

Міністерство освіти і науки України  
Державний ВНЗ «Національний гірничий університет»

Факультет інформаційних технологій  
(факультет)

Кафедра програмного забезпечення комп'ютерних систем  
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА  
дипломної роботи

*магістра*  
(назва освітньо-кваліфікаційного рівня)

галузь знань *12 «Інформаційні технології»*  
(шифр і назва галузі знань)

спеціальність *122 «Комп'ютерні науки»*  
(код і назва)

спеціалізація *Інформаційні управляючі системи та технології*  
(код і назва спеціалізації)

освітній рівень *магістр*  
(назва освітнього рівня)

кваліфікація *інженер з комп'ютерних систем*  
(назва кваліфікації)

на тему: *Підвищення ефективності програмного забезпечення дронів з використанням програми Mission Planner при моніторингу в сільському господарстві.*

Виконавець:

студент 2 курсу, групи 122м-16-1

(підпис)

*Безродний П.П.*

(прізвище та ініціали)

Керівники	Посада, прізвище, ініціали	Оцінка	Підпис
проекту	<i>проф. Алексєєв М.О.</i>		
розділів:			
Спеціальний	<i>проф. Алексєєв М.О.</i>		
Економічний			

Рецензент			
-----------	--	--	--

Нормоконтроль	<i>доц. Коротенко Л.М.</i>		
---------------	----------------------------	--	--

Дніпро  
2018



- програмного забезпечення, що реалізує оригінальний алгоритм обходу заборонених польотних зон.

#### 4 ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Результати магістерської роботи повинні відповідати вимогам паспорту наукової спеціальності 122 – «Інформаційні технології».

Результати досліджень мають бути подані у вигляді, що дозволяє побачити та оцінити безпосереднє використання оригінального алгоритму створення та перевірки геоданих БПЛА. Згідно виробничих функцій та професійних задач магістра, які виносяться на кваліфікаційну роботу, повинні бути розроблені програмна документація та програмне забезпечення, що реалізує оригінальний алгоритм створення та перевірки геоданих БПЛА.

#### 5 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	Строки виконання робіт (початок – кінець)
Формування та формалізація завдання	21.04.2017-01.05.2017
Аналіз сфери застосування алгоритмів заборони зон БПЛА	18.08.2017-21.09.2017
Проектування та розробка програмного продукту, що вирішує поставлені в дипломній роботі задачі	24.09.2017-04.10.2017
Тестування розробленої автоматизованої системи на і аналіз отриманих результатів	06.10.2017-20.11.2017

#### 6 РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

**Економічний ефект** від реалізації результатів роботи очікується позитивним завдяки зменшенню витрат на створення та перевірку геоданих БПЛА.

**Соціальний ефект** від реалізації результатів роботи очікується позитивним завдяки підвищенню захисту цілісності створення та перевірки геоданих БПЛА, створених за допомогою програмного забезпечення, що реалізує оригінальний алгоритм створення та перевірки геоданих БПЛА.

#### 7 ДОДАТКОВІ ВИМОГИ

Відповідність оформлення «ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення».

Завдання видав

\_\_\_\_\_ (підпис)

*Алексєєв М.О.*  
\_\_\_\_\_ (прізвище, ініціали)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

*Безродний П.П.*  
\_\_\_\_\_ (прізвище, ініціали)

Дата видачі завдання: 1.03.2017 р.

Термін подання дипломної роботи до ДЕК  . .2018 р.

## Реферат

Пояснювальна записка: \_\_ стор., \_\_ Рис., \_\_ таблиця., \_\_ додатка, \_\_ джерел.

Об'єкт досліджень – геоінформаційні потоки даних безпілотних літальних апаратів.

Предмет досліджень – захист геоінформаційних даних в безпольотних зонах.

Мета НДР – створити вдосконалений алгоритм обходу заборонених польотних зон.

Вихідні дані для проведення роботи: теоретичні й експериментальні дослідження та знання, а також технологічні аспекти існуючих алгоритмів керування безпілотними літальними апаратами.

У введенні приведений опис зв'язку завдань з об'єктом діяльності, особливостей проблеми процесу збору, зберігання і аналізу геоінформаційних, а також проаналізована можливість використання інформаційних технологій для вдосконалення його етапів.

У першому розділі йдеться про особливості розвитку безпілотних літальних апаратів та методів автопілотування.

У другому розділі викладений опис алгоритмів пошуку найкоротшого шляху, та проведені теоретичні дослідження.

В третьому розділі приведено опис розробленого програмного продукту.

У розділі "Економіка" розраховані трудомісткість розробки програмного забезпечення, витрати на створення ПО і тривалість його розробки.

Наукова новизна результатів, що очікуються, полягає у:

- проведені аналізу та виявленні недоліків традиційних алгоритмів створення та перевірки геоданих БПЛА;
- розробці оригінального алгоритму обходу та захисту геоданих заборонених для польотів зон.

Практична цінність результатів полягає у розробленні:

- програмного забезпечення, що реалізує оригінальний алгоритм створення та перевірки геоданих БПЛА.

Список ключових слів : ГЕОІНФОРМАЦІЙНІ СИСТЕМИ, АИС, АНАЛІЗ І ВІЗУАЛІЗАЦІЯ ДАНИХ.

## ABSTRACT

Master diploma: \_\_pages., \_\_ fig., \_\_ table., \_\_ addition, \_\_ sources.

An object of researches is геоінформаційні streams of these pilotless aircrafts.

The article of researches is a protection of геоінформаційних data in безпольотних areas.

Purpose of science report - to create the improved algorithm of round of the restricted flight areas.

Initial works given for realization : theoretical and experimental researches and knowledge, and also technological aspects of existent algorithms of management pilotless aircrafts.

In introduction the brought description over of connection of tasks with the object of activity, features of problem of process of collection, storage and analysis of геоінформаційних, and also the analysed possibility of the use of information technologies for perfection of his stages.

Speech goes in the first division about the features of development of pilotless aircrafts and methods of autoaviating.

In the second division the expounded description of algorithms of search of short cut, and theoretical researches are conducted.

Description over of the worked out software product is brought in the third division.

In a division "Economy" expected labour intensiveness of software, expense development on creation software and duration of his development.

The scientific novelty of results which are expected consists in:

- conducted to the analysis and exposure of lacks of traditional algorithms of creation and verification of геоданих UAV;

- to development of original algorithm of round and defence of the геоданих areas restricted for flights.

The practical value of results consists in development:

- software which will realize the original algorithm of creation and verification of геоданих UAV.

List of keywords : GEOGRAPHIC INFORMAION SYSTEMS, AIS, ANALYSIS And VISUALIZATION of DATA.

## РЕФЕРАТ

Объяснительная записка: \_\_ стр., \_\_ рис., \_\_ таблиц., \_\_ дополнений, \_\_ источников.

Объект исследований - геоинформационные потоки данных беспилотных летательных аппаратов.

Предмет исследований - защита геоинформационных данных в беспольотных зонах.

Цель НИР - создать усовершенствованный алгоритм обхода запрещенных полетных зон.

Исходные данные для проведения работы: теоретические и экспериментальные исследования и знания, а также технологические аспекты существующих алгоритмов управления беспилотными летательными аппаратами.

В введении приведено описание связи заданий с объектом деятельности, особенностей проблемы процесса сбора, хранения и анализа геоинформационных, а также проанализированная возможность использования информационных технологий для совершенствования его этапов.

В первом разделе идет речь об особенностях развития беспилотных летательных аппаратов и методов автопилотирования.

Во втором разделе изложено описание алгоритмов поиска кратчайшего пути, и проведены теоретические исследования.

В третьем разделе приведено описание разработанного программного продукта.

В разделе "Экономика" рассчитаны трудоемкость разработки программного обеспечения, затраты на создание ПО и длительность его разработки.

Научная новизна результатов, которые ожидаются, заключается в:

- проведенные анализу и выявлении недостатков традиционных алгоритмов создания и проверки геоданных БПЛА;
- разработке оригинального алгоритма обхода и защиты геоданных запрещенных для полетов зон.

Практическая ценность результатов заключается в разрабатывании:

- программного обеспечения, которое реализует оригинальный алгоритм создания и проверки геоданных БПЛА.

Список ключевых слов : ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ, АИС, АНАЛИЗ И ВИЗУАЛИЗАЦИЯ ДАННЫХ.

## ЗМІСТ

Вступ.....	8
РОЗДІЛ 1 .....	12
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	12
1.1. Особливості експлуатації безпілотних літальних апаратів .....	12
1.2. Склад бортового обладнання сучасних БПЛА .....	14
1.3. Бортовий комплекс навігації та управління БПЛА .....	17
1.4. Система управління просторовим становищем гостронаправлених антенних систем в комплексах БПЛА .....	19
1.5. Інерціальна система БПЛА .....	22
1.6. Бортовий комплекс управління БПЛА .....	23
Розділ 2 .....	27
Алгоритми пошуку найкоротшого шляху стосовно БПЛА.....	27
2.1. Проблема пошуку найкоротшого шляху .....	27
2.2. Застосування для пошуку шляху, стосовно БПЛА .....	41
2.3. Експериментальна перевірка .....	42
Розділ 3 .....	45
Практична частина .....	45
3.1. Побудова безпілотного літального апарату на базі польотного контролера АРМ 2.6.....	45
3.2. Модифікація модуля маршрутизації на основі модуля пошуку найкоротшого шляху .....	55
Розділ 4 .....	65
ЕКОНОМІЧНА ЧАСТИНА .....	65
4.1. Визначення трудомісткості розробки програмного забезпечення ..	<b>Error! Bookmark not defined.</b>
4.2. Витрати на створення програмного забезпечення ..	<b>Error! Bookmark not defined.</b>
Висновки .....	<b>Error! Bookmark not defined.</b>
Додаток А.....	79
Додаток Б .....	86
Додаток В .....	87
Додаток Г .....	88

## Вступ

В даний час безпілотні літальні апарати (БПЛА) почали твердо займати свою нішу в різних сферах діяльності людини. Успішне використання БПЛА в військових областях сприяло їх широкому застосуванню в цивільних цілях. Вони знаходять своє застосування в містобудуванні, екологічному моніторингу, геологорозвідці, при дистанційному контролі нафтогазопроводів і ін.

Багаторічний досвід використання даних дистанційного зондування Землі дозволив БПЛА швидко знайти застосування в географічних дослідженнях, в біогеографії. Особливо важливі такі дослідження при вивченні особливо охоронюваних природних територій, основним завданням яких, є збереження біологічного та ландшафтного різноманіття унікальних природних екосистем. Виконання цього завдання пов'язане з деякими проблемами: - необхідність постійних наукових досліджень; - малий штат співробітників; - "фактор неспокою"; - важкодоступність територій. У зв'язку з чим, застосування безпілотних літальних апаратів є найбільш ефективним засобом для спостереження і дистанційного збору даних про їх станах.

Не дивлячись на велику кількість завдань, що вирішуються за допомогою БПЛА, їх наукове застосування в Україні істотно відстає від такого за кордоном, тому видається цікавим і актуальним пошук можливостей їх застосування в нашій країні. Метою даної роботи є вивчення можливостей застосування безпілотних літальних апаратів для контролю сільгоспугідь. Для досягнення мети поставлені такі завдання:

- вивчити БПЛА і їх технічні можливості;
- проаналізувати світовий досвід використання БПЛА;
- провести польові дослідження;
- обробити отримані дані;
- показати можливості використання БПЛА в біогеографічних дослідженнях.



Згідно знаходяться у відкритому доступі документів організацій Європейського Союзу, розподіл споживчого попиту на цивільні БПЛА в період з 2015 по 2020 рр. виглядає наступним чином: 45% - урядові структури, 25% - пожежні, 13% - сільське господарство і лісництво, 10% - енергетика, 6% - огляд земної поверхні, 1% - зв'язок і мовлення.

Використання безпілотних літальних апаратів (БПЛА) для проведення аерофотозйомки є вельми актуальним напрямком розвитку методів збору геопросторових даних. Для деяких територій космічні знімки доводиться чекати місяцями. Аерофотозйомка і повітряне лазерне сканування володіють більш високою актуальністю і точністю, але забезпечують покриття менших за площею територій, в порівнянні з космічною зйомкою. Також обидва вище зазначених способу отримання знімків є дорогими. А використання БПЛА виправдано в тих випадках, коли необхідно швидко отримати точну інформацію про місцевість на невеликі території. До того ж, з урахуванням собівартості кожного з рішень, БПЛА займають дуже вигідні позиції, а в деяких випадках є оптимальними з точки зору фінансових витрат. Таким чином, плюсами застосування БПЛА є:

1. Рентабельність.
2. Можливість зйомки з невеликих висот і поблизу об'єктів, відповідно - отримання знімків високого дозволу.
3. Оперативність отримання знімків.
4. Можливість застосування в зонах надзвичайних ситуацій без ризику для життя і здоров'я пілотів

**Актуальність роботи:** Актуальність роботи: Нові інформаційні технології в сільському господарстві, хвиля яких дійшла до України в останні десять років, стали називати «точним землеробством». Точне землеробство ще називають топоорієнтованим землеробством, землеробством за приписом, точним сільським господарством, акуратним сільським господарством. Точне землеробство найкраще розглядати як комплект технологій, а не як одну технологію. Стрижнем всієї технології є

геоінформаційні системи, що дозволяють знімати, накопичувати і обробляти інформацію, що характеризує посів або рілля [1, 2]. Основна різниця між традиційним і точним землеробством знаходиться в застосуванні сучасних інформаційних технологій для збору, обробки та аналізу різних даних з високим просторовим і тимчасовим дозволом для прийняття рішень і виконання сільськогосподарських операцій. На сьогоднішній день для збору необхідної інформації застосовують знімки із супутників і аерофотозйомку. Світовий і російський досвід підтверджує, що в сільськогосподарському виробництві зйомки з космосу дають багато можливостей, Однак зйомки з космічних апаратів ведуться для отримання характеристик стану земель на федеральному (глобальному) і регіональному рівнях. Купівля космічних знімків в необхідному, наприклад, для прогнозу врожайності кількості і якості, часто недоступна не тільки звичайним сільгоспвиробникам, а й великим агрохолдингам. Крім цього, космічні зйомки в більшій мірі пов'язані з часом доби і погодними умовами. Альтернативним методом отримання інформації, що цікавить, особливо на невеликих площах (від декількох квадратних метрів до всього господарства в цілому), є використання безпілотних літальних апаратів (БПЛА) з різною знімальною апаратурою.

Цілі й завдання дослідження. Метою даної магістерської роботи є створення й впровадження алгоритмів обходу перешкоди в сферу роботи програми для вибору польотного завдання, так само опис роботи методик відбору.

Об'єкт дослідження - геоінформаційні потоки даних безпілотних літальних апаратів.

Предмет дослідження - захист геоінформаційних даних до безпілотних зонах.

Ідея роботи створення алгоритмів захисту геоданих від БПЛА.

Методи дослідження. При вирішенні поставленого завдання використовувалися наукові досягнення в областях алгоритмів пошуку Шляхів при цьому до алгоритму ставлять такі вимоги: По-перше, він не

повинен допускати помилок на прецедентах, й по-друге - задовольняти Деяке дотадове обмеження, що відображає властивості й особливості предметної області. В рамках алгебраїчного підходу була доведена можливість побудови коректних алгоритмів для широкого класу регулярних завдань

Наукова новизна. Отриманий результат полягає у створеній методиці пошуку шляху для БПЛА.

Практичне значення отриманих результатів полягає в розробці програмних засобів, які дозволяють здійснювати пошук шляху для найбільш коректного захисту геоданих.

Результати дипломної роботи можуть бути використані підприємствами, фірмами, розробниками для проектування інформаційних систем, створення програмного продукту

Особливий внесок магістра складається в:

- відборі методів ДОСЛІДЖЕНЬ й технологій реалізації;
- створення інформаційної системи, що реалізує механізми пошуку маршруту;
- розробки теоретичної часті роботи, в якій досліджені й систематизовані знання про існуючі підходи розробки інформаційних систем;
- оцінці отриманих результатів.

Структура й обсяги роботи. Робота складається з вступу, чотирьох розділів й висновка. Містить \_\_ сторінки друкованого тексту, з \_\_ малюнками, \_\_ використаних джерел.

## РОЗДІЛ 1

### АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Необхідною умовою успішного протікання сучасних виробничих процесів є своєчасне і достовірне інформаційне забезпечення. Це особливо актуально для проектів і робіт, що розгортаються на великих територіях. Аерозйомка вже протягом десятиліть є ефективним інструментом для виконання пошукових робіт в геодезії, геофізичних дослідженнях, для проведення різного виду моніторингів.

#### **1.1. Особливості експлуатації безпілотних літальних апаратів**

З появою комерційної аерокосмічної зйомки сформувався ринок даних геоінформатики. Сучасна геоінформатика надає користувачам потужний інструмент візуалізації, аналізу, систематизації, зберігання геопросторових даних. ГІС використовуються вже не тільки державними органами (наприклад, для ведення кадастрового обліку), а й існують численні корпоративні ГІС додатки, що забезпечують обґрунтоване прийняття рішень в складних проектах по використанню природних ресурсів, будівельних, сільськогосподарських та інших. Питання про інформаційне наповнення ГІС, їх актуалізації вирішується, виходячи з доступних засобів. В основному зараз - це дані космічної аерозйомки. Однак, незважаючи на безперервне вдосконалення інструментарію аерокосмічного ДЗЗ, така зйомка має відомі методичні обмеження, які визначаються, перш за все, неможливістю проводити зйомку в будь-який час в будь-якому місці через погодні умови і з огляду на геометрії орбіт супутників. БПЛА значно перевершують космічні засоби з оперативності зйомки. Для моніторингу протяжних об'єктів, якими є магістральні трубопроводи, лінії електропередачі, морські та сухопутні кордони, лінії залізних і автомобільних доріг.

Перелічимо основні переваги БПЛА: по-перше, вони в середньому на порядок дешевше пілотованих літаків, які потрібно оснащувати системами

життєзабезпечення, захисту, мають меншу ефективну поверхню розсіювання і меншу вразливість від вогню ППО противника. Для них потрібно, нарешті, готувати пілотів (а це коштує великих грошей). У системах з БПЛА якщо апаратом керують з землі, то один пілот може керувати кількома машинами. У підсумку виходить, що відсутність екіпажу на борті істотно знижує витрати на виконання того чи іншого завдання.

По-друге, легкі (в порівнянні з пілотованими літаками) безпілотні літальні апарати споживають менше палива.

По-третє, на відміну від пілотованих літаків машинам без пілота не потрібні аеродроми з бетонним покриттям. Досить побудувати ґрунтову злітно-посадкову смугу довжиною всього близько 600 м. («Безпілотники» злітають за допомогою катапульти, а приземляються «по-літаковому», як винищувачі на авіаносцях або на парашуті). Це дуже серйозний аргумент, оскільки 70% аеродромів в Україні потребують реконструкції, а темпи ремонту сьогодні - один аеродром на рік [5].

По-четверте, сучасна пілотована повітряна розвідка при всіх своїх виявних та аналізуючих можливостях обмежена радіусом дії та кутом піднесення, на якому здійснюється огляд земної поверхні. Зі зменшенням кута піднесення (до  $1^\circ$  і менше) і в міру збільшення дальності різко зростає ефект затінення цілей, обумовлений впливом рельєфу місцевості. Як показав досвід регіональних збройних конфліктів останніх років, вплив ефекту затінення цілей на різних театрах військових дій різний. У Близькосхідному регіоні, що характеризується переважанням пустельної місцевості, він близький до мінімального, а в гористому Балканському серйозно знижує ефективність застосування літаків-розвідників, для яких тут не менше 50% розвідувати території потрапляє в зони невидимості. Розвідувальні БПЛА, крім контролю таких зон, а також пошуку за зоною видимості, що переглядається бортовими засобами пілотованих літаків-розвідників, яка становить для видової розвідки 150 км, радіолокаційної - 250 км, радіо- і радіотехнічної - до 300 км (ці значення повинні бути зменшені на 50 ... 100 км через необхідність

безпечного видалення маршрутів патрулювання від лінії фронту), забезпечують багато ракурсів різномасштабних спостережень за прихованими об'єктами противника, перебуваючи далеко за лінією фронту [6].

Переваги та специфічні якості комплексів з БПЛА зумовлюють їх роль як високомобільного і боєготового компоненту ВПС, здатного в короткі терміни не тільки збирати достовірну розвідувальну інформацію про противника на великій території і в великій глибині, а й вести цілевказівки для високоточної зброї ВПС, ВМФ і сухопутних військ, наносити удари по найбільш важливим об'єктам супротивника, діяти гнучко в умовах інтенсивного протидії ППО відповідно за мінливою обстановкою. Ці якості обумовлюють зростання ролі комплексів з БПЛА, які можуть застосовуватися в ході ведення бойових дій в тактичній і оперативній ланках (в перспективі - і в стратегічній ланці), а також для вирішення народно-господарських завдань.

## **1.2 Склад бортового обладнання сучасних БПЛА**

Для забезпечення завдань спостереження підстильної поверхні в реальному масштабі часу в процесі польоту і цифрового фотографування обраних ділянок місцевості, включаючи важкодоступні ділянки, а також визначення координат досліджуваних ділянок місцевості корисне навантаження БПЛА [3,4,5] повинна містити в своєму складі:

- Пристрої отримання видової інформації;
- Спутникову навігаційну систему (ГЛОНАСС / GPS);
- Пристрої радіолінії видової і телеметричної інформації;
- Пристрої командно-навігаційної радіолінії з антенно-фідерним пристроєм;
- Пристрій обміну командної інформації;
- Пристрій інформаційного обміну;
- Бортова цифрова обчислювальна машина (БЦВМ);
- Пристрій зберігання видової інформації.

Сучасні телевізійні (ТВ) камери забезпечують уявлення оператору в реальному часі картини спостереження місцевості в форматі найбільш близькому до характеристик зорового апарату людини, що дозволяє йому вільно орієнтуватися на місцевості і при необхідності виконувати пілотування БПЛА. Можливості по виявленню, і розпізнаванню об'єктів визначаються характеристиками фотоприймача і оптичної системи телевізійної камери. Основним недоліком сучасних телевізійних камер є їх обмежена чутливість, що не забезпечує цілодобового застосування. Застосування тепловізійних (ТПВ) камер дозволяє забезпечити цілодобовість застосування БПЛА. Найбільш перспективним є застосування комбінованих теле-тепловізійних систем. При цьому оператору видається синтезоване зображення, що містить найбільш інформативні частини, властиві мабуть і інфрачервоному діапазонах довжин хвиль, що дозволяє істотно підвищити тактико-технічні характеристики системи спостереження. Однак подібні системи складні технічно і досить дорогі. Застосування РЛС дозволяє отримувати інформацію цілодобово і при несприятливих метеорологічних умовах, коли ТВ і ТПВ канали не забезпечують отримання інформації. Застосування змінних модулів, дозволяє знизити вартість і реконфігурування складу бортового обладнання для вирішення поставленого завдання в конкретних умовах застосування. Розглянемо склад бортового обладнання міні-БПЛА.

- Оглядний курсовий пристрій закріплюється нерухомо під деяким кутом до стрийової осі літального апарату, що забезпечує необхідну зону захоплення на місцевості. До складу оглядового курсового пристрою може входити телевізійна камера (ТК) з широкопольним об'єктивом (ШПЗ). Залежно від розв'язуваних завдань може бути оперативно замінена або доповнена тепловізійною камерою (ТПВ), цифровим фотоапаратом (ЦФА) або РЛС.

- Пристрій детального огляду з поворотним пристроєм складається з ТК детального огляду з узькопольним об'єктивом (УПЗ) і

трьохкоординатним поворотним пристроєм, що забезпечує розворот камери по курсу, крену і тангажу по командам оператора для детального аналізу конкретної ділянки місцевості. Для забезпечення роботи в умовах низької освітленості ТК може бути доповнена тепловізійної камерою (ТПВ) на мікроболометричній матриці з узькопольним об'єктивом. Можлива також заміна ТК на ЦФА. Подібне рішення дозволить використовувати БПЛА для проведення аерофотозйомки при розвороті оптичної осі ЦФА в надир.

- Пристрої радіолінії видової і телеметричної інформації (передавач і антенно-фідерний пристрій) повинні забезпечувати передачу видової і телеметричної інформації в реальному або близькому до реального масштабі часу на ПУ в межах радіовидимості.

- Пристрої командно-навігаційної радіолінії (приймач і антенно-фідерний пристрій) повинні забезпечувати прийом в межах радіовидимості команд пілотування БПЛА і управління його обладнанням.

- Пристрій обміну командної інформацією забезпечує розподіл командно і навігаційної інформації по споживачам на борту БПЛА.

- Пристрій інформаційного обміну забезпечує розподіл видової інформації між бортовими джерелами видової інформації, передавачем радіолінії видової інформації і бортовим пристроєм зберігання видової інформації. Цей пристрій також забезпечує інформаційний обмін між усіма функціональними пристроями, що входять до складу цільового навантаження БПЛА за обраним інтерфейсом (наприклад, RS-232). Через зовнішній порт цього пристрою перед зльотом БПЛА проводиться введення польотного завдання і здійснюється передстартовий автоматизований вбудований контроль на функціонування основних вузлів і систем БПЛА.

- Супутникова навігаційна система забезпечує прив'язку координат (прив'язка) БПЛА і спостережуваних об'єктів за сигналами глобальної супутникової навігаційної системи ГЛОНАСС (GPS). Супутникова навігаційна система складається з одного або двох приймачів (ГЛОНАСС / GPS) з антенними системами. Застосування двох приймачів, антени яких



рознесені по будівельній осі БПЛА, дозволяє визначати крім координат БПЛА значення його курсового кута.

- Бортова цифрова обчислювальна машина (БЦВМ) забезпечує управління бортовим комплексом БПЛА.

- Пристрій зберігання видової інформації забезпечує накопичення обраної оператором (або відповідно до польотного завдання) видової інформації до моменту посадки БПЛА. Цей пристрій може бути зйомним або стаціонарним. В останньому випадку повинен бути передбачений канал знімання накопиченої інформації в зовнішній пристрій після посадки БПЛА. Інформація, зчитана з пристрою зберігання видової інформації, дозволяє проводити більш детальний аналіз при дешифруванні отриманої в польоті БПЛА видової інформації.

- Вбудований блок живлення забезпечує узгодження по напрузі і струмів споживання бортового джерела живлення і пристроїв, що входять до складу корисного навантаження, а також оперативний захист від коротких замикань і перевантажень в електромережі. Залежно від класу БПЛА корисне навантаження може доповнюватися різними видами РЛС, датчиками екологічного, радіаційного та хімічного моніторингу. Комплекс управління БПЛА є складною, багаторівневою структурою, основне завдання якої - забезпечити виведення БПЛА в заданий район і виконання операцій відповідно до польотного завдання, а також забезпечити доставку інформації, отриманої бортовими засобами БПЛА, на пункт управління.

### **1.3 Бортовий комплекс навігації та управління БПЛА**

Бортовий комплекс "Лелека" є повнофункціональним засобом навігації і управління безпілотного літального апарату (БПЛА) літакової схеми. Комплекс забезпечує: визначення навігаційних параметрів, кутів орієнтації і параметрів руху БПЛА (кутових швидкостей і прискорень); навігацію і управління БПЛА при польоті по заданій траєкторії; стабілізацію кутів орієнтації БПЛА в польоті; видачу в канал передачі телеметричної

інформації про навігаційні параметри, кути орієнтації БПЛА. Центральним елементом БК "Лелека" є малогабаритна інерціальна навігаційна система (ІНС), інтегрована з приймачем супутникової системи навігації. Побудована на базі мікроелектромеханічних датчиків (MEMS гіроскопів і акселерометрів) за принципом платформеної ІНС, система є унікальним високотехнологічним виробом, гарантує високу точність навігації, стабілізації і управління ЛА будь-якого класу. Вбудований датчик статичного тиску забезпечує динамічне визначення висоти і вертикальної швидкості. Склад бортового комплексу: блок інерціальної навігаційної системи; приймач СНС; блок автопілота; накопичувач Льотних Даних; датчик повітряної швидкості У базовій конфігурації управління здійснюється по каналах: елерони; кермо висоти; кермо напрямку; контролер двигуна. Комплекс сумісний з радіоканалом РСМ (імпульсно-кодова модуляція) і дозволяє управляти БПЛА як в ручному режимі зі стандартного пульта дистанційного керування, так і в автоматичному, по командам автопілота. Керуючі команди автопілота генеруються в формі стандартних широтно-імпульсно-модульованих (ШІМ) сигналів, що підходять до більшості типів виконавчих механізмів.

Фізичні характеристики: розміри, мм: блок автопілота - 80 x 47 x 10; ІНС - 98 x 70 x 21; приймач СНС - 30 x 30 x 10; вага, кг: блок автопілота - 0,120; ІНС - 0,160; приймач СНС - 0,03. Електричні характеристики: напруга живлення, В - 10 ... 27; споживана потужність (макс.), Вт - 5. Навколишнє середовище: температура, °С - від -40 до +70; вібрація / удар, g - 20.

Управління: порти RS-232 (2) - прийом / передача даних; порти RS-422 (5) - зв'язок із зовнішніми пристроями; канали ШІМ (12) - керуючі пристрої; програмовані ПІМ (255) - поворотні пункти маршруту. Робочі діапазони: крен -  $\pm 180^\circ$ ; тангаж -  $\pm 90^\circ$ ; курс (шляховий кут) - 0 ... 360; прискорення -  $\pm 10 g$ ; кутова швидкість -  $\pm 150^\circ / \text{сек}$ .

#### **1.4 Система управління просторовим становищем гостронаправлених антенних систем в комплексах БПЛА**

Сам по собі безпілотний літальний апарат (БПЛА) - лише частина складного комплексу, одна з основних завдань якого - оперативне доведення отриманих відомостей до оперативного персоналу пункту управління (ПУ). Можливість забезпечення стійкого зв'язку є однією з найважливіших характеристик, що визначають експлуатаційні можливості комплексу управління БПЛА і забезпечує доведення відомостей, отриманих БПЛА, в режимі «реального часу» до оперативного персоналу ПУ. Для забезпечення зв'язку на значні відстані і підвищення перешкодозахищеності за рахунок просторової селекції в комплексах управління БПЛА широко використовуються гостронаправлені антенні системи (АС) як на ПУ, так і на БПЛА. Функціональна схема системи управління просторовим становищем остронаправленої АС, що забезпечує оптимізацію процесу входження в зв'язок в комплексах управління БПЛА, наведена на Рис. 1.1.

Система управління гостронаправленої АС (див. Рис. 1) включає в себе:

- Власне гостронаправлену АС, радіотехнічні параметри якої вибираються, виходячи з вимог забезпечення необхідної дальності зв'язку по радіолінії.
- Сервопривід АС, що забезпечує просторову орієнтацію ДН АС в напрямку очікуваного появи випромінювання об'єкта зв'язку.
- Систему автоматичного супроводу по напрямку (АСН), що забезпечує стійкий автосупровід об'єкта зв'язку в зоні впевненого захоплення пеленгаційної характеристики системи АСН.
- пристрій, який одночасно забезпечує формування сигналу «Зв'язок», що свідчить про прийом інформації з заданою якістю.
- Процесор управління антенною системою, що забезпечує аналіз поточного стану системи управління АС, формування сигналів управління сервоприводом для забезпечення просторової орієнтації АС відповідно до польотного завдання і алгоритмом просторового сканування, аналіз наявності

зв'язку, аналіз можливості переключення сервоприводу АС з режиму «Зовнішнє управління» в режим «Автосупроводження», формування сигналу переключення сервоприводу АС в режим «Зовнішнє управління».

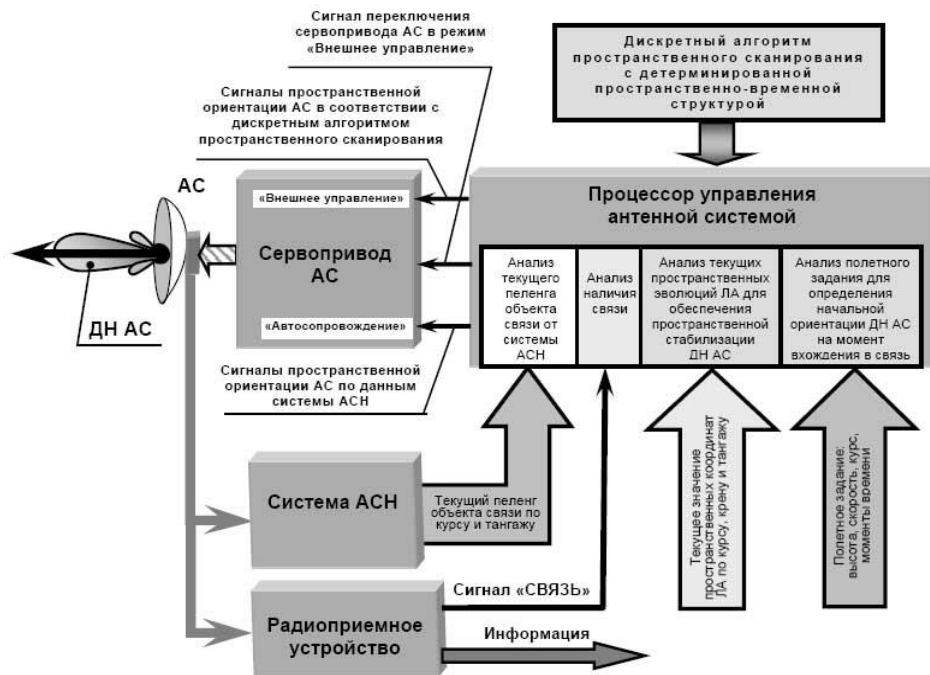


Рис. 1.1. - Функціональна схема системи управління просторовим становищем гостронаправленої АС в комплексах управління БПЛА

Основне завдання, що виконується системою управління просторовим становищем гостронаправленої АС, - забезпечити стійке входження в зв'язок з об'єктом, заданим польотним завданням.

Це завдання розпадається на ряд підзадач:

- Забезпечення просторової орієнтації ДН АС в напрямку очікуваної появи випромінювання об'єкта зв'язку і її просторової стабілізації для випадку розташування АС на борту літального апарату.
- Розширення зони стійкого захоплення випромінювання об'єкта зв'язку за рахунок застосування дискретного алгоритму просторового сканування з детермінованою просторово-часовою структурою.
- Перехід в режим стійкого автосупроводження об'єкта зв'язку системою АСН при виявленні об'єкта зв'язку.

- Забезпечення можливості повторного входження в зв'язок в разі її зриву. Для дискретного алгоритму просторового сканування з детермінованою просторово-часовою структурою можна виділити наступні особливості:

- Сканування ДН АС здійснюється дискретно в часі і в просторі. • Просторові переміщення ДН АС при скануванні здійснюються таким чином, щоб не залишалося просторових зон, які не перекриваються зоною впевненого захоплення системою АСН за весь цикл сканування (див. Рис. 1.2).

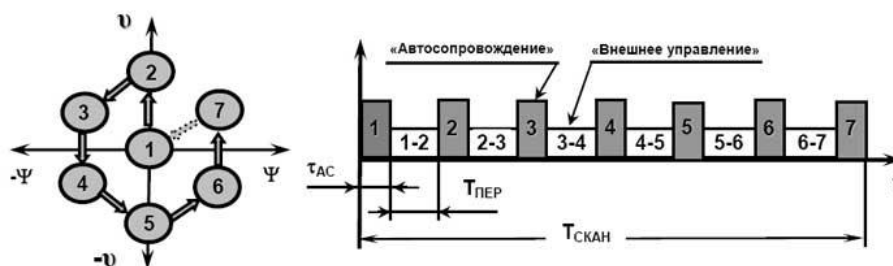


Рис. 1.2 - Приклад організації дискретного просторового сканування в азимутальній і кутомісцевої площинах

Для кожного конкретного просторового положення, що визначається алгоритмом сканування, можна виділити дві фази: «Автосупроводження» і «Зовнішнє управління».

- У фазі «Автосупроводження» система АСН здійснює оцінку можливості прийому випромінювання об'єкта зв'язку для обраного просторового положення РСН.

У разі позитивного результату оцінки: Просторове сканування припиняється. Система АСН продовжує здійснювати автосупровід випромінювання об'єкта зв'язку за своїм внутрішнім алгоритмом. На вхід сервоприводу АС надходять сигнали просторової орієнтації АС за даними поточного пеленга об'єкта зв'язку від системи АСН ХАСН (t). У разі негативного результату оцінки: Здійснюється просторове переміщення РСН АС в таке просторове положення, яке визначається алгоритмом сканування.

- У фазі «Зовнішнє управління» на виході процесора управління антенною системою формуються сигнали управління сервоприводом АС. Компоненти сигналу управління сервоприводом забезпечують:

$X_0$  - первісну просторову орієнтацію ДН АС в напрямку на об'єкт зв'язку;  $\Delta X_{ЛА}(t)$  - паріровання просторових еволюцій літального апарату;  $X_{АЛГ}(t)$  - розширення зони стійкого захоплення випромінювання об'єкта зв'язку системи АСН відповідно до дискретних алгоритмах просторового сканування з детермінованою просторово-часовою структурою.

У разі зриву зв'язку, починаючи з моменту часу  $T_{СВ} = 0$  (зникнення сигналу «ЗВ'ЯЗОК»), сигнал  $X_{АСН}(T_{СВ} = 0)$  запам'ятовується в пристрої «Обчислення і зберігання», і використовується в подальшому процесором управління АС як значення очікуваного пеленгу об'єкта зв'язку. Процес входу в зв'язок повторюється як описано вище.

Конкретний вид сигналів управління визначається конструктивними особливостями сервоприводу антенної системи.

### **1.5 Інерціальна система БПЛА**

Ключовим моментом в згаданому ланцюжку є «вимір стану системи». Тобто координат місця розташування, швидкості, висоти, вертикальної швидкості, кутів орієнтації, а також кутових швидкостей і прискорень. У бортовому комплексі навігації і управління, розробленому і виробленим ТОВ «ТеКнол», функцію вимірювання стану системи виконує малогабаритна інерціальна інтегрована система (Мінс). Маючи в своєму складі тріади інерційних датчиків мікромеханічних гіроскопів і акселерометров), а також барометричний висотомір і тривісний магнітометр, і комплексірую дані цих датчиків з даними приймача GPS, система виробляє повне навігаційне рішення по координатам і кутах орієнтації. Мінс розробки ТеКнола - це повна Інерціальна система, в якій реалізований алгоритм бесплатформенной ІНС, інтегрованої з приймачем системи супутникової навігації. Саме в цій системі міститься «таємниця» роботи всього комплексу управління БПЛА. По суті, одночасно працюють три навігаційних системи в одному

обчислювачі по одним і тим самим даним. Ми їх називаємо «платформами». Кожна з платформ реалізує свої принципи управління, маючи свої «правильні» частоти (низькі або високі). Майстер-фільтр вибирає оптимальне рішення з будь-якої з трьох платформ в залежності від характеру руху. Цим забезпечується стійкість системи не тільки в прямолінійному русі, але і при виражах, некоординованих розворотах, бічному поривчастому вітрі. Система ніколи не втрачає горизонту, і забезпечує правильні реакції автопілота на зовнішні перешкоди і адекватний розподіл впливів між органами управління БПЛА.

## **1.6 Бортовий комплекс управління БПЛА**

До складу Бортового Комплексу Навігації та Управління БПЛА входять три складові елементи (Малюнок 1.3).

1. Інтегрована Навігаційна Система;
2. Приймач супутникової Навігаційної системи
3. Модуль автопілота.

Модуль автопілота здійснює вироблення керуючих команд у вигляді ШІМ (широтно-імпульсно-модульованих) сигналів, згідно до законів управління, закладених в його обчислювач. Крім управління БПЛА, автопілот програмується на управління бортовою апаратурою:

- стабілізація відеокамери,
- синхронізоване за часом і координатами спрацьовування затвора фотоапарата,
- випуск парашута,
- скидання вантажу або відбір проб в заданій точці і інші функції. На згадку автопілота може бути занесено до 255 поворотних пунктів маршруту. Кожна точка характеризується координатами, висотою проходження і швидкістю польоту.

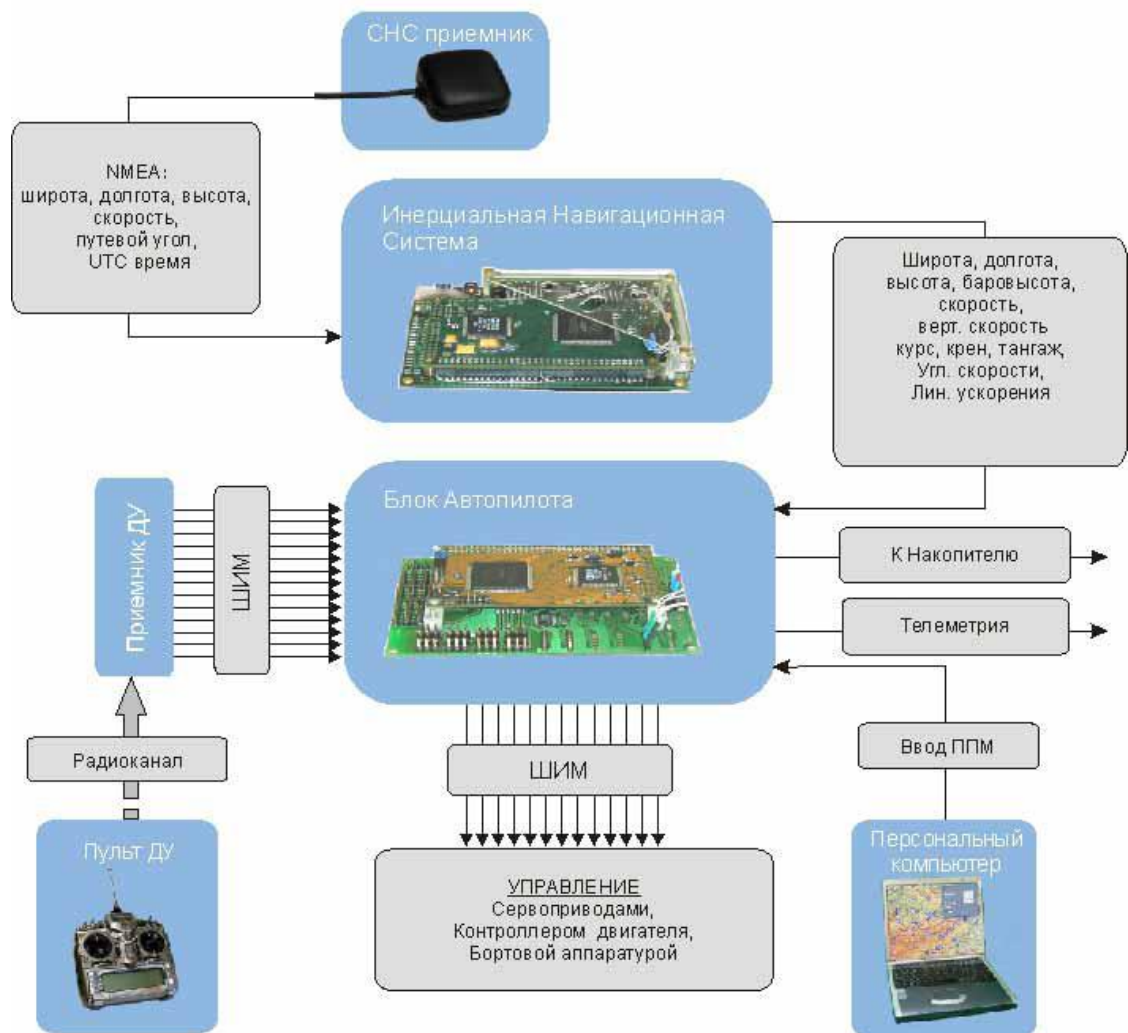


Рис. 1.3. – Бортовой комплекс навигации и управления БПЛА

У польоті автопілот також забезпечує видачу в канал передачі телеметричної інформації для стеження за польотом БПЛА (Малюнок 1.4). Багато фірм зараз декларують, що забезпечують своїм системам автоматичний політ за допомогою «найменшого в світі автопілота».

Найбільш показовий приклад такого рішення - продукція канадської фірми "Micropilot". Для формування сигналів управління тут використовують «сирі» дані - сигнали від гіроскопів і акселерометрів. Таке рішення за визначенням не є надійним (стійким до зовнішніх впливів і чутливим до умов польоту) і в тій чи іншій мірі працездатний тільки при польоті в стабільній атмосфері.



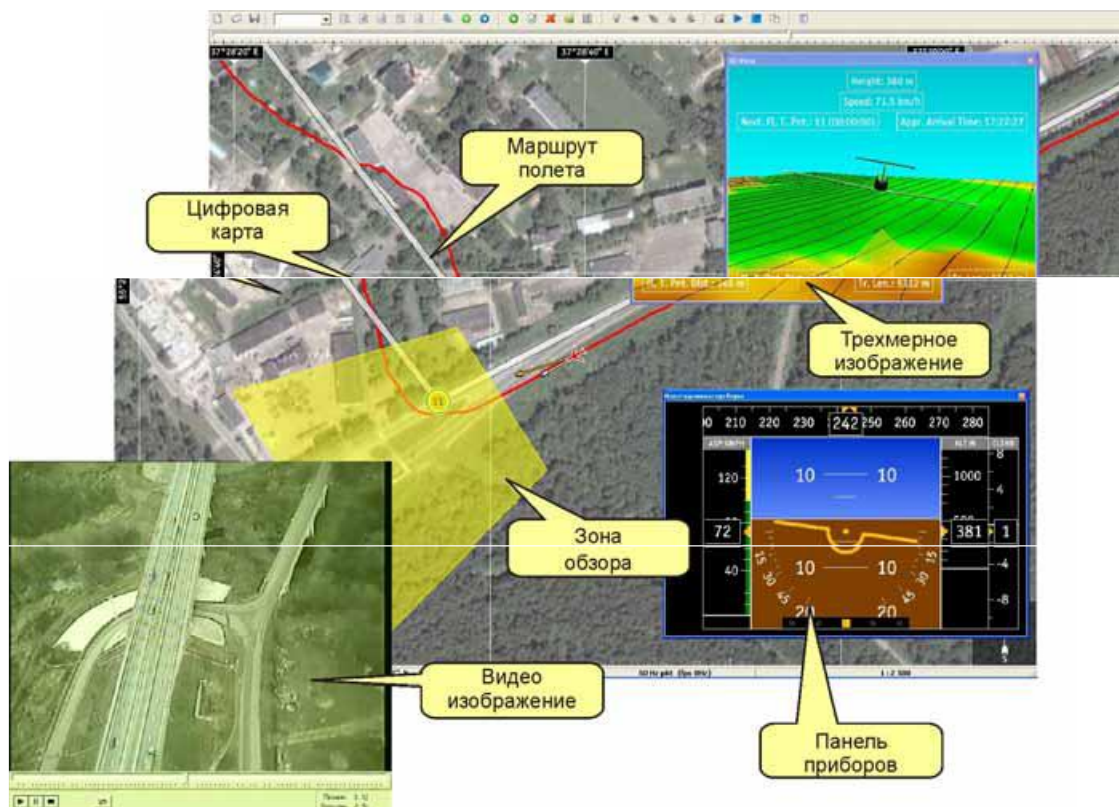


Рис. 1.4. – Телеметричні дані БПЛА

Будь-яке істотна зовнішня перешкода (порив вітру, висхідний потік або повітряна яма) може призвести до втрати орієнтації літального апарату і аварією. Тому всі, хто коли-небудь стикався з подібною продукцією, рано чи пізно розуміли обмеженість таких автопілотів, які ніяк не можуть бути використані в комерційних серійних системах БПЛА.

Розглянутий склад бортового обладнання БПЛА дозволяє забезпечити вирішення широкого кола завдань з моніторингу місцевості і важкодоступних для людини районів в інтересах народного господарства. Застосування до складу бортового обладнання телевізійних камер дозволяє в умовах гарної метеовидимості і освітленості забезпечити високий дозвіл і детальний моніторинг підстильної поверхні в режимі реального часу. Застосування ЦФА дозволяє використовувати БПЛА для проведення аерофотозйомки в заданому районі з подальшою детальною дешифруванням. Використання ТПВ апаратури дозволяє забезпечити цілодобовість застосування БПЛА, хоча і з меншим дозволом, ніж при використанні

телевізійних камер. Найбільш доцільне застосування комплексних систем, наприклад ТВ-ТПВ, з формуванням синтезованого зображення. Однак такі системи поки ще досить дорогі. Наявність на борту РЛС дозволяє отримувати інформацію з меншим дозволом, ніж ТБ і ТПВ, але цілодобово і при несприятливих метеорологічних умовах. Застосування змінних модулів пристроїв отримання видової інформації, дозволяє знизити вартість і реконфігурацію складу бортового обладнання для вирішення поставленого завдання в конкретних умовах застосування. Можливість забезпечення стійкого зв'язку є однією з найважливіших характеристик, що визначають експлуатаційні можливості комплексу управління БПЛА.

## Розділ 2

### Алгоритми пошуку найкоротшого шляху стосовно БПЛА

#### 2.1. Проблема пошуку найкоротшого шляху

##### Задача:

Дана мережа автомобільних доріг:

- орієнтований граф  $G = (V, E)$ , де
  - $V$  — безліч перехресть,
  - $E$  — безліч доріг;
- $l(u, v) \geq 0$  — середній час, який займає проїзд по дорозі,
- $|V| = n, |E| = m$ ,
  - відправна точка - вершина,  $s$ , пункт призначення - вершина  $t$

Мета: знайти найкоротший шлях  $s \rightsquigarrow t$

Кількість перехресть і доріг може бути дуже великим, тоді звичайні алгоритми пошуку шляху будуть працювати дуже довго, тому спробуємо оптимізувати їх для більш швидкої роботи.

##### Алгоритм Дейкстри

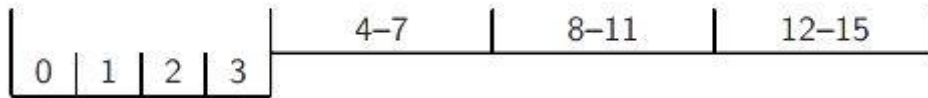
- На кожному кроці вибирає з безлічі непереглянутих вершин вершину з найменшою відстанню до старту і релаксує ребра, що виходять з неї,
- Завершує свою роботу, коли мета досягнута (або переглянуті всі вершини). Швидкість роботи алгоритму Дейкстри сильно залежить від швидкості операцій з пріоритетною чергою.

Оскільки ми розглядаємо мережу автомобільних доріг, то  $m = O(n)$  (граф планарії майже всюди). Для фібоначчійовий куп час роботи алгоритму становить  $O(n \log n + n)$ , для довічних куп:  $O(n \log n)$

Але на практиці частіше використовуються 2-, 4- і 8-ічні купи: вони простіші, оцінка часу роботи містить меншу кількість прихованих констант.

## Покращення алгоритму Дейкстри

### Багаторівневі кошика (англ. Multi-level buckets, MLB)



Підходить тільки графів з цілочисельними ребрами.

- Необхідно складати вершини в "кошика"  $B[i] = \{u \in B[i] : d(u) = i\}$ ,
- Структура даних буде підтримувати індекс  $L = \{\forall i : i < L \Rightarrow B[i] = \emptyset\}$
- на кожному кроці алгоритму, якщо  $B[L]$  порожньо, то збільшимо  $L$ , а інакше дістанемо одну вершину із  $B[L]$ ,
- при релаксації будемо прибирати вершину з вихідної кошика і класти в кошик, відповідну новому значенням  $d(u)$ .

### Порівняння різних структур даних для пошуку найкоротшого шляху на карті Європи (CPU 2,4GHz, 16MB RAM)

Структура даних	Час роботи (сек)
Двійкова купа	12,38
4- купа	11,53
8- купа	11,52
MLB	9,36
MLB + калібрування	8,04

Можна помітити, що при такій реалізації, всі операції з пріоритетною чергою будуть виконуватися за  $O(1)$ . Тоді, для однорівневої реалізації час роботи алгоритму Дейкстри можна оцінити як  $O(m + nC)$ , де  $C$  — максимальна довжина ребра в графі.

При дворівневої реалізації підтримуватимемо два рівня кошиків: перший рівень буде відповідати однорівневої реалізації, а кошики другого рівня будуть містити діапазон значень кошиків першого рівня, які в них входять. В

цьому випадку, нам потрібно підтримувати два індекси  $L_{top}$  та  $L_{bottom}$  для кожного з рівнів відповідно.

При такій реалізації, час роботи алгоритму Дейкстри можна оцінити як  $O(m + n(1 + \sqrt{C}))$

### Калібрування (англ. Caliber)

Введемо величину калібр вершини  $c(v)$  — вага мінімального ребра, що входить в  $v$ , або  $\infty$ , якщо в вершину не входить жодне ребро. Будемо говорити, що поточне значення  $d(v)$  точно, якщо воно дорівнює довжині шляху  $s \rightsquigarrow v$ .

#### Лема (1):

Припустимо, що довжина ребер невід'ємна. Нехай  $\mu$  - мінімальне з потокової значень. Тоді, якщо існує така вершина  $u$ , що  $\mu + c(u) \geq d(u)$ , то поточне значення  $d(u)$  точно.

Ця лема дозволяє нам пом'якшити правило вибору поточної вершини в алгоритмі Дейкстри, при цьому зберігаючи інваріант (майже всі вершини обробляються раз). Калібрування використовує Лему 1 щоб знаходити і обробляти вершини з точними поточними значеннями відстані до них.

Модифікуємо нашу MLB - структуру: будемо зберігати помічені вершини в двох структурах: дерево пошуку і пріоритетна черга, реалізована на MLB. Алгоритм, наведений нижче, називається алгоритмом розумної черзі (англ. Smart queue). Вершини в  $F$  матимуть точні мітки  $d(u)$ . Якщо  $F$  непусто, ми видалимо зідти вершину і прорелаксіруємо всіх її сусідів. Якщо ж  $F$  порожньо, ми дістанемо з  $B$  вершину з мінімальною міткою і прорелаксіруємо всіх її сусідів.

Розглянемо механізм релаксації: нехай ми зменшуємо  $d(u)$ . Зауважимо, що в цьому випадку  $u$  не могло лежати в  $F$  (інакше  $d(u)$  було неточно).

Якщо  $u \in B$  — застосуємо `decreasekey` до  $u$ . Ця операція або

перемістила  $u$  всередині  $B$ , або визначила, що мітка  $d(u)$  точна і перемістила  $u$  у  $F$ . Якщо  $u \notin F$  та  $u \notin B$ , ми застосуємо операцію `insert`, й  $u$  запишеться в  $F$  або  $B$ , в залежності від того, чи виконується умова леми.

### Двонаправлений пошук

Ми можемо зменшити кількість відвіданих вершин в алгоритмі Дейкстри, просто запустивши його із початкової та кінцевої вершини. Така евристика не зіпсує швидкість роботи в гіршому випадку.

Створимо дві пріоритетних черги й запустимо на одній з них алгоритм Дейкстри, який шукає  $d_{forward}(v)$  з  $s$ , а на іншій - шукає  $d_{reverse}(v)$  з  $t$ .

Алгоритм завершить свою роботу, коли яка-небудь вершина буде видалена з обох черг.

Тонкість цього алгоритму полягає в тому, що найкоротший шлях  $s \rightsquigarrow t$  не обов'язково пройде через вершину  $z$ . Тому після зупинки двонаправленого пошуку, нам необхідно перебрати всі ребра з вершин, що мають  $d_{forward}(u)$  в вершини з  $d_{reverse}(v)$  і знайти ребро  $uv$  з мінімальним  $d_{forward}(u) + \ell(uv) + d_{reverse}(v)$ . Якщо ця величина менше, ніж довжина спочатку знайденого шляху - то це і є результат роботи алгоритму.

На практиці, такий двонаправлений пошук швидше звичайного алгоритму Дейкстри приблизно в два рази.

Алгоритм  $A^*$

Наведемо трохи змінену версію цього алгоритму.

візьмемо функцію  $h(v) : V \rightarrow \mathbb{R}$  — *потенціал (англ. potential) вершини*. Тоді, з її допомогою можна визначити *потенційну вартість (англ. Reduced cost) кожного ребра* як  $\ell_h(v, w) = \ell(v, w) - h(v) + h(w)$

Зауважимо, що заміна на не змінить найкоротших шляхів: візьмемо будь-який шлях  $P = (s = v_0, v_1, \dots, v_k, v_{k+1} = t)$ . Тоді  $\ell_h(P) = \ell_h(s, v_1) + \ell_h(v_1, v_2) + \dots + \ell_h(v_k, t) =$

$$\ell(s, v_1) - h(s) + h(v_1) + \ell(v_1, v_2) - h(v_1) + h(v_2) + \dots + \ell(v_k, t) - h(v_k) + h(t) = \ell(P) - h(s) + h(t).$$

Таким чином довжини все шляхів  $s \rightsquigarrow t$  зміняться на одну і ту ж величину  $h(t) - h(s)$

У нашому випадку, алгоритм  $A^*$  буде еквівалентний алгоритму Дейкстри, на графі  $G_h$ , у якого вартості ребер замінили на їх потенційні вартості. На кожному кроці необхідно буде вибирати з черги вершину  $v$  з мінімальним значенням  $\ell(P_{s \rightsquigarrow v}) - h(s) + h(v)$ . очевидно,  $h(s)$  буде однаковим для будь-якої вершини  $v$ .

Назвемо функцію  $h$  правдоподібною (**англ. feasible**), якщо  $\forall(u, v) : \ell_h(u, v) \geq 0$ . Відомо, що, якщо  $h(t) \leq 0$  та  $h$  правдоподібна, то для будь-якого  $v$ ,  $h(v)$  — нижня границя  $d(v, t)$

Головна відмінність від алгоритму Дейкстри в тому, що  $A^*$  є поінформованим алгоритмом - він обробляє в першу чергу ті вершини, які знаходяться ближче до результату. Скорость работы алгоритма  $A^*$ :

- в гіршому випадку —  $h(v) = 0$  — вироджується в алгоритм Дейкстри
- в кращому випадку —  $\forall v : h(v) = d(v, t)$ 
  - $\ell_h(v, w) = 0$ , якщо ребро  $(v, w)$  лежить на найкоротшому шляху, інакше потенційна вартість позитивна
  - всі відвідані вершини будуть лежати на найкоротшому шляху

### Двонаправлений $A^*$

Для двобічної версії алгоритму нам потрібні дві потенційні функції:

- $p_f(v)$ , оцінює  $d(v, t)$
- $p_r(v)$ , оцінює  $d(s, v)$

У цьому випадку з'являється додаткова проблема: різні потенційні вартості у ребер для різних обходів:

- $\ell_{p_f}(v, w) = \ell(v, w) - p_f(v) + p_f(w)$  — якщо ребро обробляється в обході, розпочатому у  $s$
- $\ell_{p_r}(v, w) = \ell(v, w) - p_r(w) + p_r(v)$  — якщо ребро обробляється в обході, розпочатому у  $t$

Щоб уникнути цієї проблеми, необхідно, щоб

$\ell_{p_f}(v, w) = \ell_{p_r}(v, w) \Leftrightarrow p_f(v) + p_r(v) = p_f(w) + p_r(w) = \text{const}$ . Крім того, функції повинні бути однотонними.

Рішення - використовувати усереднені потенційні функції:

$$h_f(v) = \frac{p_f(v) - p_r(v)}{2}$$

$$h_r(v) = \frac{p_r(v) - p_f(v)}{2} = -h_f(v)$$

При такому виборі потенційних функцій, виконується  $\forall u : h_f(u) + h_r(u) = 0$  і

тоді двонаправлений A\* стає аналогічним двонаправленим алгоритму

Дейкстрифо

### Випадковий вибір (random)

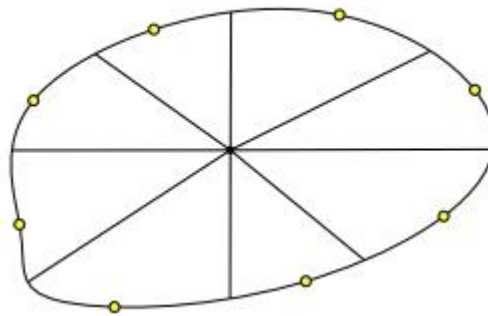


Рис. 2.2 – Орієнтири випадкового вибору

Як випливає з назви, орієнтири вибираються випадковим чином.

### Площинний (planar)

- Розділимо карту на секторів однакової площі,
- візьмемо орієнтиром найбільш віддалену точку від центру в кожному сектор



Такий спосіб підходить, тільки якщо граф має відносно правильну форму. На практиці зазвичай використовується оптимізована версія цього алгоритму.

### **Виборчий (avoid)**

Цей алгоритм додає орієнтири по одному, дивлячись на вершини, які погано покриті поточними набором орієнтирів  $S$ .

Побудуємо з випадково вибраної вершини  $r$  дерево кратчайших путей  $T_r$ .

**Вагою кожної вершини  $v$**  в цьому дереві назвемо різницю між істинною довжиною шляху  $d(v, r)$  і нижньої оцінкою цієї довжини  $\underline{d}(v, r)$ , отриманої за поточним набором орієнтирів. розміром вершини  $v$  назвемо суму її ваги і розмірів всіх її нащадків в  $T_r$ . якщо подерево  $T_r$  з коренем в  $v$  содержит орієнтир, розмір  $v$  дорівнює 0.

Починаючи з максимальною за розміром вершини, підемо вниз по дереву  $T_r$  і знайдемо лист з максимальним розміром. Прийmemo його за новий орієнтир.

### **Пошук максимального покриття (maxCover)**

Основним мінусом виборчого методу є те, що перший орієнтир вибирається випадковим чином і вибір подальших орієнтирів буде сильно залежати від початкового.

Ми можемо поліпшити знайдені орієнтири, якщо спочатку, використовуючи виборчий метод, знайдемо набір орієнтирів у кілька (зазвичай, 4) рази більше, ніж необхідно, а потім відсіємо зайві мінімізуючи час запиту.

Оцінимо якість набору орієнтирів ґрунтуючись на покритті дуг. Будемо говорити, що орієнтир  $L$  покриває дугу  $(v, w)$ , якщо вершина  $v$  знаходиться на найкоротшому шляху  $L \rightsquigarrow w$ . Тобто,  $\ell(v, w) = \text{dist}(L, w) - \text{dist}(L, v)$ , тоді такий вибір орієнтира дасть нам нижню межу для всіх шляхів, що містять  $(v, w)$ .

Якщо хоча б один орієнтир з покриває дугу  $(v, w)$ , то і весь  $S$  покриває цю дугу.

Цей метод є найкращим, але є найбільш повільним. Завдання вибору орієнтирів в цьому випадку стає NP-повною.

## Reach

Ця евристика ґрунтується на інтуїтивному спостереженні: не варто відвідувати "локальні" дороги, коли ми знаходимося досить далеко і від  $s$ , і від  $t$

Нехай вершина лежить на найкоротшому шляху  $P : s \rightsquigarrow t$ . Тоді, назвемо охопленням (англ. Reach) вершини

$v$  відносно  $P$  величину  $r(v, P) = \min\{dist(s, v), dist(v, t)\}$ . Охопленням вершини щодо всього графа назвемо величину  $r(v) = \max_P r(v, P)$  — максимум за всіма найкоротшим шляхам, що містить  $v$ .

Крім цього, буде корисним ввести поняття охоплення ребра. Назвемо охопленням ребра  $(v, w) \in P$  щодо величину  $\min\{dist(s, v), dist(w, t)\}$ .

Аналогічно, охопленням ребра щодо всього графа назвемо величину  $r(v, w) = \max_P r((v, w), P)$  — максимум за всіма найкоротшим шляхам, що містить  $(v, w)$ .

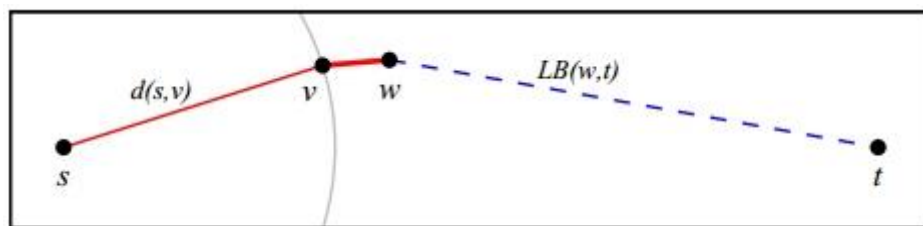


Рис. 2.3 – вибір вершини

Зауважимо, що вершини з більшим охопленням знаходяться поблизу середини деякого довгого найкоротшого шляху, тобто:

- на великих автомагістралях вершини мають велике охоплення,
- на локальних перехрестях (всередині населених пунктів) вершини мають маленький охоплення.

Під час обробки ребра  $(v, w)$ :

- видалимо вершину  $w$ , якщо  $r(w) < \min\{d(s, v) + \ell(v, w), LB(w, t)\}$ ,
- оцінка  $LB(w, t)$  повинна бути підібрана таким чином, щоб, якби  $P = (s, \dots, v, w, \dots, t)$  було найкоротшим шляхом,  $r(w)$  було б більше.

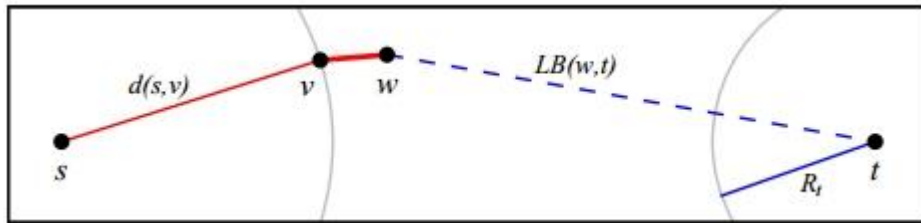


Рис. 2.4 – Вибір наступної вершини

Як шукати нижню оцінку довжини шляху  $LB(w, t)$ ?

- Явно: евклідова відстань, за допомогою орієнтирів.
- Неявно: зробити пошук двонаправленим.

Наприклад, радіус  $R_t$  пошуку в зворотну сторону може бути нижній оцінкою, тому що якщо вершина  $w$  була відвідана пошуком в зворотну сторону, то  $d(w, t) \geq R_t$

Таким чином, будемо видаляти  $w$ , якщо  $r(w) < \min\{d(s, v) + \ell(v, w), R_t\}$

Для поліпшення результату, нам необхідно збалансувати прямий і зворотний пошук.

Розглянемо препроцесінг:

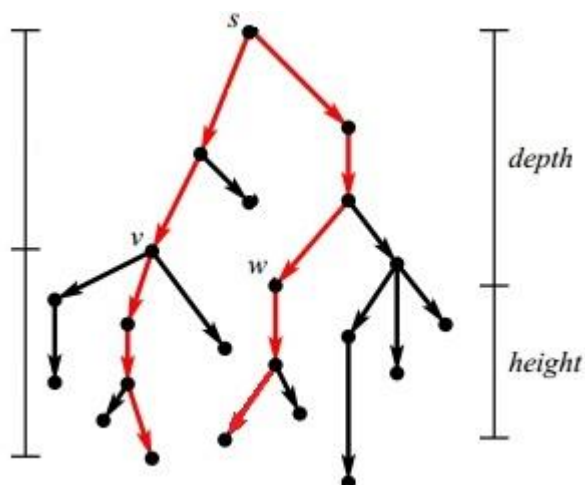


Рис. 2.5 - Препроцесінг

- на початковому етапі  $\forall v : r(v) \leftarrow 0$ ,
- для кожної вершини  $s \in V$  розглянемо дерево найкоротших шляхів до всіх інших вершин  $T_s$ ,
  - знайдемо найбільш довгий шлях  $P_{s \rightsquigarrow t}$ , що містить  $v$ ,
  - Назвемо глибиною вершини  $d_s(v)$  відстань від  $s$ , висотою вершини  $h_s(v)$  — відстань до найбільш далекого нащадка вершини,
  - Тоді, охопленням вершини в цьому дереві буде величина  $r_s(v) = \min\{d_s(v), h_s(v)\}$ ,
  - Тоді відновимо значення охоплення  $r(v) \leftarrow \max\{r(v), r_s(v)\}$ .

Складність алгоритму:  $O(nm)$ , тому він занадто повільний для великих графів і його потрібно поліпшити.

Фінальна версія препроцесінга матиме дві фази:

- основна фаза (будуються частково оброблені дерева і додаються скорочують шлях ребра)
- фаза налагодження (вершини з більшим охопленням обробляються зазначеним вище алгоритмом - їх набагато менше, тому обробка буде швидкою)

### Скорочення області пошуку

Зауважимо, що нам потрібні тільки вершини з маленьким охопленням  $r(v) < \varepsilon$ ,  $\varepsilon = \text{const}$ . Зауважимо також, що якщо  $r(v) \geq \varepsilon$ , то існує такий шлях  $P$ , що на ньому лежить вершина  $v \in P$ , для якої виконується умова  $r(v, P) \geq \varepsilon$

Назвемо найкоротший шлях  $P_{s \rightsquigarrow t} = (s, s', \dots, v, \dots, t', t)$   $\varepsilon$ - мінімальним, якщо виконуються умови:  $\text{dist}(s, v) \geq \varepsilon$ ,

- $\text{dist}(s', v) < \varepsilon$ ,
- $\text{dist}(v, t) \geq \varepsilon$ ,
- $\text{dist}(v, t') < \varepsilon$ .

Таким чином, алгоритм буде виглядати так:

- знайдемо оцінку охоплення  $r'(v)$ , використовуючи тільки  $\varepsilon$  — мінімальні шляху,
- якщо  $r'(v) < \varepsilon$ , то оцінка коректна:  $r(v) = r'(v)$ ,
- якщо  $r'(v) \geq \varepsilon$ , то вона нас не цікавить:  $r(v) \geq r'(v)$ .

Корисно буде розглядати частково оброблені дерева (англ. Partial trees) - дерева найкоротших шляхів, що зберігають шляху довжиною, меншою певного порогу. Тоді дерево найкоротших шляхів буде глибиною близько  $2\varepsilon$ :

- встановимо  $G' \leftarrow G$  и  $\varepsilon \leftarrow \varepsilon_0$  (маленьке число),
- поки  $G'$  не пусто:
  - знайдемо частково оброблене дерево найкоротших шляхів з  $v$ , щоб знайти вершини з охопленням  $r(v) \geq \varepsilon$ ,
  - видалимо з  $G'$  решта вершини (з охопленням  $r(v) < \varepsilon$ , вже оброблені),
  - встановимо  $\varepsilon \leftarrow 3\varepsilon$ .

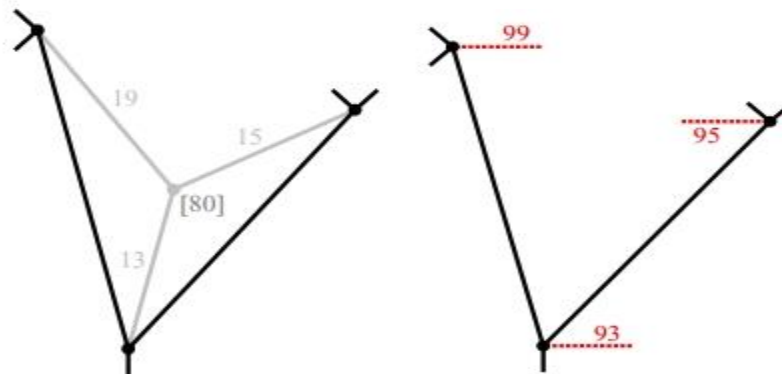


Рис. 2.6 - Частково оброблені дерева

## Пенальті

Попереднє поліпшення створює проблему: ми повинні припускати, що найкоротший шлях може починатися в вершині з маленьким охопленням, які ми відкидаємо на кожній ітерації. Для того, щоб прийняти їх до уваги, ми

введемо нову величину - пенальті (англ. Penalty) - верхню оцінку на довжину найкоротшого шляху в "відкинутої" зоні.

Нехай  $G_i = (V_i, A_i)$  — підграф вихідного графа, отриманий на  $i$ -й ітерації алгоритму, описаного вище. Назвемо входять пенальті (in-penalty) вершини  $v \in V_i$  величину  $\max\{\bar{r}(u, v) : u, v \in A \setminus A_i\}$ , якщо у  $v$  було як мінімум одне викинуте в процесі алгоритму входить ребро, або нуль в іншому випадку. Аналогічно визначимо виходить пенальті для вихідних з вершини ребер.

Змінимо алгоритм, щоб замінити вершини на їх пенальті. Для цього перевизначити глибину і висоту вершини.

- Глибиною вершини  $v \in T_x$ , де  $T_x$  — частково оброблене дерево з коренем у  $x$  величину  $depth(v) = d(v) + in-penalty(x)$ , де  $d(v)$  — довжина шляху від кореня до вершини в дереві.
- Щоб визначити висоту вершини, нам потрібно поняття "помилкових листя". Для кожної вершини в дереві найкоротших шляхів додамо нащадка  $v'$  — помилковий лист, і ребро  $(v, v') = out-penalty(v)$ . У певному сенсі, буде виступати в ролі всіх ребер, спочатку інцидентних вершині, які ми відсікли. Тоді висотою вершини  $v$  буде відстань від неї, до найбільш далекого помилкового листа. Підкреслимо, що в даний момент помилкові листя додаються неявно, і тільки після того, як дерево буде частково оброблено.

### Скорочення шляхів

Уявимо послідовність вершин ступеня 2 з великим охопленням, що йдуть підряд. Додамо, що скорочує ребро (англ. Shortcut) - ребро, що проходить з початку шляху в його кінець з довжиною, що дорівнює довжині цього шляху. Таким чином ми можемо зменшити охоплення вершин на цьому шляху і прискорити препроцесінг (зменшивши кількість прохідних ребер), але збільшимо пам'ять, необхідну для зберігання нашого графа.

На цьому кроці ми будемо шукати пропускалися (англ. Bypassable) вершини. Назвемо вершину пропускається, якщо виконується одна з двох умов:  $v$  має тільки одне вхідне ребро  $(u, v)$  і одне виходить ребро  $(v, w)$ ,

- $v$  має два вхідних ребра  $(u, v), (w, v)$  і два вихідних ребра  $(v, w), (v, u)$ .

В обох випадках мається на увазі, що  $u \neq w$ , тобто у  $v$  обов'язково є тільки два сусіда. В першому випадку,  $v$  — кандидат на односторонній пропуск (англ. one-way bypass), у другому - на двосторонній (англ. two-way bypass). Ми будемо використовувати скорочують ребра, щоб пропускати такі вершини.

Лінія (line) - шлях в графі, що містить як мінімум три вершини, так, що всі вершини, крім початкової і кінцевої, що пропускалися. Кожна пропускається вершина може належати тільки одній лінії. Лінії також можуть бути односторонне- і двухсторонне- пропускаючи, в залежності від типу вершин, які вони містять. Легко помітити, що на лінії можуть лежати вершини тільки одного типу.

Як тільки ми знайшли лінію, найпростішим рішенням буде додати одне скорочує ребро з початку в кінець. Але практика показує, що краще скоротити ще й входять до її складу лінії меншого розміру, це ще більш зменшить охоплення пропускаються вершин. Для цього перед додаванням скорочує ребра по лінії, рекурсивно знайдемо вершину, яка знаходиться приблизно на її середині і опрацюємо лівий і правий відрізки як повноцінні лінії.

### **Видалення пропущених вершин**

Уявімо односторонню лінію, що складається з трьох вершин  $u, v, w$ . Коли ми додали скорочує ребро  $(u, w)$ , ми знали, що ребро  $(u, v)$  більше ніколи не буде використовуватися на найкоротшому шляху містить подпут  $u \rightsquigarrow w$ . Крім того, будь-який найкоротший шлях, що містить  $(u, v)$ , буде закінчуватися або у  $v$ , або в прилеглої вершині з низьким охопленням. Тоді, коректної верхньої

оцінкою охоплення  $(u, v) \in \bar{r}(u, v) = \ell(u, v) + out\text{-penalty}(v)$ .

Аналогічно,  $\bar{r}(v, w) = \ell(v, w) + in\text{-penalty}(v)$ . Знаючи це, ми можемо видалити вершину і суміжні їй ребра з графа і оновити відповідні значення пенальті для її сусідів.

Таку ж процедуру можна виконати з двосторонньої лінією, тому що крім оцінок, зазначених вище, можна додати:  $\bar{r}(w, v) = \ell(w, v) + out\text{-penalty}(v)$ ,

▪  $\bar{r}(v, u) = \ell(v, u) + in\text{-penalty}(v)$ .

### Фаза налагодження

Той факт, що ми використовуємо пенальті, для прискорення обчислення коректних верхніх оцінок охоплення, призводить до того, що оцінки стають менш точними в процесі роботи алгоритму і з ростом пенальті. Отже, для вершин, які довше знаходяться в графі, помилки накопичуються. Це погано, потім що такі вершини і є найважливішими в графі - у них високий охоплення, їх відвідує велика кількість найкоротших шляхів. Якби ми могли зробити їх більш точними, під час запиту можна було б пропустити більшу кількість вершин.

В цьому і полягає мета фази налагодження. Після того, як ми знайдемо вірні значення верхніх оцінок, використовуючи частково оброблені дерева, під час фази налагодження ми перерахуємо охоплення  $\delta = const$  вершин з найбільшими значеннями охоплення (в них найбільше помилок).

Нехай  $V_\delta \subset G$  — безліч вершин з високим охопленням потужністю. Щоб перерахувати їх охоплення, спочатку необхідно знайти підграф  $G_\delta = (V_\delta, A_\delta)$ . Цей граф містить не тільки початкові ребра, а й додані під час основної фази скорочують ребра. Потім ми запусимо пошук точних значень охоплення для кожної вершини цього подграфа. Так як дерева найкоротших шляхів будуть містити тільки вершини з  $G_\delta$ , то нам все ще потрібно використовувати вхідні та вихідні пенальті для інших вершин. Але, тим не менш, вони будуть менше, тому що для найбільших ми підраховали точне значення охоплення.



Під час фази налагодження ми піклуємося про точність, а не про швидкість, тому не додаємо нових скорочують ребер і використовуємо точний алгоритм. Тому, час роботи алгоритму як мінімум  $\Omega(\delta^2)$

## 2.2. Застосування для пошуку шляху, стосовно БПЛА

Завдання визначення найкоротшого шляху для безпілотних літальних апаратів зазвичай формується гранично просто: найкоротший шлях для літального апарату - пряма. Однак у ряді випадків доводиться змінювати цьому правилу, наприклад у разі забороненої для польотів зони (наприклад, військового об'єкта, або ж аеропорту). У таких випадках пропонується використовувати один із варіантів пошуку найкоротшого шляху, обходять заборонений об'єкт по найкоротшій траєкторії. Таким чином, завдання дослідження полягає в пошуку найкоротшого шляху обходу забороненого об'єкта, заданого вершинами багатокутника на координатній сітці (рис. 2.7).

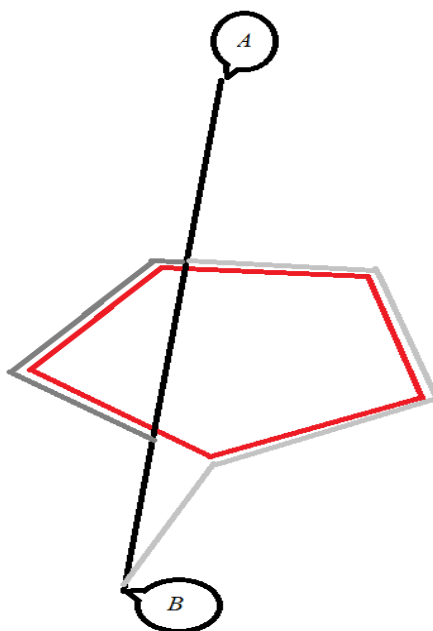


Рис. 2.7 – Обхід забороненої зони

### 2.3. Експериментальна перевірка

Для порівняння ефективності існуючих алгоритмів планування траєкторії на площині і запропонованого був проведений ряд експериментів. На мові програмування C ++ були реалізовані алгоритми HGA \*, A \* і WA \*. За допомогою розроблених програмних засобів були згенеровані МТ-графи розміром  $50 \times 50$ ,  $100 \times 100$ ,  $250 \times 250$ ,  $500 \times 500$ ,  $1000 \times 1000$  з різним ступенем заповнення перешкодами (СЗП). Перешкоди представляли собою прямокутники шириною в одну клітку і довжиною, розподіленої рівномірно із середнім  $L = 10$  клітин. Ступінь заповнення МТ-графа перешкодами розраховувалася за формулою:  $\lambda = ((L * 2) + 4) * N$  (5) Використовуючи вираз (5) для кожного МТ графа визначалося необхідну кількість перешкод N для досягнення параметра  $\lambda$  дорівнює 0,3, 0,5 і 0,8. Приклади згенерованих МТ-графів наведені на Рис. 7. Всього було створено 150 МТ-графів (табл.1). Побудовані граfi є модельні приклади важливою практичного завдання - побудови траєкторії переміщення малого безпілотного літального апарату (БЛА) типу «вертоліт» в міських умовах, коли для забезпечення максимальної скритності траєкторія вертольота повинна бути еквідистантним земної поверхні і проходити нижче рівня висотних міських будівель (т. е. БЛА повинен використовувати лише маневри в горизонтальній площині для досягнення мети). В ході експериментів на кожному МТ-графі випадковим чином вибиралися з безлічі крайніх клітин початкова і цільова (так щоб глибина рішення - число клітин в найкоротшому можливому шляху -  $d$  дорівнювала 50, 100, 250, 500 і 1000 відповідно). Для відшукування шляху між обраними клітинами застосовувалися такі алгоритми: - A \* (в таблицях позначений як «A Search 1»); - WA \* з вагою евристики рівним 3 (в таблицях позначений як «A Search 3»); - WA \* з вагою евристики рівним 5 (в таблицях позначений як «A Search 5»); - HGA \* (в таблицях позначений як «HGA»). Як було сказано вище, основним недоліком існуючих підходів до побудови траєкторії на площині є надмірне розгляд клітин МТ-графа, які необхідно зберігати в оперативній пам'яті. Саме тому за основний показник

ефективності порівнюваних алгоритмів був узятий параметр  $Q$  - кількість розглянутих (і збережених в пам'яті ЕОМ) клітин МТ-графа. Також спостерігалось параметр  $W$  - вага побудованого шляху.

Відомо, що алгоритм  $A^*$  буде найкоротший шлях до мети, тому крім абсолютних показників  $Q$  і  $W$  розглядався також інтегральний показник ефективності алгоритму  $E = (Q / W) / (Q_1 / W_1)$ , де  $Q_1$  - кількість розглянутих клітин алгоритмом  $A^*$  для побудови шляху з вагою  $W_1$ , що є найкоротшим. Даний коефіцієнт дозволяє судити про те, у скільки разів даний алгоритм ефективніше використовує обчислювальні ресурси при збереженні «відносного якості» рішення (в таблицях з результатами параметр  $E$  відображений в процентному відношенні).

Аналіз отриманих даних дозволяє зробити два основні висновки.

Побудова найкоротших шляхів досить ресурсоемна через надмірне витрачання обчислювальних ресурсів.

Відзначимо особливо той факт, що результати, показані алгоритмами  $WA^* -3$  і  $WA^* -5$ , практично ідентичні, з чого можна зробити висновок, що подальше підвищення ваги евристики не призведе до скорочення кількості розглянутих клітин. Отже, можна припустити, що алгоритм  $HGA^*$  в середньому в 1,5-4 рази ефективніше будь-яких наявних аналогів, в основу яких закладено принцип ітерационного  $A^*$ -Пошук.

Завдання планування траєкторії на площині є однією з ключових при розробці систем управління безпілотними транспортними засобами нового покоління. Для вирішення цього завдання традиційно використовуються методи штучного інтелекту, і саме - різні варіації  $A^*$ -пошук на графах, в основу яких закладено принцип ітерационного обходу з використанням евристик. До теперішнього часу розроблені і досліджені різні модифікації алгоритму  $A^*$  для вирішення приватних завдань. Однак жоден із запропонованих алгоритмів не в змозі забезпечити планування траєкторії за

довільно короткий проміжок часу, поряд з урахуванням динаміки навколишнього середовища і її часткової наблюдаємостю. Також загально визнано, що застосування ітераційного підходу при побудові траєкторії характеризується надмірним використанням обчислювальних ресурсів. Дані обставини істотно обмежують застосування відомих методів і алгоритмів планування траєкторії на практиці. Для вирішення зазначених проблем пропонується використовувати графи спеціальної структури - МТ-графи, які дозволяють формалізувати знання про предметну область і використовувати їх при плануванні. У статті описаний новий алгоритм побудови траєкторії на площині HGA \*, який використовує МТ-графи в якості формальної моделі. Алгоритм HGA \* реалізує принципи ієрархічного планування і являє собою надзвичайний-но гнучкий і ефективний метод побудови траєкторії. При цьому HGA \* дозволяє здійснювати планування в динамічній, частково спостерігається середовищі, а також враховувати обмеження об'єкта управління. Обчислювальна ефективність HGA \* і перевага алгоритму над існуючими аналогами б-чи підтверджені результатами проведених експериментів.

## Розділ 3

### Практична частина

#### 3.1. Побудова безпілотного літального апарату на базі польотного контролера APM 2.6

Останнім часом високу ступінь інтересу викликають дрони або безпілотні літаючі роботи, побудовані на базі Мультикоптер. Серед них найбільш поширеним є чотиримоторний коптер - квадрокоптер. Зараз вони використовуються для дослідження місцевості, для проведення рятувальних операцій МНС, в роботі пожежних служб, військової розвідки. Вони можуть бути використані для доставки невеликих вантажів на відстань не більше 5-7км зі швидкістю 30-40км / год. Застосовуються фотоапаратами та операторами для зйомок пейзажів, архітектури, відеороликів. В даний час розроблено велику кількість польотних контролерів з програмним забезпеченням. Це контролери Multiwii, ArduCopter (APM 2.6, APM 2.8), контролери DJI (Naza-M Lite, DJI Naza-M V2, DJI Wookong), MicroKopter, Zero UAV X4 / X6, AutoQuad, KK., XAircraft і ін. Метою роботи є побудова БПЛА, який здатний виконувати такі основні польотні режими:

1. Stabilize - в цьому режимі виконується зліт і посадка. Задіяні гіроскоп і акселерометр для утримання горизонту. Компас використовується додатково для контролю і корекції.

2. AltHold - режим утримання висоти. В даному режимі додається використання барометра, який сприяє утриманню висоти по тиску повітря.

3. Land - режим автоматичної посадки в поточному положенні. Використовується барометр для контролю висоти.

4. Simple - режим, який дозволяє «забути» про орієнтацію БПЛА щодо пілота. В даному режимі найважливішим є компас.

5. Loiter - режим утримання точки (по координаті і висоті). Використовує GPS. Режим добре підходить для фото-і відеозйомки.

6. RTL (Return To Launch) - повернення додому, в точку зльоту. Контролер запам'ятовує точку, де зроблений Arming і дозволяє повернути коптер в цю точку.

7. Auto - політ по точкам місії. Місія може створюватися вручну через програмне забезпечення наземної станції перед польотом.

8. Failsafe - режим порятунку, який відправляє БПЛА додому (в точку, де проведений запуск двигунів - Arming). Наприклад, в разі втрати зв'язку з наземною станцією.

БПЛА повинен бути побудований на базі Мультикоптер з 4-ма двигунами з рамою типорозміру F450 і розташуванням моторів QUAD X.

Для вирішення поставленого завдання необхідний правильний вибір польотного контролера спільно з програмним забезпеченням для його функціонування. Тільки невелика частина перерахованих вище контролерів в змозі виконати поставлене завдання. Наприклад, найдешевший контролер KK має мінімальний набір датчиків (3-х осьовий гіроскоп), мікроконтролер ATmega168 малої продуктивності, обмеженою пам'яттю і периферією, що дозволяє йому реалізувати лише режим Stabilize. Контролер DJI Wookong відомої китайської приватної компанії SZ DJI Technology Co., Ltd. (Dajiang Innovation Technology Co.) дозволяє виконувати всі перераховані польотні режими, однак він має високу вартість і закрите програмне забезпечення. Наприклад, неможливо змінити код програми в разі установки додаткових пристроїв, які надавали б вплив на роботу рухової установки. У таблиці 1 представлені можливості найбільш поширених польотних контролерів.

Таблиця 1. Можливості польотних контролерів

Контролер	Стабілізація польоту	Утримання висоти	Утримання позиції	Політ по точкам
MultiWii	+	+	+	+(?)
ArduCopter	+	+	+	+
Rabbit	+	+	+	-
DJI Naza Lite	+	+	+	-
DJI Naza V1/V2	+	+	+	+
DJI Wookong	+	+	+	+
Zero UAV X4/X6	+	+	+	+
XAircraft	+	+	+	-
XAircraft SuperX	+	+	+	-
KK	+	-	-	-
MicroKopter	+	+	+	+
Autoquad	+	+	+	+

Відносно недорогими з відкритим вихідним кодом є контролери MultiWii (на базі ATmega2560) і ArduCopter, які підтримують всі задані польотні режими (в програмному забезпеченні MultiWii явно реалізована можливість утримання позиції і повернення в точку вильоту за допомогою GPS. Політ по точкам явно не документований. Однак контролери MultiWii виконують політ по точках з використанням прошивки MegaripateNG. Але це портований ArduCopter, який виконує політ по точках в разі використання ПЗ наземної станції Mission Planner). Тому в роботі розглядається побудова

БПЛА на базі ArduCopter. В даний час проект ArduCopter підтримує два типи контролерів - APM 2.x і Pixhawk. Представлені також нові розробки - це контролери NAVIO + і NAVIO2 на основі одноплатних міні комп'ютерів Raspberry Pi 2, Raspberry Pi 3, які працюють на базі операційної системи Linux. Але вони знаходяться в стадії експериментальної відпрацювання. Контролер APM 2.x побудований на основі 8-й бітного мікроконтролера ATmega2560, Pixhawk - на основі 32-х розрядного високопродуктивного процесора STM32F427 з вбудованим співпроцесором. Другий контролер має суттєві переваги по продуктивності, обсягу пам'яті в порівнянні з першим, проте набір польотних режимів у них однаковий і контролер Pixhawk має перевагу для побудови швидкісних літальних апаратів з подальшим розширенням їх функціональних можливостей в майбутньому. Вартість Pixhawk приблизно в 1.6 ... 2 рази вище APM 2.x. Тому в роботі розглядається використання контролера APM 2.5 / 2.6.

На Рис. 1 представлений контролер APM 2.5 / 2.6 з позначенням назв компонентів.

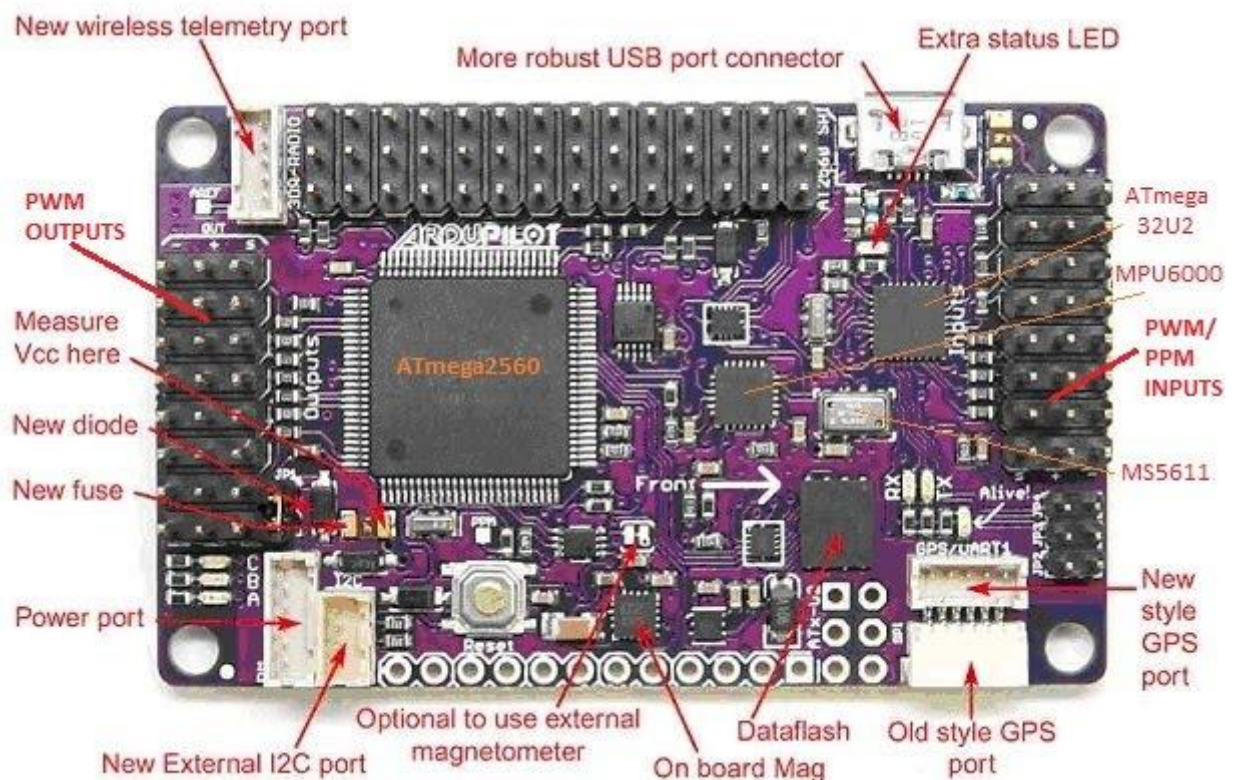


Рис. 1. Польотний контролер APM 2.5 / 2.6



Контролер включає в себе:

1. MPU-6000 - 3-х осьовий гіроскоп і 3-х осьовий акселерометр для утримання горизонту.
2. Мікроконтролер ATmega32U2, для виконання функцій PPM encoder і USB інтерфейсу.
3. Барометричний датчик для утримання висоти MS5611.
4. Флеш пам'ять 4Мбайт для запису польотних даних (ведення логів польоту).
5. Мікроконтролер ATmega2560 для управління польотом.

Відмінність APM 2.6 від APM 2.5 в тому, що в першому відсутній компас. Для його кращої роботи він винесений в зовнішній модуль і встановлений з GPS приймачем (малюнок 2).



Рис.2. GPS модуль u-blox NEO-6M з компасом

Для управління безколекторними двигунами використовуються регулятори ESC (Electronic Speed Controller) моделі Hobbypower ESC-30A. Функціональна схема підключення цих регуляторів до польотного контролера представлена на малюнку 3. Вони управляються сигналом прямокутної форми з тривалістю імпульсу від 1мс до 2мс, яка визначає швидкість обертання двигунів.

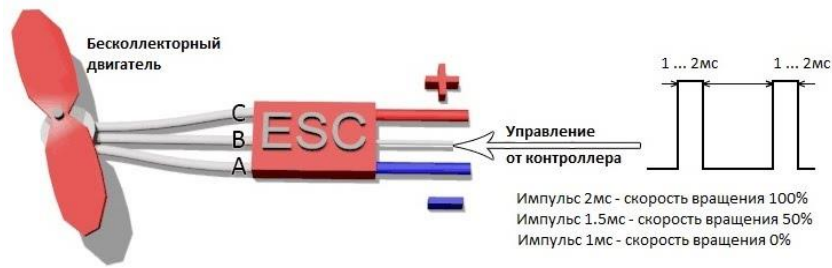


Рис.3. Схема підключення регулятора двигуна і його функціонування.

На малюнку 4 представлений регулятор Hobbypower ESC-30A з вхідними туди електронними компонентами.

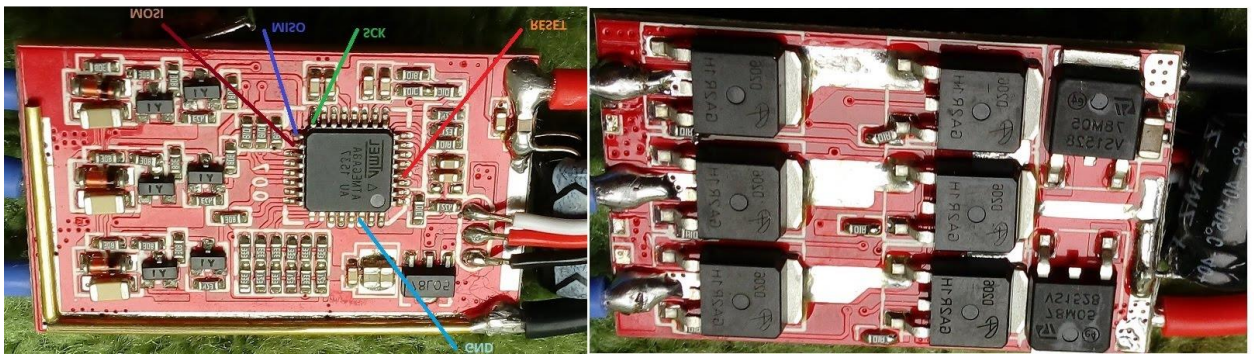


Рис.4. Регулятор Hobbypower ESC-30A

У роботі в якості основного джерела живлення використовуються 3 послідовно підключені високотокові акумуляторні li-ion батареї типорозміру 18650 - SAMSUNG INR18650-25R замість стандартних літій полімерних батарей (Lipo). Вони здатні в режимі безперервного розряду віддавати в навантаження струм величиною 20А, а при імпульсному до 30А, що досить для польотних режимів AltHold, Loiter, Land, RTL і польоту по точкам в режимі Auto зі швидкістю до 5м / с. Використовувані регулятори за замовчуванням мають настройки для використання Lipo, зокрема виконують автоматичне вимикання двигунів при зниженні напруги на одну батарею до 3.15В. Однак li-ion батареї дозволяють знижувати напругу до 2.9В, тому потрібно перепрограмування використовуваних регуляторів ESC. Існує два основних способи програмування ESC - за допомогою карти і пульта. Перший спосіб вимагає використання спеціальної картки вартістю \$ 5, \$ 15,

другий вимагає використання пульта управління БПЛА. В роботі використовується другий спосіб програмування.

Розглянемо існуючі програмовані функції, які використовуються в роботі ESC:

1. Гальмо двигуна: вкл / викл, за замовчуванням відключений. При включеному гальмі йде гальмування двигунів при їх відключенні.

2. Тип акумулятора: Li-xx (літій-іонний або Lipo) / Ni-xx (NiMH або NiCd). За замовчуванням Li-xx.

3. Режим відсічення подачі напруги (Cut-Off Mode): М'яка Cut-Off (поступово зменшити потужність на двигун) або Cut-Off (повна відсічення вихідної потужності). За замовчуванням - м'яка Cut-Off.

4. Низький поріг захисту напруги (Cut-Off Threshold): низький / середній / високий, за замовчуванням середній.

4.1. Для літєвих батарей, кількість акумуляторних батарей розраховується автоматично. Низький / середній / високий напруги відсічення для кожного осередку наступні: 2.85V / 3.15V / 3.3V. Наприклад: Для 3 банкових літєвих батарей, коли встановлений "середній" поріг відсічення, напруга відсічення буде:  $3.15 * 3 = 9.45V$ .

4.2. Для нікелевих батарей, низький / середній / високий напруги відсічення - це 0% / 50% / 65 при початковій напрузі акумулятора. Так 0% означає, що функція низького напруження відсічення відключена. Наприклад: Для 10 осередків NiMH акумулятора, напруга повністю зарядженої становить  $1,44 * 10 = 14,4$ , коли встановлений "середній" поріг відсічення, напруга відсічення буде:  $14,4 * 50\% = 7,2 V$ .

5. Старт двигуна: Normal / Soft / Super-Soft. (300 мс / 6s / 12 сек), за замовчуванням Normal. Нормальний - це найбільш прийнятний для літаків. М'який або супер-м'який - кращі для вертольотів, коли необхідно поступове прискорення обертання пропелера.

6. Таймінг -. Низький / Середній / Високий (3,75 ° / 15 ° / 26,25 °), за замовчуванням низький. Як правило, низький або середній таймінг підходить

для більшості двигунів. Для того щоб отримати більш високу швидкість і більшу вихідну потужність, вибирають високий таймінг.

Досвід експлуатації квадрокоптера показав доцільним наступний вибір налаштувань:

Brake: OFF. Гальмо вимкнений.

Battery Type: Ni-xx (NiMH or NiCd). Чи не вибирається тип батарей Li-Po, тому що регулятор зупинить мотори, коли напруга акумулятора просяде, і БПЛА впаде.

CutOff Mode: Soft-Cut. При плавному виключенні мотора контролер скидає оберти поступово.

CutOff Threshold: Low. Мотор буде вимкнений тільки при досягненні мінімальної напруги на акумуляторі.

Start Mode: Normal. Вибираємо перше значення з доступних. М'який і супер-м'який старт не рекомендується.

Timing: MEDIUM. Параметр, від якого залежить потужність і ККД двигуна. Може знаходитися в межах від 0 ° до 30 °. Фізично це електричний кут випередження комутації обмоток. Практично програмування виконується наступним чином:

1. Відключається квадрокоптер і від'єднуються три мотора від регуляторів ESC.

2. Чи включається передавач і рукоятка регулювання обертів двигуна (газ) встановлюється на максимум. Після цього включається квадрокоптер.

3. Мотор підключеного регулятора почне пищати. Спочатку два коротких писку, потім мелодія, що говорить про перехід в режим програмування. Після цього починаються сигнали-питання і сигнали-відповіді.

- сигнали-питання - це варіанти з точок (коротких сигналів) і тире (довгих):. ... .. - - - .. - Далі виконується безпосереднє програмування.

- одна точка це тип гальма регулятора. Тут необхідно "увійти" в режим вибору відповіді - різким зменшенням газу до нуля. Регулятор запропонує

варіанти відповіді: точка = гальмо викл, дві точки - гальмо вкл. Необхідно вимкнути гальмо, тому як тільки буде відповідь "точка", так відразу газ повертаємо на максимум. Регулятор підтверджує вибір короткою мелодією і починає запитувати далі.

- дві точки це тип акумулятора. Входимо в це меню скиданням газу до нуля, потім чекаємо двох точок (NiMh акумулятор), і повертаємо газ на максимум.

- три точки - режим відсікання. Вибираємо Soft-cut, очікуємо одну точку і піднімаємо газ до максимуму.

- чотири точки - поріг відсічення по напрузі акумулятора. Вибираємо одну точку, що для NiMH акумулятора відповідає 0% від напруги при старті.

- тире - режим старту. Вибираємо Normal, одна точка.

- тире точка - таймінг. Вибираємо Medium, дві точки.

- тире точка точка - це скидання налаштувань. Це необхідно пропустити.

- тире тире - це вихід. Виконується скидання газу до нуля. Після цього прозвучить один сигнал, відповідний кінця програмування регулятора. На малюнку 5 в табличній формі представлені значення програмованої Функції регулятора.

мелодія / функція	"бип-" 1 короткий сигнал	"бип-бип-" 2 короткі сигнали	"бип-бип-бип -" 3 короткі сигнали
тормоз	выкл	вкл	
тип батареи	Li-ion / Li-poly	NiMh / Nicd	
режиме отсечки	Уменьшить мощность	отключение	
порог отсечки	низкий	средний	высокий
Режим старта	нормальный	плавный	очень плавный
тайминг	низкий	средний	высокий

Рис.5. Значення програмованої функції

Для планування місії польоту БПЛА по подорожнім точкам використовується програмне забезпечення наземної станції Mission Planner. При правильному налаштуванні приймача GPS у вкладці програми FLIGHT



PLAN висвічується карта місцевості з поточним становищем коптера. Далі на мапі задаються положення шляхових точок з їх параметрами (наприклад, висотою польоту, часом зависання над точкою, необхідністю виконання будь яких дій над точкою і т.д.) і в пам'яті контролера записується маршрут. Приклад простого маршруту місії представлений на малюнку 6.

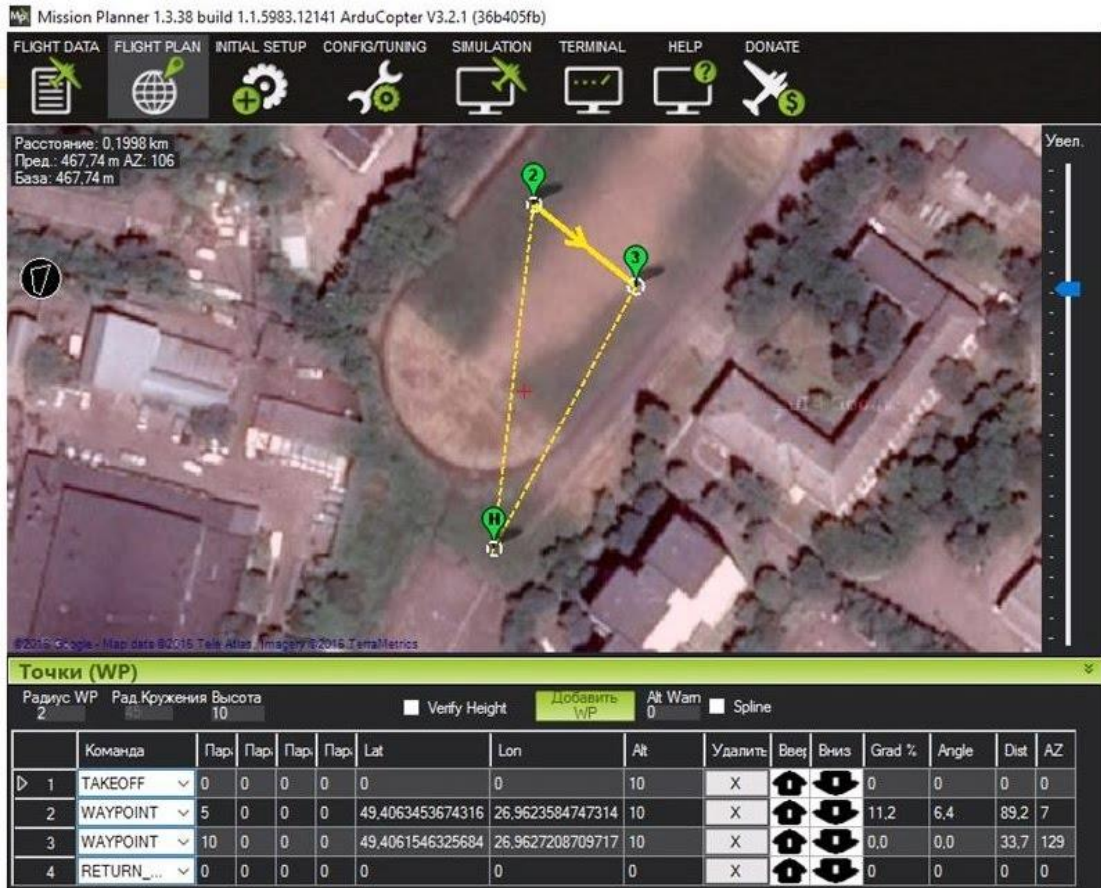
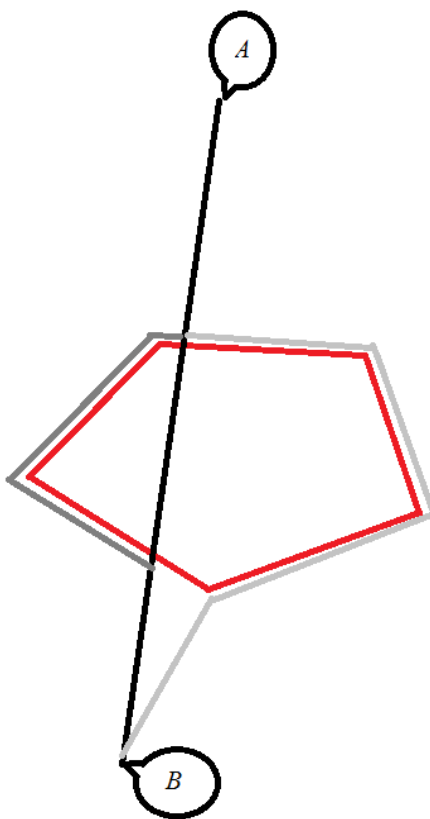


Рис.7. Планування місії БПЛА по подорожнім точкам в ПО Mission Planner

### 3.2. Модифікація модуля маршрутизації на основі модуля пошуку найкоротшого шляху

Завдання визначення найкоротшого шляху для безпілотних літальних апаратів зазвичай формулюється гранично просто: найкоротший шлях для літального апарату - пряма. Однак в ряді випадків доводиться змінювати цьому правилу, наприклад в разі забороненої для польотів зони (наприклад, військового об'єкту, або аеропорту). У таких випадках пропонується використовувати один з варіантів пошуку найкоротшого шляху, що ходять по заборонений об'єкт по найкоротшій траєкторії.

Таким чином, завдання дослідження полягає в пошуку найкоротшого шляху обходу забороненого об'єкту, заданого вершинами багатокутника на координатній сітці (рис. 3.8).



- Рис. 3.8 - Обхід забороненої зони
- Типи журналів (пам'ять польотного контролера або tlogs - телеметричний журнал)

- Є два шляхи для записи ваших польотних даних з ArduCopter, ви можете вибрати будь-який із шляхів: Журнал польотного контролера (dataflash) – використовується на платі польотного контролера arduino c 16Мб пам'яті на борті, ви зможете скачати після польоту.
- Журнал телеметрії (звісний як "tlogs" описаний на цій сторінці) може бути записаний за допомогою ПО Mission Planner коли ви підключені до вашого АРМ за допомогою телеметричних радіомодемів, або USB.

### **Телеметричні логи (tlogs)**

Tlogs записуються з телеметричного протоколу MAVLink обмінюється повідомленнями між польотним контролером і наземною станцією і автоматично створюються, коли ви натискаєте в ПО Mission Planner кнопку "connect".

Формат файлів іменується як YYYY-MM-DD hh-mm-ss.tlog і перебуває в папці вашого ПО Mission Planner або в заздалегідь зазначеної вами папці в налаштуванні ПЗ Config / Tuning-> Planner.

Крім файлів ".tlog" створюються файли ".rlog". Вони містять всі дані .tlog з додатковим налагодженням з Mission Planner. Вони не можуть бути розібрані або проаналізовані тому ви повинні їх ігнорувати.

відтворення Міс

Якщо ви використовуєте бездротову телеметрію і у вас встановлене з'єднання між польотним контролером АРМ і ПО Mission Planner ваші польотні дані будуть автоматично записані. Ви можете пролистати їх з журналів наступним способом:

- Щоб відкрити вікно Flight Data
- Перейдіть на вкладку "Telemetry logs"
- Натисніть кнопку "Load Log" і знайдіть польотний tlog файл



Натисніть

кнопку

"Play"



Можливо встановити часовий інтервал повзунок і вибрати швидкість відтворення натискаючи клавiші швидкості.

Пролистуючи журнал інформація екрану HUD буде змінюватися і місце розташування апарата буде оновлюватися як це відбувається в польоті. Індивідуальні значення можливо побачити через вкладку Status і відобразити на графіку натиснувши на "Tuning". Потім подвійним клацанням на даних викликати вікно з додатковими параметрами, які вам буде потрібно

відобразити. Це покаже записані дані в процесі польоту.



Створення 3D шляху польоту.

Ви можете створити KML файлі роблячи наступне:

- Під час використання екран "Flight Data"
- Перейдіть на вкладку "Telemetry Logs"
- Натисніть на кнопку "Tlog> KML or Graph"
- Натисніть на кнопку "Create KML + GPX"

Виберіть

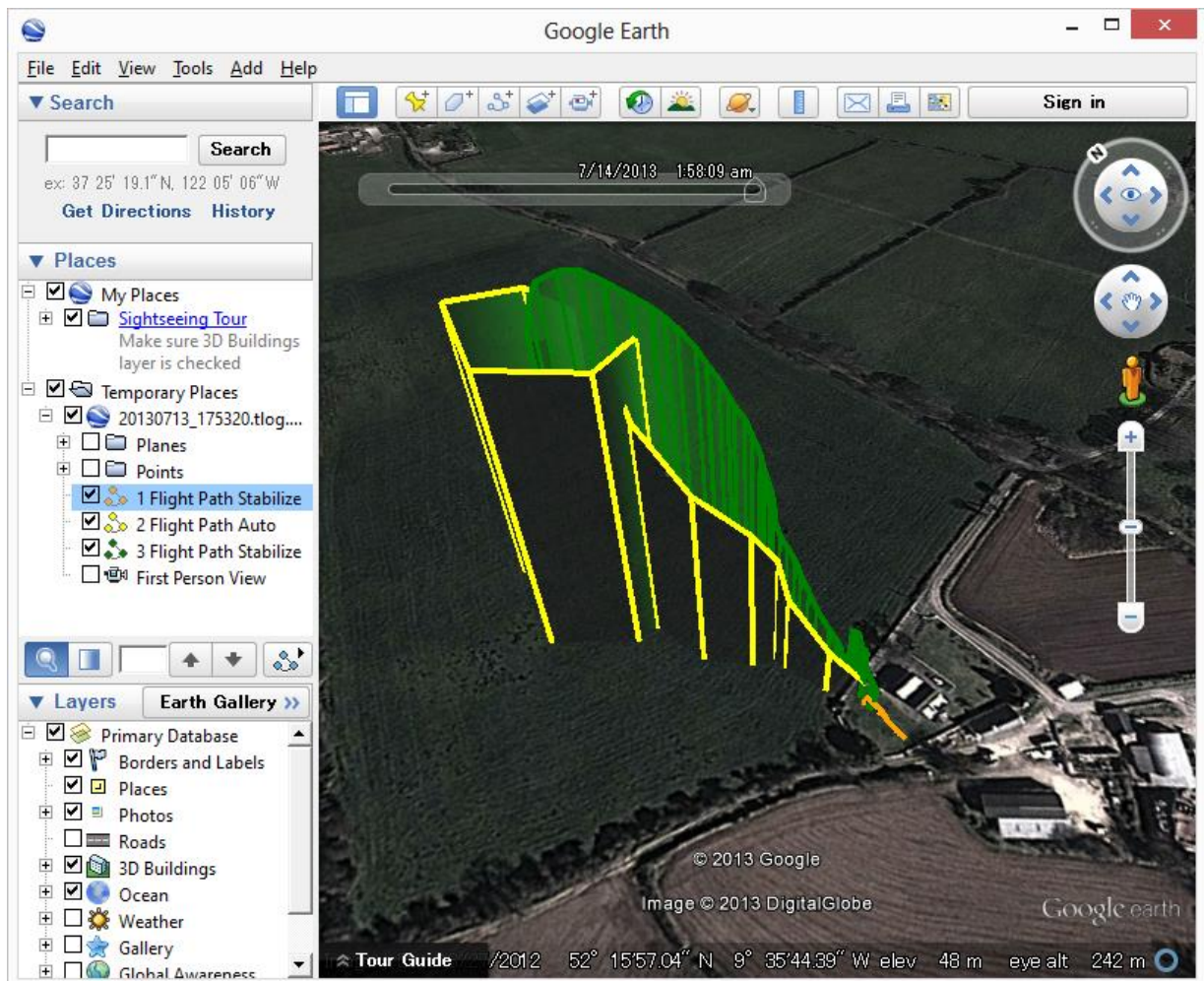
польотний

tlog

файл



Файли .kmz і .kml будуть створені поруч з оригінальним .tlog і їх можна відкрити в Google Earth для інтерактивного перегляду 3D польоту. Ви можете відкрити kmz файл в Google Earth для перегляду польоту або шляху. Просто двічі клацніть на файл або перетягніть його в Google Earth.



Різні режими польоту будуть відображені різними кольорами доріжок. Ви можете змінити деталі відображення шляху і кольору виконуючи наступне:

- Знайдіть файл журналу в панелі зліва "Places". Він повинен з'явитися в папці "Temporary Places"
- Клацніть на індивідуальний шлях і виберіть "Properties" для відкриття вікна редакування шляху
- Колір можливо поміняти на вкладці "Style, Color"
- Область нижче може бути видалена / додана на вкладці "Altitude" зняттям / установкою "Розширений шлях до землі"

Витяг параметрів і шляхових точок



Ви можете отримати параметри і точки з tlog слідуючи тим же діям, що і для створення KML файлу виключаючи останній крок Виберіть "Extract Params" або "Extract WP's"

Витяг параметрів створить .param файл який буде поруч з tlog. Цей файл (.param) містить повний список параметрів (в тому ж порядку, як вони з'являються в EEPROM) Ра з їх значеннями під час польоту. Вони можуть бути відкриті в екселі або текстовим редактором.

Витяг шляхових точок створить один або кілька .txt файлів будь-яких місій завантажених на польотний контролер APM. Ці файли можуть бути відкриті в Mission Planner перемкнувшись на екран "Flight Plan", правою кнопкою миші на карті вибрати "File load / save", "Load WP File".

---

#### Графічні дані з польоту

Дані з польоту можуть бути відображені графічно, для цього виконайте наступне:

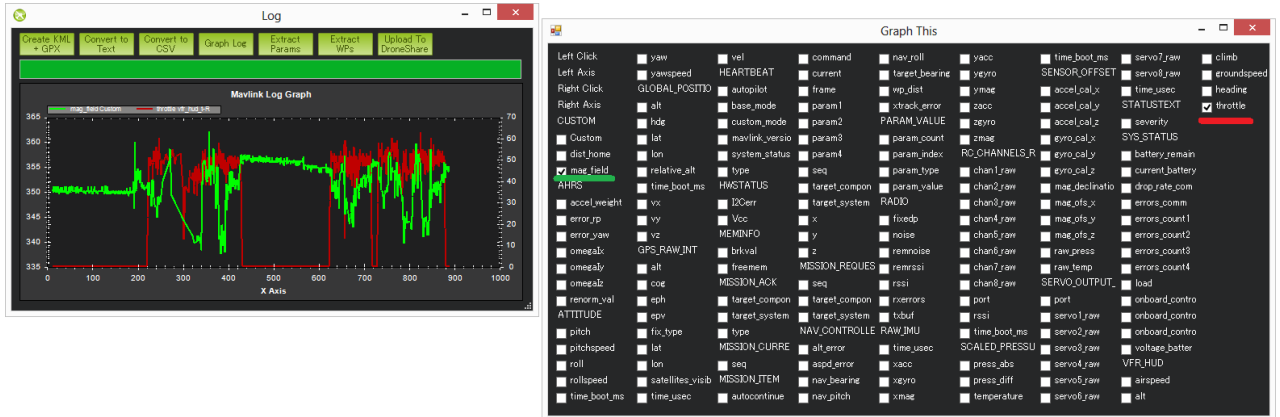
- Під час використання Flight Data екран
- Натисніть на вкладку "Telemetry Logs"
- Натисніть на кнопку "Tlog> KML or Graph"
- Натисніть на кнопку "Graph Log"

- Виберіть потрібний файл польоту log



- Коли з'явиться екран "Graph This" використовуючи кнопку миші виберіть потрібний пункт на основі якого ви бажаєте побудувати графік. Зверніть увагу, що елементи згруповані за категоріями, таких як "RC\_CHANNELS" і "RAW\_IMU".
- Якщо ви використовуєте ліву кнопку миші масштаб для елемента з'явиться зліва від графіка. Якщо праву, то справа.
- Встановлюйте прапорець кілька разів що б вибрати необхідний колір.
- Змінити величину графіка можна з допомогою колеса миші, вибираючи область графіка затисніть ліву клавішу або клацнувши правою

кнопкою на графіку виберіть "Set Scale To Default" для скидання величини.



Режим пошуку найкоротшого шляху



Підключивши модуль sway.cpp (init.ini - use sway) потрапляємо режим автоматичного визначення безпілотних зон і пошуку шляхів обходу.

Результати, показані алгоритмами WA \* -3 і WA \* -5, практично ідентичні, з чого можна зробити висновок, що подальше підвищення ваги евристики не призведе до скорочення кількості розглянутих клітин. Отже, можна припустити, що алгоритм HGA \* в середньому в 1,5-4 рази ефективніше будь-яких наявних аналогів, в основу яких закладено принцип ітерационного A \* -Пошук.

Завдання планування траєкторії на площині є однією з ключових при розробці систем управління безпілотними транспортними засобами нового

покоління. Для вирішення цього завдання традиційно використовуються методи штучного інтелекту, і саме - різні варіації  $A^*$ -пошук на графах, в основу яких закладено принцип ітерационного обходу з використанням евристик. До теперішнього часу розроблені і досліджені різні модифікації алгоритму  $A^*$  для вирішення приватних завдань.

Метод успішно запрограмований і вбудований в вільно розповсюджується Mission Planner, що дозволяє використовувати автоматичні пошукові алгоритми при прокладанні маршруту БПЛА.



## Розділ 4

### ЕКОНОМІЧНА ЧАСТИНА

#### РОЗДІЛ 4. ЕКОНОМІКА

- 1) Передбачуване число операторів - 356.
- 2) Коефіцієнт складності програми - 1,6.
- 3) Коефіцієнт корекції програми в ході її розробки - 0,9.
- 4) Коефіцієнт збільшення витрат праці унаслідок недостатнього опису завдання - 1,3.
- 5) Коефіцієнт кваліфікації програміста - 0,8.
- 6) Середня годинна заробітна плата програміста – 30 грн / год.
- 7) Вартість машино-годин ЕОМ - 8,5 грн / год.

#### 4.1. Визначення трудомісткості розробки програмного забезпечення

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_{\partial} , \text{людино – годин}$$

де  $t_o$  – витрати праці на підготовку й опис поставленої задачі (приймається 50);

$t_u$  – витрати праці на дослідження алгоритму рішення задачі;

$t_a$  – витрати праці на розробку блок-схеми алгоритму;

$t_n$  – витрати праці на програмування по готовій блок-схемі;

$t_{oml}$  – витрати праці на налагодження програми на ЕОМ;

$t_{\partial}$  – витрати праці на підготовку документації.

$$t = 50 + 33,82 + 56,36 + 56,36 + 124,6 + 391,81 = 712,95 , \text{людино-годин}$$

Складові витрати праці визначаються через умовне число операторів у ПЗ, яке

розробляється.

Умовне число операторів (підпрограм):

$$Q = q \cdot C \cdot (1 + p),$$

де  $q$  – передбачуване число операторів;

$C$  – коефіцієнт складності програми (в інтервалі від 1.25 до 2.0);

$p$  – коефіцієнт корекції програми в ході її розробки (в інтервалі від 0.05 до  $Q = 356 \cdot 1,6 (1 + 0,9) = 1082,24$

Витрати праці на вивчення опису задачі  $t_u$  визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75 \dots 85)}, \text{ ЛЮДИНО-ГОДИН}$$

де  $B$  – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі (в інтервалі від 1.2 до 1.5);

$k$  – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності.

Таблиця 3.1 – Значення коефіцієнта кваліфікації програміста

Стаж програміста	Значення коефіцієнта $k$
до 2-х років	0,8
від 2 до 3 років	1,0
від 3 до 5 років	1,1 ... 1,2
від 5 до 10 років	1,2 ... 1,3
понад 10 років	1,3 ... 1,5

$$t_u = \frac{1082,24 \cdot 1,9}{76 \cdot 0,8} = 33,82, \text{ ЛЮДИНО-ГОДИН}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20 \dots 25)}, \text{ ЛЮДИНО-ГОДИН}$$

$$t_a = \frac{1082,24}{24 \cdot 0,8} = 56,36 \text{ ЛЮДИНО-ГОДИН}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20 \dots 25) \cdot k}, \text{ ЛЮДИНО-ГОДИН}$$

$$t_n = \frac{1082,24}{24 \cdot 0,8} = 56,36 \text{ ЛЮДИНО-ГОДИН}$$

Витрати праці на налагодження програми на ЕОМ:

— за умови автономного налагодження одного завдання:

$$t_{oml} = \frac{Q}{(4 \dots 5) \cdot k}, \text{ ЛЮДИНО-ГОДИН}$$

$$t_{oml} = \frac{1082,24}{4,23 \cdot 0,8} = 319,81, \text{ ЛЮДИНО-ГОДИН}$$

— за умови комплексного налагодження завдання:

$$t_{oml}^k = 1,5 \cdot t_{oml}, \text{ ЛЮДИНО-ГОДИН}$$

$$t_{oml}^k = 1,5 \cdot 319,81 = 479,715 \text{ ЛЮДИНО-ГОДИН}$$

Витрати праці на підготовку документації:

$$t_{\partial} = t_{\partial p} + t_{\partial o}, \text{ ЛЮДИНО-ГОДИН}$$

де  $t_{\partial p}$  – трудомісткість підготовки матеріалів і рукопису.

$$t_{\partial p} = \frac{Q}{(15 \dots 20) \cdot k}, \text{ ЛЮДИНО-ГОДИН}$$

$$t_{\partial p} = \frac{1082,24}{19 \cdot 0,8} = 71,2 \text{ ЛЮДИНО-ГОДИН}$$

$t_{\partial o}$  – трудомісткість редагування, печатки й оформлення документації

$$t_{\partial o} = 0,75 \cdot t_{oml}, \text{ ЛЮДИНО-ГОДИН}$$

$$t_{\partial o} = 0,75 \cdot 71,2 = 53,4 \text{ ЛЮДИНО-ГОДИН}$$

$$t_{\partial} = 53,4 + 71,2 = 124,6 \text{ ЛЮДИНО-ГОДИН}$$

#### 4.2. Витрати на створення програмного забезпечення

Витрати на створення ПЗ  $K_{no}$  включають витрати на заробітну плату виконавця програми  $Z_{3n}$  і витрат машинного часу, необхідного на налагодження програми на ЕОМ.

$$K_{no} = Z_{zn} + Z_{mv} \text{ ,грн}$$

$$K_{no} = 21388,5 + 71,2 = 21459,2 \text{ грн}$$

Заробітна плата виконавців визначається за формулою:

$$K_{zn} = t + C_{pp} \text{ ,грн}$$

де  $t$  – загальна трудомісткість, людино-годин;

$C_{pp}$  – середня годинна заробітна плата програміста, грн/година

$$K_{zn} = 712,95 \cdot 30 = 21388,5 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$Z_{mv} = t_{oml} \cdot C_{mч} \text{ ,ГРН}$$

де  $t_{oml}$  – трудомісткість налагодження програми на ЕОМ, год.

$C_{mч}$  – вартість машино-годин ЕОМ, грн/год.

$$Z_{mv} = 71,2 \cdot 8,5 = 605,2 \text{ грн.}$$

Визначені в такий спосіб витрати на створення програмного забезпечення є частиною одноразових капітальних витрат на створення АСУП.

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p} \text{ , міс.}$$

де  $B_k$  – число виконавців;

$F_p$  – місячний фонд робочого часу (при 40-годинному робочому тижні  $F_p=176$  годин).

$$T = \frac{712,95}{2 \cdot 176} = 2 \text{ міс.}$$

Вартість даного програмного забезпечення становить 21459,2 грн. і не вимагає додаткових витрат як при розробці програми, так і при її експлуатації. Очікуваний час розробки становить 4 місяці.

#### **4. 3. Маркетингові дослідження ринку збуту розробленого програмного продукту**

Вже найближчим часом аграрний сектор стане одним з основних споживачів дронів. Технології стрімко розвиваються і дешевшають, на ринку з'являється все більше пропозицій як від великих виробників, так і від компаній, які займаються виключно виробництвом коптерів.

Дрони дозволяють створювати ортофотоплани, здійснювати відео та фотомоніторинг, вносити СВР і добрива. Це універсальний інструмент для аграріїв, який може покривати до 1000 га в день.

Чималі вигоди обіцяє і диференційоване внесення добрив, застосування ЗЗР. В даному випадку кошти хімізації вносяться строго там, де вони потрібні, і в оптимальному дозуванні. Машинобудівники вже досить давно пропонують кошти для реалізації цієї технології, однак на практиці найбільшу складність являє собою збір інформації, який або передбачає суттєві трудовитрати (як при складанні ґрунтових карт шляхом відбору проб), або є недостатньо оперативним і не володіє достатньою глибиною (як у випадку із супутниковим моніторингом).

Метою дипломної роботи є дослідження оптимізації програмного забезпечення дронів за допомогою програми APM Planner, в сільському господарстві. У даній темі в якості оптимізації дрона буде розглянута програма APM Planner, тому що це ключовий спосіб для програмування дрона, завдання йому алгоритмів системи управління і установка поставленої перед ним завдання. Так само виявлення плюсів і мінусів для порівняння і вибору оптимальної програми.

Наукова новизна результатів роботи визначається наступними отриманими автором теоретичними і практичними результатами:

- 1) Вдосконалено алгоритм польоту дрона. Тепер політ дрону обмежен спеціальними «критичними зонами», котрі він не буде перетинати і не зможе вилетіти за рамки своєї траєкторії.
- 2) Підвищен захист даних дрону. Тепер він має свою чітку траєкторію польоту запрограмовану заздалегідь. Це означає, що його

не зможуть «перехватити» злочинці, адже він не має доступу до його «мозгів».

3) Підвищена стабілізація дрону. Завдяки цьому, дрон зможе облітати різноманітні перешкоди, будь-то які дерева, гілки, кущі.

В мережі інтернет є багато різноманітних програмних продуктів для програмування дронів. Є як платні, так безкоштовні, але, на думку автора, для кожної конкретної задачі необхідно своє налаштування необхідних замовнику характеристик. Було проаналізовано багато програмних продуктів на виявлення недоліків традиційного підходу до програмування дронів, та розроблено оптимальне рішення для моніторингу в сільському господарстві.

#### 4.4. Оцінка економічної ефективності впровадження програмного забезпечення

Таблиця 4.1. Розрахунок чистих грошових надходжень від розробки ПЗ

Показники, грн	За роками						Усього за 5 років	Середнє за 5 років
	0	1	2	3	4	5		
<b>Інвестиції на ПЗ</b>	<b>21459</b>	-	-	-	-	-	<b>21459</b>	<b>4292</b>
<b>Витрати до впровадження ПЗ</b>	-	<b>15300</b>	<b>15000</b>	<b>15700</b>	<b>15400</b>	<b>18100</b>	<b>79500</b>	<b>15900</b>
- на щорічну перевірку на погрішність	-	2500	2800	3000	3300	3600	15200	3040
- на щорічну перевірку держстандарту	-	4000	4000	4000	4000	4000	20000	4000
- на консультацію з іншими спеціалістами	-	4800	4000	4200	4400	4600	22000	4400
- на електроенергію	-	6000	6200	6400	6600	6800	32000	6400
<b>Витрати після впровадження ПЗ</b>	-	<b>4500</b>	<b>4700</b>	<b>4850</b>	<b>5000</b>	<b>5150</b>	<b>24200</b>	<b>4840</b>
- на щорічну перевірку на погрішність	-	500	600	650	700	750	3200	640
- на щорічну перевірку держстандарту	-	1000	1000	1000	1000	1000	5000	1000

- на консультацію з іншими спеціалістами	-	500	500	500	500	500	2500	500
- на електроенергію	-	2500	2600	2700	2800	2900	13500	2700
Економія	-	10800	10300	10850	10400	12950	55300	11060
Амортизація	-	4292	4292	4292	4292	4292	21459	4292
Чисті грошові надходження	-	6508	6008	6558	6108	8658	33841	6768
Коефіцієнт дисконтування	-	0,83	0,756	0,658	0,572	0,497	-	-
Дисконтові грошові надходження	-	5401	4542	4315	3493	4147	21898	4379

### Коефіцієнти економічної ефективності

Чиста поточна вартість доходів:

$$NPU = 21898 - 21459 = 439 \text{ грн} > 0$$

Строк окупності:

$$T = 21459 / 4379 = 4,9 \text{ роки}$$

Індекс прибутковості:

$$ИД = 21898 / 21459 = 1,02$$

Показник економічної ефективності (NPU - чиста поточна вартість доходів) за роки реалізації впровадження (5 років) складе 439 грн, тобто відповідає умовам ефективності, тому що  $NPU > 0$ .

Середній термін окупності капвкладень складе 4,9 року.

Індекс прибутковості за 5 років складе 1,02, тобто  $ИД > 1$ , проект слід прийняти.

Таким чином, показник ефективності свідчить про те, що дане впровадження є економічно вигідним.

## Список використаних джерел

1. Салычев О.С. Автопилот БПЛА с инерциальной интегрированной системой – основа безопасной эксплуатации беспилотных комплексов. [http://www.uav.ru/articles/TeKno1\\_Autopilot.pdf](http://www.uav.ru/articles/TeKno1_Autopilot.pdf)
2. Чистяков Н.В. Что такое ДПЛА // <http://dpla.ru/Articles/WhaIstDPLA.htm>
3. Henri Eisenbeiß. UAV Photogrammetry. ETH ZURICH. DISS. ETH NO. 18515. Zurich, 2009.
4. World Robotics 2014 Service Robots [Electronic resource]: Service Robot Statistics. – IFR International Federation of Robotics. – Mode of Access: <http://www.ifr.org/service-robots/statistics/>. – [Accessed 20 Januare 2017].
5. Unmanned Systems Integrated Roadmap FY2013-2038 [Electronic resource]. – Washington, D.C.: Department of Defense, 2013. – Mode of Access: <http://www.defense.gov/pubs/DOD-USRM-2013.pdf>. – [Accessed 31 March 17].
6. Преображенский Н. Мировой рынок беспилотников [Электронный ресурс] // Военно-промышленный курьер: электрон. журн. – 2014. – № 3 (521). – Режим доступа: [http://vpk.name/news/104319\\_mirovoi\\_ryinok\\_bespilotnikov.html](http://vpk.name/news/104319_mirovoi_ryinok_bespilotnikov.html). – Заглавие с экрана. – (Дата обращения: 13.01.2017).
7. Юферев С. Перспективы развития российских БПЛА [Электронный ресурс] // Информационно-новостной портал ARMY MAN. INFO. – Заглавие с экрана. – Режим доступа: <http://armyman.info/stati/15140-perspektivu-razvitiya-rossiyskih-bpla.html>. – (Дата обращения: 13.11.2017).
8. Панков С.Е. Роль военных технологий в развитии системы вооружения Вооруженных Сил Российской Федерации [Электронный ресурс]: Управление перспективных межвидовых исследований и специальных проектов Министерства обороны Российской Федерации. – Режим доступа: <http://federalbook.ru/files/ОПК/Soderjanie/ОПК-10/III/Pankov.pdf>. – Заглавие с экрана. – (Дата обращения: 06.10.2017).



9. Мазур С. 2015: начало заката эры беспилотных летательных аппаратов [Электронный ресурс]. – Режим доступа: [http://vpk.name/news/124839\\_2015\\_nachalo\\_zakata\\_eryi\\_bespilotnyih\\_letatelnyih\\_apparatorov.html?last#last](http://vpk.name/news/124839_2015_nachalo_zakata_eryi_bespilotnyih_letatelnyih_apparatorov.html?last#last) – Заглавие с экрана. – (Дата обращения: 12.02.2017).
10. Кондратьев А. Перспективы развития и применения беспилотных и роботизированных средств вооруженной борьбы в ВС ведущих зарубежных стран [Электронный ресурс] // Зарубежное военное обозрение: информационно-аналитический иллюстрированный журнал Министерства обороны России. – 2011. – № 5. – С. 14–21. – Режим доступа: [http://pentagonus.ru/publ/perspektivy\\_razvitiya\\_i\\_primeneniya\\_bespilotnykh\\_i\\_robotizirovannykh\\_sredstv\\_vooruzhennoj\\_borby\\_v\\_vs\\_vedushhikh\\_zarubezhnykh\\_stran/19-1-0-1772](http://pentagonus.ru/publ/perspektivy_razvitiya_i_primeneniya_bespilotnykh_i_robotizirovannykh_sredstv_vooruzhennoj_borby_v_vs_vedushhikh_zarubezhnykh_stran/19-1-0-1772). – Заглавие с экрана. – (Дата обращения: 11.02.2017).
11. Попов В.А., Федутинов Д.В. Развитие направления миниатюрных беспилотных летательных аппаратов за рубежом ФГУП «ГосНИИАС» [Электронный ресурс]. – Режим доступа: <http://topwar.ru/27492-rossiyskaya-armiya-povernulas-v-storonu-robotizirovannoy-tehniki.html>. – Заглавие с экрана. – (Дата обращения: 18.11.2017).
12. A Roadmap for U.S. Robotics From Internet to Robotics [Electronic resource]. – 2013. – Mode of Access: <https://robotics-vo.us/sites/default/files/2013%20Robotics%20Roadmap-rs.pdf>.
13. The Strategic Research Agenda for Robotics in Europe [Electronic resource]: Robotic Visions to 2020 and beyond. – European Robotics Technology Platform, 07/2009 (second edition). – Mode of Access: [http://www.robotics-platform.eu/cms/upload/SRA/2010-06\\_SRA\\_A3\\_low.pdf](http://www.robotics-platform.eu/cms/upload/SRA/2010-06_SRA_A3_low.pdf).
14. Общие виды и характеристики беспилотных летательных аппаратов: справ. пособие /А.Г. Гребеников, А.К. Мялица, В.В. Парфенюк и др. – Х.: Нац.аэрокосм. ун-т «Харьковский авиационный институт», 2008. – 377 с.
15. Збірник наукових праць Харківського університету Повітряних Сил. [Текст] А.Г. Корченко, О.С. Ильяш – НАУ, 2012. – Випуск 4(33)С. 27-36.

16. Глобальная эксплуатационная концепция ОрВД [Текст]: ICAO Doc 9854 AN/458 ИКАО. – Монреаль, Канада: ИКАО, 2005. – 100 с.
17. Организация воздушного движения [Текст]: ICAO. DOC 4444 ATM/501/ИКАО. – Монреаль, Канада: ИКАО, 2007. – 474 с.
18. Поправка № 3 18/11/10 к DOC 4444. ICAO. – Монреаль, Канада: ИКАО, 2010.
18. Беспилотные авиационные системы (БАС) [Текст]: ICAO CIR 328 AN/190 ИКАО. – Монреаль, Канада: ИКАО. 2011. – 66 с.
19. Дементьев Д.О. Бойові літальні комплекси в складі єдиної інформаційно-розвідувально-навігаційної ударної системи [Текст] / Д.О. Дементьев // Зб. наук. пр. Військового інституту Київського нац. ун-ту ім. Тараса Шевченка. – К.: ВІКНУ, 2010. – Вип. 25. – С. 74-77. Открытые информационные и компьютерные интегрированные технологии № 74, 2016
20. Сальник, Ю.П. Аналіз технічних характеристик і можливостей безпілотних авіаційних комплексів оперативно-тактичного та тактичного радіуса дії армій розвинених країн [Текст] / Ю.П. Сальник, І.В. Матала // Військово-технічний зб. – 2010. – № 3. – С. 70-74.
21. Харченко, О.В. Класифікація та тенденції створення безпілотних літальних апаратів військового призначення [Текст] / О.В. Харченко, В.В. Кулешин, Ю.В. Коцуренко // Наука і оборона. – 2005. – № 1. – С. 47-54.
22. Луцький, М.Г. Розвиток міжнародного регулювання та нормативної бази використання безпілотних літальних апаратів [Текст] / М.Г. Луцький, В.П. Харченко, Д.О. Бугайко // Вісник НАУ. – 2011. – № 2. – С. 5-14. – ISSN 1813-1166.
23. Операции беспилотных летательных аппаратов в общей системе воздушного пространства [Текст] / A.Urbahs; V.Petrovs; K.Savkovs // Space and Global Security of Humanity Riga, 2010. – 21 с.
24. Ростопчин, В.В. Современная классификация беспилотных авиационных систем военного назначения [Электрон. ресурс] / В.В. Ростопчин, ООО "Техкомтех"// UAV.RU: Беспилотная авиация. – Электрон. дан. – 2003. –

Систем. требования: ПО Adobe Reader. – Режим доступа: World Wide Web. – URL: <http://www.uav.ru/articles/bas.pdf>. – Загл. с экрана (просмотрено 29.4.2017).

25. Ростопчин, В.В. Беспилотные авиационные системы [Текст]: Основные понятия / В.В. Ростопчин, И.Е. Бурдун / ЭЛЕКТРОНИКА: Наука, Технология, Бизнес. – 2009. – №4. – С. 82-88.

26. Ерохин, Е. Современная классификация российских БПЛА [Электронный ресурс] / Евгений Ерохин // MISSILES.RU: 1-й Российский сайт о ракетной технике и технологии. – Электрон. дан. – 2009. – Режим доступа: World Wide Web. – URL: [http://www.missiles.ru/UAV\\_class.htm](http://www.missiles.ru/UAV_class.htm). – Загл. с экрана (просмотрено 29.10.2017).

27. Зинченко, О.Н. Беспилотные летательные аппараты [Электрон. ресурс]: Применение в целях аэрофотосъемки для картографирования (часть 1) / О.Н. Зинченко // Ракурс: Программные решения в области геоинформатики, цифровой фотограмметрии и дистанционного зондирования. – Электрон. дан. – М., Россия: Компания «Ракурс», 2011. – Режим доступа: World Wide Web. – URL: <http://www.racurs.ru/page=681>. – Загл. с экрана (просмотрено 29.4.2017, последнее обновление 27.09.2011).

28. Беспилотные летательные аппараты [Текст] / С.М. Ганин, А.В. Карпенко, Н.Н. Колногоров, Г.Ф. Петров. – СПб.: Невский бастион, 1999. – 160 с.

29. Павлушенко, М. Беспилотные летательные аппараты: история, применение, угроза распространения и перспективы развития [Текст] / М. Павлушенко, Г. Евстафьев, И. Макаренко // Научные записки ПИР Центра: национальная и глобальная безопасность. – М.: Изд.-во «Права человека», 2005. – 612 с.

30. Современные информационные технологии в задачах навигации и наведения беспилотных маневренных летательных аппаратов [Текст]: моногр. / К.К. Веремеенко [и др.]; под ред. М.Н. Красильщикова, Г.Г. Себрякова. – М.: ФИЗМАТЛИТ, 2009. – 554 с.

31. Reg Austin. Unmanned Aircraft systems; UAVS Design, Development and Deployment / Reg Austin. –A John Wiley and Sons, Ltd. Publication –2010. –332 p. – AIAA Education Series.
32. Classification of Unmanned Aerial Vehicles [Text] / Dr. Maziar Arjomandi – MECH ENG 3016 Aeronautical engineering –The University of Adelaide Australia, 2011. –49 p.
33. Civil Aviation Safety Authority, Australia. "Unmanned Aircraft and Rocket Operations" [Text]: CASR Part 101. – Australia: CASR, January 2003. – 56 p.
- Открытые информационные и компьютерные интегрированные технологии № 74, 2016
34. Unmanned Aerial Vehicles for Rapid Environmental Assessment and Mine Countermeasures [Text]: DSTO-GD-0439 / Manuel de Sousa. – Adelaide, Australia: Defence Science and Technology Organisation "Maritime Operations Division", June 2005. – 18 p.
35. Unmanned Aerial Vehicles [Text]: Issues paper / Civil Aviation Authority of New Zealand. – 22 January 2007. – 21 p.
36. Advancing Unmanned Systems in Canada 2007-2010 [Text]/ Canadian Centre for Unmanned Vehicle Systems -2010. – 36 p.
37. Unmanned Air Vehicle Working Group Final Report [Electronic resource]: Draft / Transport Canada Civil Aviation Working Group, Inc. – Electronic data. - Transport Canada Civil Aviation Working Group, September 2007. – Software requirements: Adobe Reader 9.x or higher. – Mode of access: World Wide Web. – URL: <http://www.tc.gc.ca/eng/civilaviation/standards/general-recavi-uavworkinggroup-2266.htm>. – Description based on home page (viewed on Oct. 29, 2011). – Language: English.
38. Safety Considerations for operation of Unmanned Aerial Vehicles in the National Airspace System [Text]: Report No.ICAT-2005-1/ Roland E. Weibel and R. John Hansman; MIT International Center for Air Transportation Department of Aeronautics & Astronautics Massachusetts Institute of Technology Cambridge. – MA 02139 USA, March 2005. – 107p.

39. A Report Overview of the Civil UAV Capability Assessment [Electronic resource]: Draft / Timothy H. Cox, Christopher J. Nagy, Mark A. Skoog, Ivan A. Somers, Ryan Warner; NASA Dryden Flight Research Center; CSM, Inc. – Electronic data. — Washington, D.C., USA: NASA, 2005. -Software requirements: Adobe Reader 5.x or higher. - Mode of access: World Wide Web. - URL:  
[http://www.nasa.gov/centers/dryden/pdf/111760main\\_UAV\\_Assessment\\_Report\\_Overview.pdf](http://www.nasa.gov/centers/dryden/pdf/111760main_UAV_Assessment_Report_Overview.pdf) - Description based on home page (viewed on Oct. 29, 2011). - Language: english.
40. Chris J. Hodson. Civil Airworthiness for a UAV Control Station [Text]: This report is submitted to satisfy the project requirements of the Master of Science in Safety Critical Systems Engineering at the Department of Computer Science/Chris J. Hodson. - September 2008. - 119 p.
41. Unmanned Aircraft System Operations in UK. Airspace - Guidance [Text]: CAP 722 - Civil Aviation Authority - 6 April 2010. – 96p.
42. Unmanned Aerial Vehicles Systems Airworthiness Requirements (USAR) [Text]: STANAG 4671, NSA/0976 (2009)-JAIS/4671, Edition 1. - NSA - 9 may 2007. – 214 p.
43. Безпілотники Національного авіаційного університету [Електрон. ресурс]: Історія і сьогодення /Михаїл Матийчик. – Електрон. дані. - К.: Нац. авіац. ун-т, 2011. – Режим доступу: World Wide Web. - URL: <http://www.nau.edu.ua/uk/Science/GotoviRozrobky/bezpilotniki/> – Загл. с экрана (просмотрено 29.3.2017).
44. А-4К Альбатрос [Электрон. ресурс] / Беспилотные летательные аппараты БПЛА ДПЛА БЛА. Описания и технические характеристики беспилотников – Электронные данные - Беспилотные летательные аппараты, 8 июля 2011. – Режим доступа: World Wide Web. – URL: <http://bp-la.ru/a-4k-albatros/>- Загл. с экрана (просмотрено 29.3.2017).
45. Беспилотные аппараты [Электрон. ресурс]. Авиационная энциклопедия «Уголок неба». – Электронные данные - Авиационная энциклопедия «Уголок

неба», 2010. – Режим доступа: World Wide Web. - URL: <http://airwar.ru/bpla.html> - Загл. с экрана (просмотрено 29.10.2011).

46. Boeing Scan Eagle [Электронный ресурс] / Avia Deja vu – Электронные данные - Avia Deja vu - Режим доступа: World Wide Web. - URL: Открытые информационные и компьютерные интегрированные технологии № 74, 2016 <http://crimso.msk.ru/Site/Crafts/Craft20732.htm> – Загл. с экрана (просмотрено 29.3.2017).

47. Unmanned Aircraft System Operations in UK. Airspace [Text]: CAP 722 Document: UAVS/001, Version 01. – Civil Aviation Authority – 7 April 2011. – 10 p.

48. Сотрудничество гражданских и военных органов при организации воздушного движения [Текст]: ICAO CIR 330 AN/189 ИКАО, утверждено Генеральным секретарём и опубликовано с его санкции. – Монреаль, Канада: ИКАО 2011 - 68 с.

## Додаток А

Модифікований текст модулю nofly

```
using
System;
    using System.Collections.Generic;
    using System.Linq;
    using System.Text;
    namespace MissionPlanner.Maps
    {
        using System;
        using GMap.NET.Projections;
        using System.Globalization;
        using GMap.NET.MapProviders;
        using GMap.NET;
        using System.Reflection;
        /// <summary>
        /// MapboxNoFly
        /// </summary>
        public class MapboxNoFly : GMapProvider
        {
            public static readonly MapboxNoFly Instance;
            MapboxNoFly()
            {
            }
            static MapboxNoFly()
            {
                Instance = new MapboxNoFly();
                Type mytype = typeof (GMapProviders);
                FieldInfo field = mytype.GetField("DbHash",
BindingFlags.Static | BindingFlags.NonPublic);
                Dictionary<int, GMapProvider> list = (Dictionary<int,
GMapProvider>) field.GetValue(Instance);
                list.Add(Instance.DbId, Instance);
            }
            #region GMapProvider Members
            readonly Guid id = new Guid("22d6e496-842b-4549-bd6d-
8f013e6c019f");
            public override Guid Id
            {
                get { return id; }
            }
            readonly string name = "MapboxNoFly";
```

```

public override string Name
{
    get { return name; }
}
GMapProvider[] overlays;
public override GMapProvider[] Overlays
{
    get
    {
        if (overlays == null)
        {
            overlays = new GMapProvider[] { this };
        }
        return overlays;
    }
}
public override PureProjection Projection
{
    get { return MercatorProjection.Instance; }
}
public override PureImage GetTileImage(GPoint pos, int zoom)
{
    string url = MakeTileImageUrl(pos, zoom, LanguageStr);
    return GetTileImageUsingHttp(url);
}
#endregion
string MakeTileImageUrl(GPoint pos, int zoom, string language)
{
    string ret;
    ret = string.Format(CultureInfo.InvariantCulture, CustomURL,
pos.X, pos.Y, zoom, mapsource);
    return ret;
}
//http://a.tiles.mapbox.com/v3/examples.map-
i86l3621/{z}/{x}/{y}.png
public static string CustomURL =
"http://a.tiles.mapbox.com/v3/{3}/{2}/{0}/{1}.png";
public object mapsource = "mslee.h1kk2o6r";
}
}

```

```

using
System;

```



```

using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using DotSpatial.Projections.Transforms;
using GMap.NET;
using GMap.NET.WindowsForms;
using MissionPlanner.Properties;
using MissionPlanner.Utilities;
namespace MissionPlanner.Maps
{
    [Serializable]
    public class GMapMarkerOverlapCount : GMapMarker
    {
        List<GMapPolygon> footprintpolys = new List<GMapPolygon>();
        RectLatLng area;
        readonly Dictionary<PointLatLng, int> overlapCount = new
Dictionary<PointLatLng, int>();
        public GMapMarkerOverlapCount(PointLatLng p)
            : base(p)
        {
            area = new RectLatLng(p, SizeLatLng.Empty);
            if (colorbrushs == null)
            {
                colorbrushs = new SolidBrush[color.Length];
                int a = 0;
                foreach (var color1 in color)
                {
                    colorbrushs[a] = new SolidBrush(Color.FromArgb(140,
color1.R, color1.G, color1.B));
                    a++;
                }
            }
        }
        static Color[] color = new[]
        {
            Color.Purple,
            Color.Blue,
            Color.Aqua,
            Color.Green,
            Color.Yellow,
            Color.Orange,
            Color.Red,
            Color.DarkRed
        }
    }
}

```

```

};
static SolidColorBrush[] colorbrushs;
public override void OnRender(Graphics g)
{
    DateTime start = DateTime.Now;
    var pos = Overlay.Control.FromLatLngToLocal(Position);
    pos.Offset(-LocalPosition.X, -LocalPosition.Y);
    double width =

(Overlay.Control.MapProvider.Projection.GetDistance(Overlay.Control.F
romLocalToLatLng(0, 0),

Overlay.Control.FromLocalToLatLng(Overlay.Control.Width,
0))*1000.0);
    double m2pixelwidth = Overlay.Control.Width/width;
    Rect screenRect = new Rect(Overlay.Control.Width/-2,
Overlay.Control.Height/-2, Overlay.Control.Width,
Overlay.Control.Height);
    int skipped = 0;
    foreach (var pg in overlapCount)
    {
        GPoint p = Overlay.Control.FromLatLngToLocal(pg.Key);
        p.Offset(-pos.X, -pos.Y);
        if (p.X < screenRect.Left || p.X > screenRect.Right ||
            p.Y < screenRect.Top || p.Y > screenRect.Bottom)
        {
            skipped++;
            continue;
        }
        var col = Math.Min(pg.Value - 1, 7);
        var coloruse = colorbrushs[col];
        var widthc = 5*m2pixelwidth;
        var halfwidthc = widthc/2.0f;
        g.FillPie(coloruse, (float) (p.X - halfwidthc), (float) (p.Y -
halfwidthc), (float) (widthc),
            (float) (widthc), 0, 360);
    }
    drawLegend(g);
    Console.WriteLine("OnRender "+(DateTime.Now - start));
}
public void Add(GMapPolygon footprint)
{
    DateTime start = DateTime.Now;
    // if the same name footprint exists exit

```

```

if (footprintpolys.Any(p => p.Name == footprint.Name))
{
    return;
}
// if this is the first entry reset area
if (footprintpolys.Count == 0)
{
    Position = footprint.Points[0];
    area = new RectLatLng(footprint.Points[0], new SizeLatLng(0,
0));
}
// add the new footprint
footprintpolys.Add(footprint);
// recalc the area
foreach (var point in footprint.Points)
{
    if (!area.Contains(point))
    {
        double tllat = Math.Max(area.Lat, point.Lat);
        double tllng = Math.Min(area.Lng, point.Lng);
        double brlat = Math.Min(area.Bottom, point.Lat);
        double brlng = Math.Max(area.Right, point.Lng);
        // enlarge the area
        area = RectLatLng.FromLTRB(tllng, tllat, brlng, brlat);
    }
}
generateCoverageFP(footprint);
Console.WriteLine("Add "+(DateTime.Now - start));
}
public void drawLegend(Graphics g)
{
    // top left corner
    var tl = Overlay.Control.FromLocalToLatLng(0, 0);
    // top right corner
    var tr =
Overlay.Control.FromLocalToLatLng(Overlay.Control.Width, 0);
    // width in m
    double width =
Overlay.Control.MapProvider.Projection.GetDistance(tl, tr)*1000.0;
    // meters per pixel
    double m2pixelwidth = Overlay.Control.Width/width;
    var widthc = 20;
    var halfwidthc = widthc/2.0f;
    var pos = Overlay.Control.FromLatLngToLocal(Position);

```

```

pos.Offset(-LocalPosition.X, -LocalPosition.Y);
using (StringFormat stringFormat = new StringFormat())
{
    stringFormat.Alignment = StringAlignment.Center;
    stringFormat.LineAlignment = StringAlignment.Center;
    int a = 0;
    foreach (var brush in colorbrushs)
    {
        GPoint p = new GPoint(20, (long) (100 + a*(widthc + 5)));
        p.Offset(-pos.X, -pos.Y);
        g.FillPie(brush, (float) (p.X - halfwidthc), (float) (p.Y -
halfwidthc), (float) (widthc),
                (float) (widthc), 0, 360);
        g.DrawString((a + 1).ToString(), SystemFonts.DefaultFont,
Brushes.White,
                    new RectangleF((float) (p.X - halfwidthc), (float) (p.Y -
halfwidthc), (float) widthc,
                    (float) widthc), stringFormat);
        a++;
    }
}
}
public void generateCoverageAll()
{
    overlapCount.Clear();
    if (area.WidthLng > 1 || area.HeightLat > 1)
        return;
    for (double lat = Math.Round(area.Lat, 4); lat >= area.Bottom; lat
-= 0.0001)
    {
        for (double lng = Math.Round(area.Lng, 4); lng <= area.Right;
lng += 0.0001)
        {
            var p = new PointLatLng(Math.Round(lat, 4),
Math.Round(lng, 4));
            foreach (var footprintpoly in footprintpolys)
            {
                if (footprintpoly.IsInside(p))
                {
                    if (overlapCount.ContainsKey(p))
                    {
                        overlapCount[p]++;
                    }
                    else

```



## ВІДГУК

**на дипломний проект студента групи 122м-16-1 Безродного П.П.  
на тему: «Підвищення ефективності програмного забезпечення дронів з  
використанням програми Mission Planner при моніторингу в сільському  
господарстві»**

Тема дипломного проекту - конвертація растрової графіки у векторну з використанням генетичних алгоритмів.

Актуальність поставленого завдання обумовлюється широким попитом на різного роду інформаційних систем в різних сферах діяльності.

Тема дипломного проекту прямо пов'язана з об'єктом діяльності 122 спеціальності Комп'ютерні науки магістр з інформаційних технологій, інженер з комп'ютерних систем.

Зміст роботи включає створення, дослідження, спостереження за способи вирішення задач пошуку найкоротшого шляху для розрахунку траєкторії безпілотного літального апарату в вілно розповсюджене програмне забезпечення Mission Planner. При вирішенні поставленого завдання використовувалися наукові досягнення в областях розробки інформаційних систем і програмного забезпечення. Наукова новизна отриманих результатів полягає в проведенні аналізу та виявленні недоліків способи вирішення задач пошуку найкоротшого шляху для розрахунку траєкторії безпілотного літального апарату, а також у використанні методик аналізу та моделювання задач пошуку найкоротшого шляху для розрахунку траєкторії безпілотного літального апарату.

Практична цінність полягає у розробці модуля для програмного пакету Mission Planner, який дозволяє аналізувати заборонені для польотів зони, та обходити їх по найкоротшому шляху таким чином, щоб не залітати в заборонені зони. Це дає можливість автоматизувати обхід заборонених зон, підвищуючи таким чином захист геоданих від неавторизованих користувачів.

Ступінь опрацювання компонентів даного проекту, дозволяє оцінити роботу на «\_\_\_\_\_» і рекомендувати присвоїти студенту П.П. Безродному кваліфікацію «Інженер з комп'ютерних систем».

Керівник дипломного проекту,  
д.т.н., проф. кафедри ПЗКС

/Алексєєв М.О./

## РЕЦЕНЗІЯ

**на дипломний проект студента групи 122м-16-1 Безродного П.П.  
на тему: «Підвищення ефективності програмного забезпечення  
дронів з використанням програми Mission Planner при моніторингу в  
сільському господарстві»**

Як свідчать різні джерела інформації, в даний час в нашій країні та за кордоном за результатами оцінки безпілотних систем вони набувають щонадалі більшої ваги та значущості. В той же час безпека деяких геоданих досі перебуває у відносній небезпеці, тому важливою задачею сучасності є захист геоданих навіть через свободно розповсюджене програмне забезпечення.

В дипломному проекті в результаті проведеної роботи було проведено аналіз та виявлення недоліків способів вирішення задач пошуку найкоротшого шляху для розрахунку траєкторії безпілотного літального апарату, а також використання методик аналізу та моделювання задач пошуку найкоротшого шляху для розрахунку траєкторії безпілотного літального апарату.

Розроблена система дозволить підвищити швидкість обробки інформації, оптимізувати місце зберігання, скоротить терміни автоматичного розрахунку траєкторії з урахуванням заборонених зон і заощадить час роботи користувача користувачів.

Студент Безродний П.П. розібрався в специфіці застосування різноманітних інформаційних технологій задач пошуку найкоротшого шляху для розрахунку траєкторії безпілотного літального апарату у різних сферах діяльності.

З огляду на вищевикладене, можна зробити висновок, що даний проект цілком відповідає вимогам, що пред'являються до кваліфікаційних робіт рівня магістра.

В цілому проведений аналіз використаних компонентів даного проекту, дозволяє оцінити роботу на «\_\_\_\_\_» і рекомендувати присвоїти студенту П.П. Безродному кваліфікацію «Інженер з комп'ютерних систем».

Рецензент

## Додаток Г

### Перелік документів на диску

<b>ім'я файлу</b>	<b>опис</b>
<b>Пояснювальна записка</b>	
Пояснювальна записка.docx	Пояснювальна записка
<b>програма</b>	
projezt.zip	Архів. Містить вихідні коди програми.
<b>презентація</b>	
Презентація.pptx	Презентація дипломного проекту