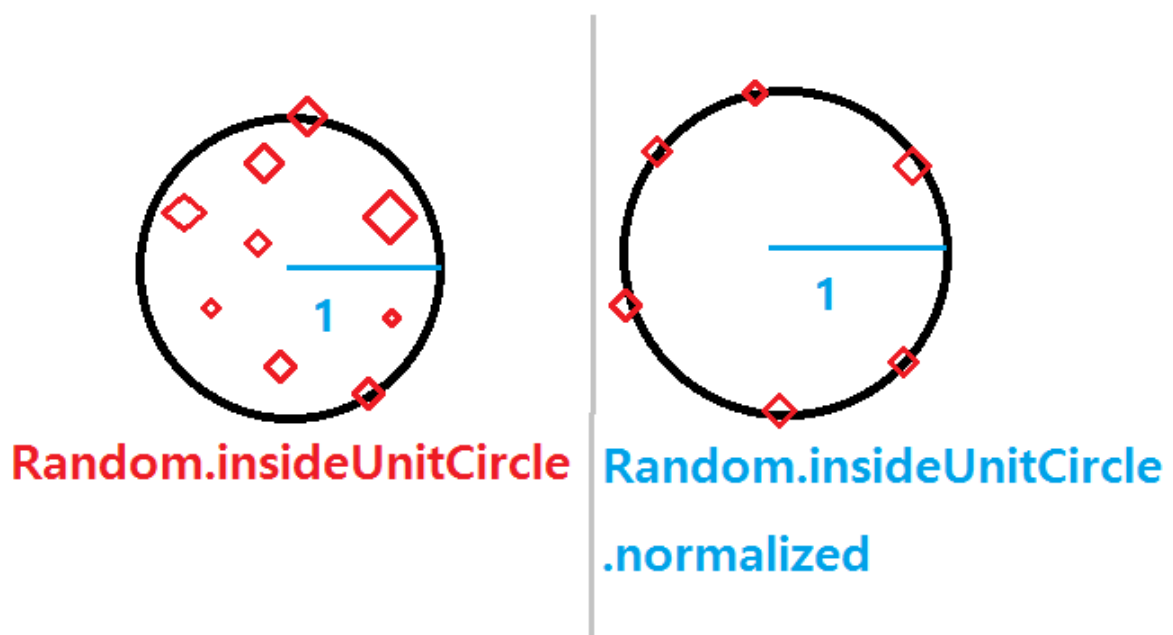


Рисунок 3.7 - Спавн з використанням `normalized`.

Внизу наведено фрагмент коду з функцією AddDrone().

```
SCR_Drone AddDrone(Vector3 position)
{
    drone_count1 += 1;
    GameObject new_drone = Instantiate(
        drone_prefab,
        position,
        Quaternion.FromToRotation(Vector3.up, (game_area.transform.position -
position)),
        gameObject.transform
    );

    SCR_Drone drone_script = new_drone.GetComponent<SCR_Drone>();
    drone_script.drone_spawner = this;
    drone_script.game_area = game_area;
    drone_script.speed = Random.Range(slowest_speed, fastest_speed);
    drone_script.altitude = Random.Range(min_altitude, max_altitude);

    return drone_script;
}
}
```

Останньою функцією для скрипта створення дронів літакового типу буде AddDrone. Тут буде код для створення дронів та задання основних параметрів цих дронів. Спочатку ми додаємо +1 до кількості дронів, для того щоб не створювати занадто багато безпілотників. Потім ми приєднуємо до нового БПЛА зображення з папки з ресурсами, позицію в зоні а також повертаємо його щоб він був повернутий до центру зони.

Після створення БПЛА ми можемо задавати основні параметри. За допомогою Random.Range задаємо швидкість та висоту нашого дрона, після

чого повертаємо всі ці значення в SCR\_Drone для того щоб інший скрипт мав доступ до цих даних.

Можемо переходити до скрипту спавнера дронів вертолітного типу.

### 3.2.2 Код спавнера коптерів

Далі наведено фрагмент коду з описом бібліотек та з створенням компонентів для класу SCR\_CopterSpawner.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SCR_CopterSpawner : MonoBehaviour
{
    public GameObject game_area2;
    public GameObject drone_prefab;

    public int copter_count = 0;
    public int copter_limit = 10;

    public int drone_per_frame = 1;

    public float spawn_circle_radius = 25.0f;
    public float death_circle_radius = 26.0f;

    public float fastest_speed = 2.0f;
    public float slowest_speed = 0.5f;

    public float max_altitude = 100.0f;
    public float min_altitude = 1.0f;
```

Задаємо ті ж самі компоненти змінюючи тільки числа в них які мають більше сенсу для коптерних дронів (зменшення максимальної швидкості). Для задання мінімальної та максимальної швидкості знову використаємо класифікації дронів, на цей час ми візьмемо швидкість мало швидкісних БПЛА, максимальна швидкість польоту яких не більше 200 км/год, за мінімум ми візьмемо 50 км/год.

Висота польоту теж буде менше ніж у БПЛА літакового типу. Середній робочий діапазон висот у вертолітних БПЛА 10-1000 м, по цим даним змінюємо значення висот. Рисунок 3.8. показує створені нами компоненти, та їх значення.

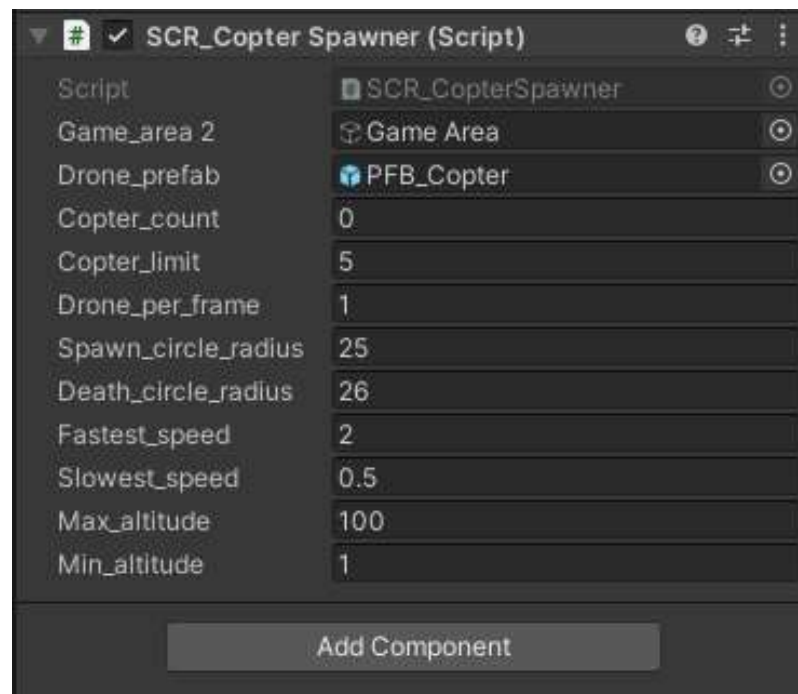


Рисунок 3.8 - Компоненти скрипта SCR\_CopterSpawner.

Далі наведено фрагмент коду з функціями Start(), Update() та MaintainPopulation2().

```
void Start()
{
}
}
```

```

void Update()
{
    MaintainPopulation2();
}

void MaintainPopulation2()
{
    if (copter_count < copter_limit)
    {
        for (int i = 0; i < drone_per_frame; i++)
        {
            Vector3 position = GetRandomPosition();
            SCR_Drone2 drone_script2 = AddCopter(position);

        }
    }
}

```

Також видалимо повертання дрона на 45 градусів, тому що змінювати шлях яким буде летіти коптер на цей раз буде скрипт SCR\_Drone2. Саме там ми будемо змінювати траєкторію. Далі наведено фрагмент коду з функціями GetRandomPosition() та AddCopter().

```

Vector3 GetRandomPosition()
{
    Vector3 position = Random.insideUnitCircle.normalized;
    position *= spawn_circle_radius;
    position += game_area2.transform.position;
    return position;
}

SCR_Drone2 AddCopter(Vector3 position)

```

```

{
    copter_count += 1;
    GameObject new_drone = Instantiate(
        drone_prefab,
        position,
        Quaternion.FromToRotation(Vector3.up, (game_area2.transform.position -
position)),
        gameObject.transform
    );

    SCR_Drone2 drone_script2 = new_drone.GetComponent<SCR_Drone2>();
    drone_script2.copter_spawner = this;
    drone_script2.game_area2 = game_area2;
    drone_script2.speed = Random.Range(slowest_speed, fastest_speed);
    drone_script2.altitude = Random.Range(min_altitude, max_altitude);

    return drone_script2;
}
}

```

Решта коду залишається такою ж як в `SCR_DroneSpawner`. Ми створюємо випадкову позицію на колі. Робимо наш БПЛА, та задаємо йому параметри, зовнішній вигляд, кут погляду, швидкість та висоту.

### 3.3. Моделювання польотів БПЛА

Коли написання коду для задання параметрів та створення БПЛА закінчено, необхідно переходити до моделювання польотів цих дронів. Почнемо з коду для безпілотників літакового типу.

### 3.3.1. Код літакових дронів

Далі наведено фрагмент коду з описом бібліотек та з створенням компонентів для класу SCR\_Drone.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SCR_Drone : MonoBehaviour
{
    public SCR_DroneSpawner drone_spawner;
    public GameObject game_area;

    public float speed;
    public float altitude;
```

Перші дві строки у нашому класі з'єднують цей скрипт зі скриптом SCR\_DroneSpawner та з зоною індикатора game\_area. Потім ми робимо значення швидкості та висоти публічними для того щоб ми могли бачити їх в іконці скрипта. На рисунку 3.9. показані значення які програма створила для одного з дронів.

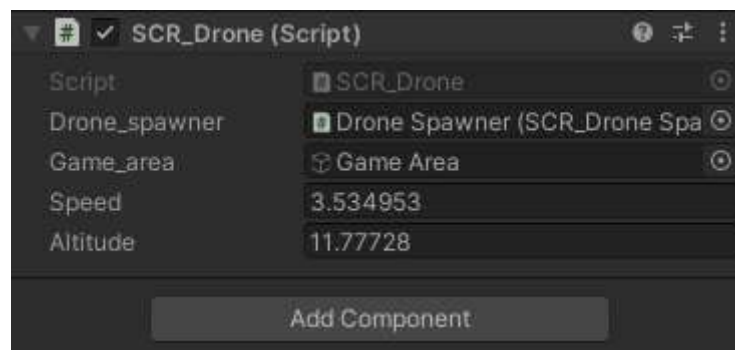


Рисунок 3.9 - Швидкість та висота одного з дронів.

Далі наведено фрагмент коду з функціями Update() та Move().

```
void Update()
```

```

{
    Move();
}

void Move()
{
    transform.position += transform.up * (Time.deltaTime * speed);
    float distance = Vector3.Distance(transform.position,
game_area.transform.position);
    if (distance > drone_spawner.death_circle_radius)
    {
        RemoveDrone();
    }
}

```

В цій функції ми рухаємо дрон вперед один раз за фрейм в залежності від швидкості. Напрявлення його польоту ми задавали ще в SCR\_DroneSpawne, тому все що нам тепер залишається це написати код на видалення БПЛА якщо він залетить за межі нашого кола смерті. Для цього ми створимо ще одну функцію RemoveDrone(). Далі наведено фрагмент коду з функцією RemoveDrone().

```

void RemoveDrone()
{
    Destroy(gameObject);
    drone_spawner.drone_count1 -= 1;
}
}

```

Ця функція зменшує значення drone\_count1 на один, та знищує об'єкт. Переходимо до коптерних дронів.

### 3.3.2 Код коптеров

Далі наведено фрагмент коду з описом бібліотек та з створенням компонентів для класу SCR\_Drone2.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class SCR_Drone2 : MonoBehaviour
{
    public float moveDuration = 4f;
    public float pauseDuration = 2f;
    public float timer;
    public bool paused;
    private Vector3 movementDirection;
    private Vector3 movementPerSecond;

    public SCR_CopterSpawner copter_spawner;
    public GameObject game_area2;

    public float speed;
    public float altitude;
```

Основна різниця між БПЛА літакового та вертолітного типу це можливість останніх зупинятися в польоті та змінювати свою траєкторію. Для моделювання польотів таких БПЛА нам потрібен таймер, він буде змінювати наш об'єкт з стану руху до стану зупинки і навпаки.

Для початку напишемо два компоненти moveDuration який відповідає за те скільки секунд дрон буде в стані руху. А також pauseDuration який буде зберігати час який дрон буде стояти на місці. Створюємо самий таймер, та робимо його публічним щоб ми бачили скільки часу залишилось між

станами. Нашим умовним виразом буде `paused`. `movementDirection` відповідає за зміну напрямлення БПЛА, а `movementPerSecond` за швидкість. Все інше залишаємо незмінним. На рисунку 3.10. показані значення які програма створила для одного з дронів.

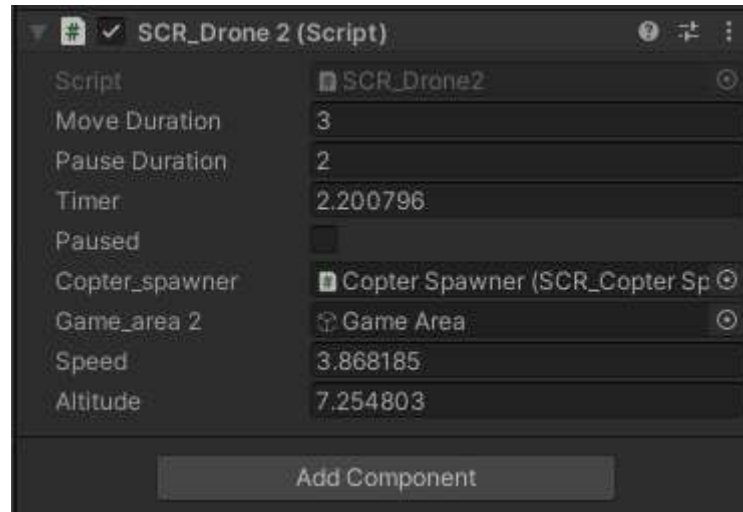


Рисунок 3.10 - Усі компоненти одного з вертолітних дронів.

Далі наведено фрагмент коду з функцією `Start()`.

```
void Start()
{
    timer = moveDuration; //to get it moving from the start
    paused = false;
    FindNewDirection();
}
```

Для того щоб БПЛА рухався з самого початку роботи програми в функції `Start()` вмикаємо таймер та викликаємо функцію `FindNewDirection()` для вибору траєкторії та руху. Далі наведено фрагмент коду з функцією `Update()`.

```
void Update()
{
    timer -= Time.deltaTime;
    if (timer <= 0)
    {
```

```

    if (paused)
    {
        timer = moveDuration;
        FindNewDirection();
        paused = false;
    }
    else
    {
        timer = pauseDuration;
        paused = true;
    }

    return;
}

if (!paused)
{
    transform.position = new Vector3(transform.position.x +
(movementPerSecond.x * Time.deltaTime),
    transform.position.y + (movementPerSecond.y * Time.deltaTime));
}
Move2();
}

```

В функції Update() ми пишемо код нашого таймера, для того щоб він працював на протязі всього часу роботи програми. Якщо паузи немає то таймер буде відраховувати moveDuration і рухати дрон. Коли пауза буде увімкнута таймер відраховує pauseDuration і в кінці вимикає паузу. Функція Move2() буде лише видаляти дрон якщо той залетить за межі кола смерті. Далі наведено фрагмент коду з функціями Move2(), RemoveDrone() та FindNewDirection().

```

void Move2()
{
    float distance = Vector3.Distance(transform.position,
game_area2.transform.position);
    if (distance > copter_spawner.death_circle_radius)
    {
        RemoveDrone();
    }
}

void RemoveDrone()
{
    Destroy(gameObject);
    copter_spawner.copter_count -= 1;
}

void FindNewDirection()
{
    movementDirection = new Vector3(Random.Range(-1.0f, 1.0f),
Random.Range(-1.0f, 1.0f)).normalized;
    movementPerSecond = movementDirection * speed;
}
}

```

FindNewDirection() створює вектор випадкового напрямку з величиною 1, після чого після чого задає нашому дрону швидкість.

Зберігаємо готовий код, після чого перевіряємо працездатність програми. Натискаймо кнопку «Плей», це почне роботу симуляції. Якщо потрібно, для зупинки програми натиснемо кнопку «Паузи», яка по праву сторону від кнопки «Плей». Натиснувши «Плей» другий раз припинить роботу симуляції (Рис.3.11.).



Рис. 3.11 - Кнопки «Плей» та «Паузи».

Зліва в іконці ієрархії вибираємо вид дрону, після чого справа з'явиться іконка скрипта, там можливо міняти кількість дронів у реальному часі, як і їх швидкість та висоту (швидкість та висота змінюється тільки у нових БПЛА).

Якщо ми натиснемо кнопку «Плей», або в віконці File натиснемо кнопку Build And Run, то почнеться робота симуляція і програма запрацює(Рис.3.12.).

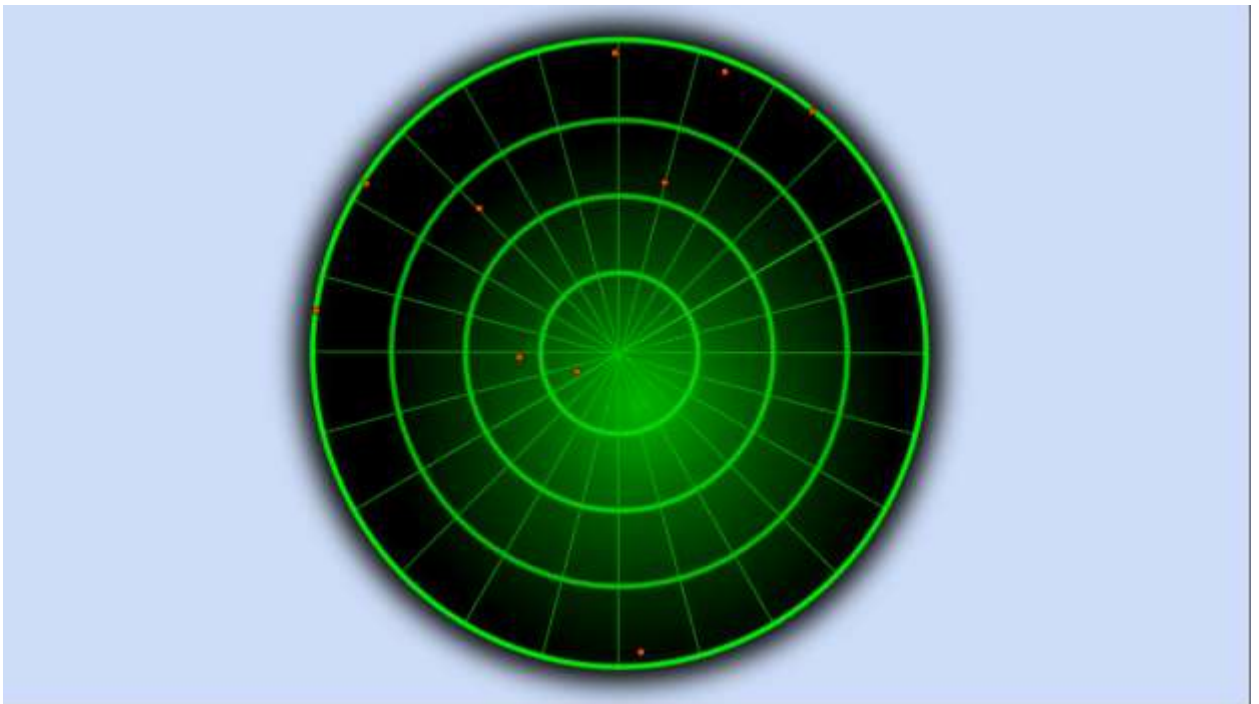


Рисунок 3.12 - Відображення повітряної обстановки на ІКО

## ВИСНОВКИ

Метою даної магістерської кваліфікаційної роботи є моделювання польоту літальних апаратів у зоні критично важливих об'єктів інфраструктури для вирішення завдань тестування існуючих засобів виявлення БПЛА з використанням індикатора кругового огляду РЛС шляхом симуляції польотів БПЛА.

Методом виконання роботи є математичне моделювання, розробка математичних моделей, реалізація моделей з використанням мови програмування C#, в середовищі розробки Unity. Unity забезпечує широкі можливості для створення моделей різних видів для різних областей використання. Середовище програмування - Visual Studio.

В роботі виконано огляд і аналіз існуючих безпілотних літальних апаратів, а також відомих методів виявлення БПЛА. Розроблено математичні моделі вторгнення БПЛА різних видів і типів в зону КВОІ, а також параметрів польотів БПЛА- висоти, швидкості, напрямку польоту. З використанням індикатора кругового огляду радіолокаційної станції створена модель динамічної повітряної обстановки в у зоні критично важливого об'єкту інфраструктури. Модель включає випадкові потоки літальних апаратів, які перетинають зовнішню межу зони, що відноситься до критично важливого об'єкту інфраструктури.

Реалізовано моделювання польотів двох видів БПЛА, літакового та вертолітного. Кожен цих з видів БПЛА має свою максимальну та мінімальну швидкість, висоту, траєкторію руху. Усі параметри польоту цілій можливо редагувати, що забезпечує функціональність і ефективність розробленої програми. Також редагуванню підлягає окремо і кількість БПЛА кожного з видів.

Отримані в роботі результати можуть бути використані на практиці для створення сукупності таблиць, що відповідають різним умовам і режимам інтенсивності динамічної повітряної обстановки в зоні КВОІ, які можуть

використовуватися при моделюванні повітряних ситуацій різного ступеня складності.

Область використання отриманих результатів– тестування існуючих засобів виявлення БПЛА з використанням індикатора кругового огляду РЛС шляхом симуляції польотів БПЛА.

## ПЕРЕЛІК ПОСИЛАНЬ

1. А.О. Пархомов. Радиолокационные методы обнаружения беспилотных летательных аппаратов // 24-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у ХХІ столітті». Зб. матеріалів форуму. Т. 3. – Харків: ХНУРЕ. 2021. – с. 30 – 32.
2. А. Бойко. Основні конструкції безпілотників - Класифікація безпілотників. 2019. <http://robotrends.ru/robopedia/osnovnye-konstrukcii-bespilotnikov>. Дата звернення 15.09.2021 г.
3. Карташов В.М., Олейников В.Н., Шейко С.А., Бабкин С.И., Корытцев И.В., Зубков О.В., Анохин М.А. Моделирование класса ЛА. С. 1-2. Department of Media Engineering and Information Radio Electronic Systems Kharkiv National University of Radio Electronics Kharkiv, Ukraine [http://dspace.univd.edu.ua/xmlui/bitstream/handle/123456789/10496/Aviatsiia\\_Pr\\_omyslovist\\_Suspilstvo\\_1.09\\_2021.pdf](http://dspace.univd.edu.ua/xmlui/bitstream/handle/123456789/10496/Aviatsiia_Pr_omyslovist_Suspilstvo_1.09_2021.pdf). Дата звернення 23.09.2021 г.
4. В.М. Карташов, В.Н. Олейников, С.А. Шейко. Особенности обнаружения и распознавания малых беспилотных летательных аппаратов // Радиотехника. 2018. Вып. 195 – с.235 – 242.
5. В.М. Карташов, И.В. Корытцев, С.А. Шейко, В.Н. Олейников, О.В. Зубков, С.И. Бабкин. Оптико-электронные методы обнаружения воздушных 82 объектов и измерения их координат // Радиотехника: Всеукр. межвед. науч.-техн. сб. – 2020. – Вып. 202. – С. 153 – 159.
6. Корытцев И.В., Зубков О.В., Анохин М.А., В. М. Карташов. Обнаружение БПЛА на фоне акустических шумов и помех, с. 68-70. <https://journals.indexcopernicus.com/search/article?articleId=2139491>. Дата звернення 25.10.2021 г.
7. Zelnio A.M. Detection of small aircraft using an acoustic array. Thesis. B.S. / A.M. Zelnio. – Electrical Engineering, Wright State University. - 2007. – 55 p.
8. И.В. Корытцев, С.А. Шейко, В.М. Карташов, О.В. Зубков, В.Н. Олейников, С.И. Бабкин, И.С. Селезнев. Обработка сигналов при пеленгации и

определении дальности до среднеразмерных БПЛА в радиолокационном диапазоне // Радиотехника: Всеукр. межвед. науч.-техн. сб. – 2020. – Вып. 202. – С. 125 – 134.

9. Bisio I., Garibotto C., Lavagetto F., Sciarrone A., Zappatore S. Blind Detection: Advanced Techniques for WiFi-Based Drone Surveillance. IEEE Trans. Veh. Technol. 2019;68:938–946. doi: 10.1109/TVT.2018.2884767. Дата звернення 16.10.2021 г.

10. М.А. Беляева. Моделювання систем. 2012, С. 43-50. <http://simulation.su/uploads/files/default/2012-belyaeva-lekcii-part1.pdf>. Дата звернення 11.11.2021 г.

11. А. И. Леонов и др. Моделирование в радиолокации.; Под ред. А. И. Леонова. — М.: Сов. радио, 1979. - 264 с

12. Т.Г. Ли и др. Введение в интегрированную среду разработки Visual Studio | C#, 2021. <https://docs.microsoft.com/ru-ru/visualstudio/get-started/csharp/visual-studio-ide?view=vs-2022>. Дата звернення 20.11.2021 г.

13. Карташов В.М., Олейников В.Н., Шейко С.А., Бабкин С.И., Корытцев И.В., Зубков О.В., Анохин М.А. Информационные характеристики звуковых сигналов малых беспилотных летательных аппаратов// Радиотехника. Всеукр. Межвед. Науч.-техн. Збірник. Вып. 191. - Харьков, 2017. - С. 181-187.

14. Ю.Г Даник, І.В.Пулеко, М.В.Бугайов. Виявлення безпілотник літальних апаратів на основі аналізу акустичних та радіолокаційних сигналів//Вісник ЖДТУ. 2014, № 4 (71). С.71- 80.

15. Samaras S., Diamantidou E., Ataloglou D., Sakellariou N., Vafeiadis A., Magoulianitis V., Lalas A., Dimou A., Zarpalas D., Votis K., et al. Deep Learning on Multi Sensor Data for Counter UAV Applications—A Systematic Review. Sensors. 2019;19:4837. doi: 10.3390/s19224837.

16. Цепляева Т.П., Поздышева Е. М., Поштаренко А. Г. Анализ применения беспилотных комплексов. [Текст] Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ». Харьков.-

[https://www.khai.edu/csp/portal//Archiv/OIKIT39/p\\_149-154.pdf](https://www.khai.edu/csp/portal//Archiv/OIKIT39/p_149-154.pdf). Дата звернення 22.11.2021 г.

17. Корольков В.А. Автоматизированные акустические и оптоэлектронные комплексы и системы для экологического и метеорологического мониторинга атмосферы. Дисс. ... д.т.н. по специальности 05.11.13 – Приборы и методы контроля природной среды, веществ, материалов и изделий. Научн. конс. – д.т.н., проф. Тихомиров А.А. –Томск, 2017. – 471 с.

18. Kartashov V.M., Oleynikov V.N, Zubkov O.V., Korytsev I.V., Babkin S. I., Sheiko S.A., Kolendovskaya M.M. Spatial-temporal Processing of acoustic Signals of Unmanned Aerial Vehicles/ Telecommunications and Radio Engineering. –New York. – 2020. – Vol. 79, №9. – P.769-780.

19. V. Kartashov, V. Oleynikov , I. Koryttsev, S. Sheiko, O. Zubkov, S. Babkin. Processing of Wide Band Acoustic Signals During Detection of Unmanned Aerial Vehicles // 2020 IEEE Ukrainian Microwave Week (UkrMW). Kharkiv, Ukraine, September 21 - 25, 2020. Volume 1 on 2020 IEEE 12th International Conference on Antenna Theory and Techniques (ICATT). pp. 35-39.

20. V. Kartashov, V. Oleynikov, O. Zubkov, S. Sheiko. Optical detection of unmanned air vehicles on a video stream in a real-time // The Fourth International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo'2019), 9–13 September 2019, Odessa, Ukraine, 4 p.

21. I. Koryttsev, S. Sheiko, V. Kartashov, O. Zubkov, V. Oleynikov, I. Selieznov, M. Anohin. Practical Aspects of Range Determination and Tracking of Small Drones by Their Video Observation // 2020 International Scientific-Practical Conference. Problems of Infocommunications. Science and Technology. Kharkiv, Ukraine. October 6-9, 2020. – 5 p.