

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 APM Planner при моніторингу в сільському господарстві.*

Виконавець:

студент _____ *2* **курсу, групи** _____ *122-16-1м*

(підпис) _____ *Кудальцев Є.Є.*
(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

**Міністерство освіти і науки України
Державний вищий навчальний заклад
«Національний гірничий університет»**

ЗАТВЕРДЖЕНО:

завідувач кафедри

програми забезпечення комп'ютерних систем
_____ (повна назва)

_____ І.М. Удовик
(підпис) (прізвище, ініціали)

« » _____ 20 18 року

ЗАВДАННЯ

на виконання кваліфікаційної роботи магістра

спеціальності _____ 122 «Комп'ютерні науки»
(код і назва спеціальності)

студенту _____ 122М-16-1 _____ Кудальцеву Є.Є.
(група) (прізвище та ініціали)

Тема дипломної роботи _____ Підвищення ефективності програмного забезпечення дронів з використанням програми ARМ Planner при моніторингу в сільському господарстві.

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора Державного ВНЗ «НГУ» від 26.12.2017 р. № 2127 -л

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБИТ

Об'єкт досліджень – робота з геоінформаційними потоками даних безпілотних літальних апаратів.

Предмет досліджень – захист геоінформаційних даних в безпольотних зонах.

Мета НДР – створити вдосконалене ПО об'льоту заборонених польотних зон за допомогою програми ARМ Planner.

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

3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна результатів, що очікуються, полягає в обґрунтуванні методики об'льоту критичних зон та знаходження найкоротшого шляху за допомогою програми ARМ Planner.

Практична цінність результатів полягає у розробленні програмного забезпечення, що реалізує алгоритм об'льоту 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 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)

Кудальцев Є.Є.

_____ (прізвище, ініціали)

Дата видачі завдання: 1.03.2017 р.

Термін подання дипломної роботи до ДЕК2018 р.

Реферат

Пояснювальна записка: __ стор., __ мал., __ таблиця., __ додатка, __ джерел.

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

Предмет досліджень – захист геоінформаційних даних в безпольотних зонах.

Мета НДР – створити вдосконалене ПО обльоту 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 даних, 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.

У 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 в обґрунтуванні методики обльоту критичних зон та знаходження найкоротшого шляху за допомогою програми APM Planner.

Практична цінність результатів полягає у розробленні програмного забезпечення, що реалізує алгоритм обльоту заборонених польотних зон.

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

ABSTRACT

Master diploma: __ pages., __ fig., __ table., __ addition, __ sources.

An object of researches is geographic information streams of these pilotless aircrafts.

The article of researches is a protection of geographic information 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 INFORMATION 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 усовершенствованное ПО облёта запрещенных 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" рассчитанные трудоемкость разработки программного обеспечения, затраты на создание ПО и длительность его разработки.

Научная новизна результатов, которая ожидается заключается в обосновании методики облёта критических зон и нахождение кратчайшего пути с помощью программы APM Planner.

Практическая ценность результатов заключается в разработке программного обеспечения, реализующего алгоритм облёта запрещенных полетных зон.

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

ЗМІСТ

Вступ.....	9
РОЗДІЛ 1.....	12
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	12
1.1. Особливості експлуатації безпілотних літальних апаратів.....	12
1.2. Регулювання використання малих БПЛА в різних країнах.....	19
1.3. Бортовий комплекс управління БПЛА.....	22
Розділ 2.....	26
Алгоритми пошуку найкоротшого шляху стосовно БПЛА.....	25
2.1. Проблема пошуку найкоротшого шляху.....	25
2.2. Застосування для пошуку шляху, стосовно БПЛА.....	40
2.3. Експериментальна перевірка.....	41
Розділ 3.....	50
Практична частина.....	50
3.1. Побудова безпілотного літального апарату на базі польотного контролера АРМ 2.6.....	50
3.2. Модифікація модуля маршрутизації на основі модуля пошуку найкоротшого шляху.....	57
Розділ 4.....	65
ЕКОНОМІЧНА ЧАСТИНА.....	65
4.1. Визначення трудомісткості розробки програмного забезпечення.....	65
4.2. Витрати на створення програмного забезпечення.....	68
4.3. Маркетингові дослідження ринку збуту розробленого програмного продукту.....	69
4.4. Оцінка економічної ефективності впровадження програмного забезпечення.....	72
Висновки.....	74
Список використаних джерел.....	75
Додаток А.....	83
Додаток Б.....	95

Додаток В.....	96
Додаток Г.....	97

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 та ін.

Одне з найважливіших напрямків в сучасній авіації пов'язано з розробкою безпілотних літальних апаратів (БПЛА), перші зразки яких з'явилися ще в середині минулого століття, як окремий вид перспективної зброї. В даний час БПЛА різних типів і призначень не тільки стоять на озброєнні багатьох армій світу, а й починають активно використовуватися в цивільній сфері. Широкий спектр практичних застосувань БПЛА охоплює вирішення таких основних завдань:

- Оптична, радіолокаційна, хімічна та радіаційна розвідка;
- Радіоелектронна боротьба;
- Моніторинг екологічної обстановки;
- Підтримку мережевих телекомунікацій;
- Контроль морського судноплавства

Логіка розвитку безпілотної авіації на рубежі ХХ-ХХІ ст. привела до виникнення класу малогабаритних літальних апаратів (МЛА).

Аналіз тенденцій, досвіду та проблем розробки МЛА представляє істотний інтерес і актуальність.

Поява класу малогабаритних БПЛА обумовлено цілим рядом різних чинників, головними з яких є виникнення принципово нових областей потенційного застосування БПЛА, наприклад, в складі комплектів індивідуального оснащення бійця як засіб оперативного збору інформації про

особливості та характер бойової обстановки або як засіб локального ураження противника; посилення вимог до функціональних і експлуатаційних можливостей БПЛА, включаючи підвищення скритності і оперативності збору та передачі розвідувальної інформації, а також значне зниження великогабаритних характеристик.

Актуальність роботи: 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 та 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 по оперативності зйомки. Для моніторингу протяжних об'єктів, якими є магістральні трубопроводи, лінії електропередачі, морські та сухопутні кордони, лінії залізних і автомобільних доріг.

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.

Цілі і завдання дослідження.

Метою даної магістерської роботи є створення і впровадження алгоритмів обльоту перешкод в сферу роботи програми для обрання польотного завдання, а також опис роботи методик відбору.

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

Предмет дослідження – захист геоінформаційних даних в безполітних 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 ставляться такі вимоги: по-перше, він не повинен допускати помилок на прецедентах, і по-друге – задовольняти деяким додатковим обмеженням, що відображає властивості і особливості предметної області. В рамках алгебраїчного підходу було доведено можливість побудови коректних алгоритмів для широкого класу регулярних завдань

Наукова новизна результатів, що очікуються, полягає в обґрунтуванні методики обльоту критичних зон й знаходження найкоротшого шляху за допомогою програми APM Planner.

Практичне значення отриманих результатів полягає у розробленні програмного забезпечення, що реалізує алгоритм обльоту заборонених польотних зон.

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

Особливий внесок магістра полягає в:

- виборі методів досліджень і технологій реалізації;
- створенні інформаційної системи, що реалізує механізми пошуку маршруту;

- розробці теоретичної частини роботи, в якій досліджені і систематизовані знання про існуючі підходи розробки інформаційних систем;
- оцінці отриманих результатів.

Структура і обсяг роботи. Робота складається з вступу, чотирьох розділів і висновку. Містить 95 сторінки друкованого тексту, з 16 малюнками, 92 використаних джерел.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Особливості експлуатації безпілотних літальних апаратів

Сучасне розширення наукових знань виявляє все нові шляхи отримання достовірних даних. Найчастіше для забезпечення всієї потреби в отриманні інформації в областях, пов'язаних з проведенням гірничих робіт, будівництвом, сільськогосподарськими роботами, екологічним контролем застосовуються методи дистанційного зондування.

В даний час існують різні засоби моніторингу. Серед них можна виділити 2 групи - контактні і дистанційні. До групи контактних методів можна віднести спостережні станції, інші спостереження, маркшейдерську зйомку. Дистанційні методи включають в себе аерофотозйомку з пілотованих апаратів, космічну зйомку, аерофото і відео зйомку з безпілотних літальних апаратів (БПЛА).

Традиційна аерофотозйомка дозволяє отримувати знімки великих площ з достатнім для задач моніторингу просторовим дозволом. Недоліком є висока ціна обслуговування літального апарату.

Космічна зйомка дозволяє отримувати знімки будь-яких ділянок земної поверхні, що може бути використано для моніторингу танення льодовиків або спостереження процесів виверження вулканів. Крім цього, при космічній зйомці є можливість отримувати мультиспектральні знімки, що дозволяє вирішувати пошукові завдання і спрощувати дешифровочні процеси. Основними недоліками космозйомки можна назвати високу залежність від

погодних умов, низьку просторову роздільність і відносно низьку періодичність зйомки.

В даний час інтенсивно розвиваються методи використання БПЛА. Ці технології мають ряд переваг: не вимагають спеціально підготовлених майданчиків і стартових комплексів, порівняно недорогі на відміну від великих авіакомплексів, безпечні. Найбільший розвиток, і як наслідок - поширення в світі отримали БПЛА крилатого або літакового типу. Сучасні БПЛА мають широкий спектр можливостей, починаючи від простого фото- і відеоспостереження у видимому спектрі, закінчуючи тепловізійною і лідарною зйомкою. Якщо розглядати можливість застосування БПЛА для цілей моніторингу, то розкриваються досить великі перспективи розвитку даних технологій.

Безпілотні літальні апарати як засіб моніторингу

БПЛА діляться за типом управління на ручне (з дистанційного пульта управління в межах оптичної спостережливості), автоматичне (з використанням бортових програмних пристроїв по заданій траєкторії на заданій висоті із заданою швидкістю і зі стабілізацією кутів орієнтації) і напівавтоматичне (автономний політ з можливістю оператора внесення змін в маршрут в інтерактивному режимі). Останні два способи сьогодні є найбільш затребуваними з боку користувачів безпілотних систем, тому що пред'являють найменші вимоги до підготовки персоналу і забезпечують безпечну та ефективну експлуатацію систем безпілотних літальних апаратів.

Повністю автоматичне управління допомагає полегшити вирішення задач аерофотозйомки заданої ділянки, коли потрібно знімати на великій відстані від місця базування поза контактом з наземною станцією. Для виконання аерофотозйомки, БПЛА повинен розглядатися в сукупності з його допомогою приладів оснащенням і корисним навантаженням, для чого введений термін безпілотна авіаційна система (БАС), яка крім БПЛА, складається з бортового комплексу управління, корисного навантаження і наземної станції управління.

Застосування БПЛА має безліч переваг: рентабельність, можливість зйомки з невеликих висот і поблизу об'єктів і отримання знімків високого дозволу, оперативність отримання знімків, можливість застосування в зонах надзвичайних ситуацій без ризику для життя і здоров'я пілотів.

Недоліками застосування безпілотних літальних апаратів є відсутність нормативно-правової бази для інтеграції БПЛА в єдиний повітряний простір, підвищена аварійність, нерегульовані питання сертифікації, страхування, реєстрації. Виходячи з вищенаведеного, можна сформулювати ряд ознак для визначення аерофотознімальних БПЛА, що застосовуються з метою моніторингу:

1. Тип конструкції: БПЛА літакового або вертолітного типу.

2. Спосіб управління: автоматичний або напівавтоматичний.

3. БПЛА для аерофотозйомки з метою картографування повинен мати на своєму борту повноцінний автопілот, здатний витримувати параметри зйомки (маршрут, кути нахилу фотоапарата, відсоток поздовжнього і поперечного перекриття, висоту) навіть при малій масі апарату в широкому діапазоні метеоумов.

4. Корисне навантаження: цифрова автоматична фотокамера (як альтернатива – відеокамера, тепловізор, ІК-камера).

Екологічний моніторинг за допомогою БПЛА

Негативні наслідки господарської діяльності і техногенного впливу людини на навколишнє середовище для біосфери сьогодні вже об'єктивна реальність. Однак негативні результати антропогенного впливу в сучасних умовах розвитку людської цивілізації не є неминучими. Багато в чому погіршення стану навколишнього середовища пов'язані з нераціональним використанням природних ресурсів, низьким рівнем розробки і подальшого впровадження сучасних безвідходних технологій, помилками в екологічній і технічній політиці, малої вивченістю можливих наслідків антропогенного впливу на екосистему. Комплекс спостережень за станом навколишнього середовища, оцінки і прогнозу змін стану навколишнього середовища під

впливом природних і антропогенних факторів називають екологічним моніторингом.

Зазвичай на територіях, де ведеться моніторинг, є ряд спостережних мереж, що належать різним службам, які спорудно-роз'єднані і не скоординовані в хронологічному і параметричному аспектах.

У зв'язку з цим, головними проблемами екологічного моніторингу є еколого-господарське районування і вибір інформативних показників екологічного стану територій з перевіркою їх системної достатності.

Застосування БПЛА може виступати як загальне джерело інформації для різних відомств на конкретну територію, а різноманітність, яку приносить корисне навантаження дозволить отримувати інформацію про різні аспекти одних і тих самих процесів.

БПЛА здатні вирішувати завдання:

- Спостереження за вогнищами лісових пожеж;
- Опустелювання та ерозія ґрунтів;
- Моніторинг динаміки повеней;
- Контроль атмосферних опадів.

Крім перерахованих завдань можливе спостереження за об'єктами з підвищеним радіаційним фоном. Наприклад, під час аварії на атомному реакторі Фукусіма - 1 в березні 2011 року для контролю над поширенням радіаційного фону застосовували БПЛА літакового типу.

Також під час цих подій БПЛА показали ефективність як засіб координації дій по боротьбі з аваріями подібного роду. Як відомо, екологічний моніторинг являє собою багаторівневу структуру, що складається з наступних рівнів: детальний, локальний, національний, глобальний. Застосування БПЛА доцільно для постійного спостереження за невеликими ділянками земної поверхні, тобто для детального рівня, а при сукупному використанні декількох комплексів і інтерполяції даних з цих комплексів, і для локального рівня. [8]

Застосування БПЛА для вирішення моніторингових завдань в різних галузях промисловості.

До теперішнього часу розроблено та доведено до дослідної експлуатації більше десятка різних комплексів БПЛА. Основні завдання комплексів - аерофотознімальні роботи з метою складання ортофотопланів, а також відеоспостереження в режимі реального часу.

Аерофотозйомка з використанням комплексу БПЛА може використовуватися при вирішенні наступного спектра моніторингових завдань:

- контроль стану лісових і сільськогосподарських угідь;
- оцінка наслідків повеней;
- спостереження за веденням дорожніх робіт і оцінка стану ліній електропередач, теплотрас, нафто і газопроводів;
- моніторинг гірничих робіт

Моніторинг відкритих гірських робіт

За рахунок деталізованості знімків і високої швидкості знімальних робіт відкриваються можливості для здійснення моніторингу технологічних процесів при проведенні відкритих гірських робіт, таких як: бурові, вибухові роботи, формування породних відвалів, пошук небезпечних ділянок, спостереження в важкодоступних місцях. Візуально не завжди можливо визначити наявність руху великих мас породи, а іноді і фізично неможливо через значний обсяг відвалів і великої віддаленості від наземних точок спостереження.

Комплекс БПЛА дозволяє вирішувати це завдання, виробляючи спостереження за важливими ділянками при виникненні такої потреби. З використанням комплексу можна вирішувати цілий ряд дослідницьких завдань, таких як вивчення динаміки провітрювання кар'єрів після вибухових робіт, дистанційне картування систем тріщин на уступах кар'єру з метою аналізу стійкості його бортів.

Виходячи зі знімальних даних, виробляються рекомендації з техніки безпеки на тих чи інших ділянках гірських робіт. Крім цього, можуть бути прийняті рішення про закриття тих чи інших кар'єрів, з огляду на їх невідповідність необхідним стандартам. Тривимірні моделі відвалів гірської породи можуть дати первинні уявлення про обсяги гірничих виробок, а постійні зйомки однієї і тієї ж ділянки дозволять більш раціонально розподілити використовувані ресурси. Крім зйомки в візуальному діапазоні можливе застосування мультиспектральних сенсорів для вирішення завдання як пошуку корисних копалин, так і для аналізу складу вже відкритих родовищ.

Моніторинг лісових і сільськогосподарських угідь

Завдання моніторингу в сільському і лісовому господарстві пов'язані з об'єктами великої площі, оцінка стану яких контактними методами є трудомісткою і дорогою. Застосування БПЛА дозволяє досліджувати ці об'єкти набагато швидше, що дозволяє збільшити продуктивність сільськогосподарських робіт.

Використання комплексів БПЛА дозволяє вирішувати широкий спектр завдань:

- Інвентаризація с / г угідь;
- Прогнозування сходів і врожайності;
- Побудова карт вегетаційних індексів;
- Хімічний аналіз ґрунтів на предмет наявності добрив і пестицидів.

Використання камер з можливістю зйомки в ближньому інфрачервоному діапазоні дозволяє отримувати карти вегетаційного індексу NDVI. Цей індекс дає можливість оцінки стану сільськогосподарських культур та прогнозування врожайності.

БПЛА дозволяє здійснювати моніторинг лісових пожеж, встановлюючи повітряне спостереження за осередком займання до моменту повної його ліквідації. Під час огляду діючих лісових пожеж визначаються основні напрямки поширення вогню, наявність загрози поширення пожежі до

населених пунктів, наявність окремих осередків горіння і по можливості виявляє місцезнаходження людей і техніки, зайнятих на гасінні пожежі з метою визначення правильності їх розстановки на кромці пожежі. Одночасно з отриманням відеоінформації представниками лісової служби приймаються рішення про тактичні способи гасіння. Проведення детального огляду чинної лісової пожежі проводиться з висоти 200-400 м. При огляді локалізованих пожеж основна увага приділяється димовим точкам по периферії пожежі, оцінюється ступінь їх небезпеки з точки зору можливості відновлення і поширення вогню. Для виявлення прихованих осередків горіння застосовуються тепловізори. Польоти для виявлення прихованих осередків горіння виробляються в ранні ранкові або пізні вечірні години, коли вплив сонячної радіації мінімально.

Моніторинг гідротехнічних споруд

Греблі ГЕС є найважливішим типом гідротехнічних споруд. Візуальний аналіз стану гребель не завжди дає можливість детально розглянути важливі частини греблі, як наприклад кріплення ліній електропередачі. Для поліпшення візуального контролю можна використовувати БПЛА або дистанційно-керований ЛА вертолітного типу з відеокамерою. Такий підхід дозволяє отримувати вичерпну інформацію про стан електровузлів, а також і про стан самої греблі. БПЛА літакового типу дозволяють проводити моніторинг водосховищ.

При розгляді знімків різного сезону можна робити висновки зонах постійного підтоплення від сезонних паводків. Крім цього, аерофотозйомка з БПЛА дозволяє визначати ширину зони абразії 50 ґрунту, напрямок міграції берегової лінії і зміну морфології схилів. Як результат постійного моніторингу на основі збору статистичних даних може бути побудована модель протяжного тимчасового процесу розмиття берегових ліній з метою прогнозування подальших змін.

Крім моніторингу вже існуючих гребель, БПЛА можна застосовувати як засіб спостереження за спорудами що будуються. Контроль за

дотриманням проекту будівництва ГЕС є важливим завданням, тому що наслідки порушень технології будівництва можуть бути дуже серйозними.

Моніторинг нафтогазових споруд та родовищ

Дана сфера промисловості вимагає можливості отримувати інформацію про лінійні протяжні об'єкти з високою роздільною здатністю. Комплекси БПЛА надають необхідний для цього інструментарій, а також служать для вирішення наступних завдань:

- Спостереження за станом трубопроводів;
- Пошук і розвідка родовищ;
- Екологічний моніторинг атмосферних викидів;
- Віддалений контроль досліджень і підрядних робіт.

Висока деталізація одержуваних зображень дозволяє виявляти незаконні підключення до трубопроводів для оперативного їх усунення. Для цілей моніторингу, крім фотозйомки у видимому діапазоні, також використовуються тепловізійні сенсори.

Комбіноване застосування фотокамери і тепловізора дозволяє вести практично цілодобове спостереження за станом трубопроводів і як наслідок своєчасно знаходити витік і порушення герметичності трубопроводів. Крім того, БПЛА дає можливість оцінювати екологічну обстановку, наприклад, визначати наявність нафтопродуктів в поверхневому шарі ґрунту або атмосфері. Використання мікрорадарів може дозволити оцінити кількість нафтопродукту, завдяки високій щодо проникаючої здатності.

Можливість інтеграції БПЛА з найпростішими газоаналізаторами дозволяє будувати картину екологічного забруднення атмосфери і вести підфакельний моніторинг. Аерофотозйомка з БПЛА, а також подальше створення тривимірних моделей місцевості, дозволяють виявляти певні закономірності в структурі рельєфу, що вказують на родовища вуглеводнів.

Моніторинг споруд енергетичної промисловості

Використання БПЛА дозволяє в короткі терміни знаходити пошкодження ліній електропередач, а також виробничі роботи.

1.2 Регулювання використання малих БПЛА в різних країнах

Федеральне Управління Громадянської Авіацією США (FAA) з серпня 2016- го року внесло зміни до законодавства, що стосуються використання БПЛА [12]. Згідно з деякими витягами з цього документа відомо, що комерційні БПЛА повинні важити до 55 кг, висота польоту до 122 м, швидкість не більше 100 км / ч. Комерційні БПЛА можуть активно експлуатуватися тільки в денний час. Пілоти «комерційних БПЛА повинні мати відповідні сертифікати, що дозволяють їм управляти даними апаратами і бути не молодше 16 років. Так введення цих правил було оприлюднено, ще один документ, який встановлює, що з грудня 2015 року всі власники БПЛА зобов'язані провести реєстрацію своїх апаратів. порушникам загрожує штраф у розмірі 27 000 доларів.

Для малих БПЛА (вага від 250 гр до 25 кг), які передбачають використання закликів в некомерційних цілях залишаються ті ж умови дотримання висоти польота і часу пілотування. Крім того, в кожному штаті вводяться свої норми і правила польотів, тому в кожному окремому випадку необхідно керуватися не тільки вимогами FAA, а й вимогами місцевої влади. До теперішнього моменту в США вироблено вже понад 300 000 реєстрацій БПЛА. У Великобританії всі питання з ліцензування та дозволу на пілотування БПЛА регулює Управління цивільної авіації (CAA).

Відповідно до британського законодавства управління дозволено тільки в зоні прямої видимості, яка становить простір в 500 м по горизонталі і в 122 м по вертикалі. Також БПЛА, які оснащені фото- і відеокамерою, забороняється наближати до людини, будівлі, автомобілю менш ніж на 50 м. Некомерційні апарати не можна пілотувати над великим скупченням людей. Також, британська влада намагається впровадити норму про програмування БПЛА, щоб автоматично обмежувати зону їх дії, наприклад, не дозволяти їм наближатися до аеропортів.

В даний час в ряді квадрокоптерів ці обмеження вже введені. Наприклад, у квадрокоптерів компанії DJI при плануванні місії автоматично показуються заборонені зони відповідно до карт Google.

У Таїланді до 2016 року були заборонені будь-які польоти всіх видів БПЛА з відеокамерою на борту. Згідно з новим законом з травня 2017р. можливо пілотувати БПЛА без ліцензії, якщо їх вага становить не більше 2 кг, а апарат використовується суто для особистих цілей. Крім цього, необхідно завжди тримати БПЛА в полі зору і здійснювати політ тільки в денний час доби. Висота польоту БПЛА не повинна перевищувати 90 м. Заборонено пілотувати БПЛА поруч з урядовими об'єктами, не можна зазіхати на приватне життя, а також забороняється наближатися менш ніж на 30 м до людей, будівлям і транспорту.

У В'єтнамі пілотування БПЛА повністю заборонено, як і ввезення даного обладнання на територію країни. По приїзду в країну пристрій буде вилучено до виїзду з країни. Всіма пунктами по здійсненню пілотування на тому чи іншому засобі займається Міністерство оборони.

У Росії польоти БПЛА регулюються законом, який вступить в силу з 5 липня 2017: «Про внесення змін до Повітряного кодексу Російської Федерації в частині використання безпілотних повітряних суден», згідно якого:

- власник радіокерованого апарату вагою від 250 грамів зобов'язаний зареєструвати літаюче пристрій (отримати ліцензію пілота).
- польоти БПЛА згідно нововведеного порядку необхідно узгодити з органами Росавіації (має бути дозвіл на здійснення польотів).
- вимога дотримання правил польотів. Крім цього, в документі сказано, що заборонено літати над людьми, щоб уникнути нещасних випадків при падінні БПЛА.

Здійснювати польоти над охоронюваними зонами (військові і адміністративні об'єкти, аеропорти) і на висоті вище 150 метрів.

В Україні польоти БПЛА регулюються наказом, який вступить в силу з 08 грудня 2016: «Про затвердження Правил виконання польотів

безпілотними авіаційними комплексами державної авіації України», згідно якого:

- власник радіокерованого апарату вагою від 250 грамів зобов'язаний зареєструвати літаючий пристрій.

- вимога дотримання правил польотів. Крім цього, в документі сказано, що заборонено літати над людьми, щоб уникнути нещасних випадків при падінні БПЛА.

Здійснювати польоти над охоронюваними зонами (військові і адміністративні об'єкти, аеропорти) і на висоті вище 150 метрів.

1.3 Бортовий комплекс управління БПЛА

До складу Бортового Комплексу Навігації та Управління БПЛА входять три складові елементи (Малюнок 1.1).

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.

На згадку автопілота може бути занесено до 255 поворотних пунктів маршруту. Кожна точка характеризується координатами, висотою проходження і швидкістю польоту.

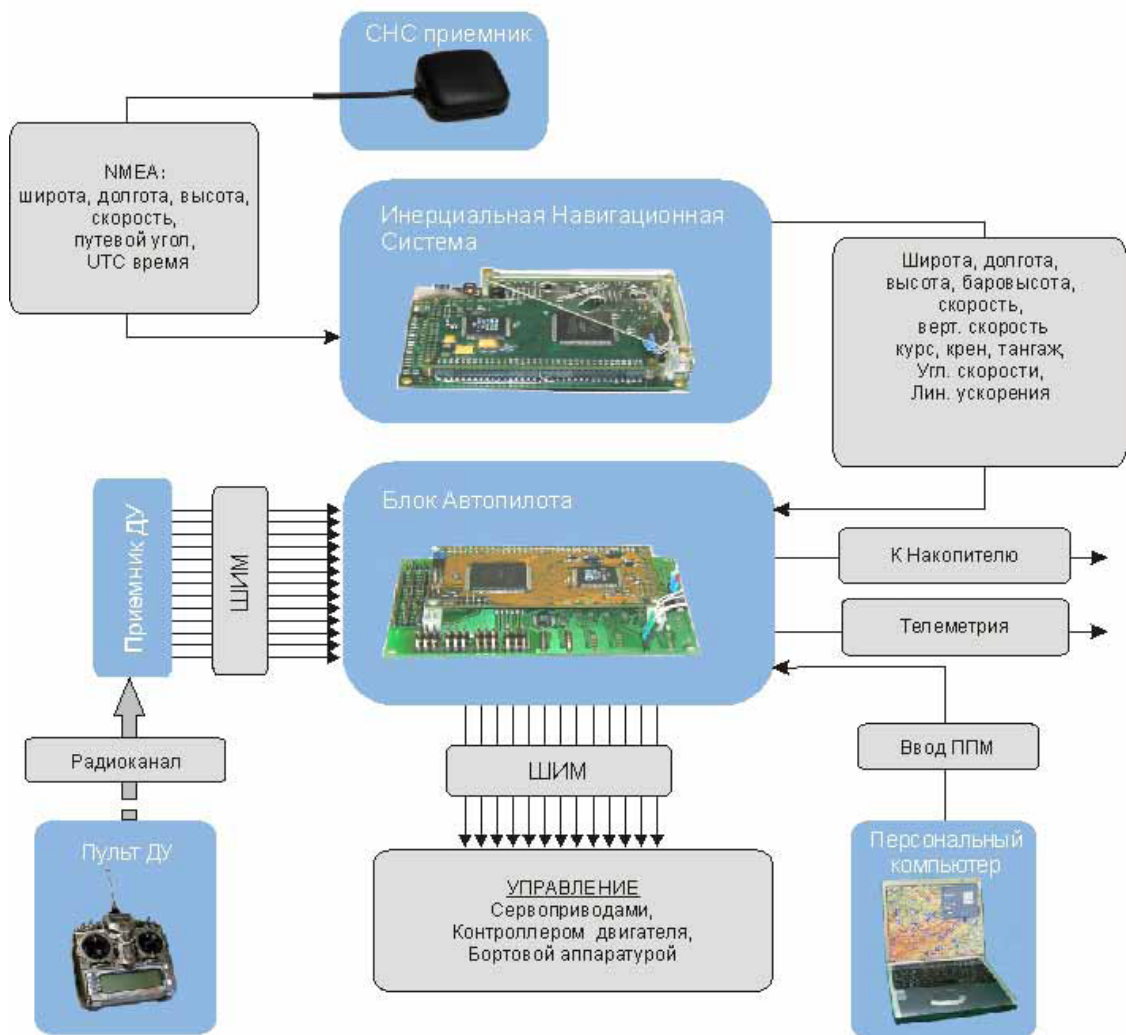


Рис. 1.1. – Бортовой комплекс навигации и управления БПЛА

У польоті автопілот також забезпечує видачу в канал передачі телеметричної інформації для стеження за польотом БПЛА (Рис. 1.2). 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. 1.2. – 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 у 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 детальної 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

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-ічніє купи: вони простіші, оцінка часу роботи містить меншу кількість прихованих констант.

Покращення алгоритму Дейкстри. Багаторівневі кошики (англ. Multi-level buckets, MLB) (табл. 1).

Таблиця 1.

Порівняння різних структур даних для пошуку найкоротшого шляху

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

Підходить тільки графів з цілочисельними ребрами.

- Необхідно скласти вершини в "кошика" $B[i] = \{u \in B[i] : d(u) = i\}$,
- Структура даних буде підтримувати індекс $L = \{\forall i : i < L \Rightarrow B[i] = \emptyset\}$
- L_{bottom} 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 $d(u)$.

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 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 ∞ , якщо в вершину не входить жодне ребро. Будемо говорити, що поточне значення $d(v)$ точно, якщо воно дорівнює довжині шляху $s \rightsquigarrow v$.

Припустімо, що довжина ребер невід'ємні. Нехай c - мінімальне з потокової значень. Тоді, если існує така вершина, що, то Поточне значення 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 прорелаксіруем всіх її сусідів. Якщо ж F порожньо, ми дістанемо з B вершину з мінімальною міткою і прорелаксіруем 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`, и 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 і з початкової і з кінцевої вершини. Така евристика не зіпсує швидкість роботи в гіршому випадку.

Створимо дві пріоритетних черзі і запусимо 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 $\ell_h(P) = \ell_h(s, v_1) + \ell_h(s, v_w)$.

$$\begin{aligned} \text{Тоді} \quad \ell_h(P) &= \ell_h(s, v_1) + \ell_h(v_1, v_2) + \dots + \ell_h(v_k, t) = \\ \ell(s, v_1) - h(s) + h(v_1) + \ell(v_1, v_2) - h(v_1) + h(v_2) + \dots + \ell(v_k, t) - h(v_k) + h(t) &= \\ \ell(P) - h(s) + h(t). \end{aligned}$$

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 роботи 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

Двоетапні алгоритми

На жаль, двонаправлений алгоритм Дейкстри всього в два рази швидше, ніж звичайно, а це занадто повільно. Розглянемо алгоритм пошуку найкоротшого шляху, що складається з двох етапів:

1. Препроцессинг:

- запускається один раз для графа,
- може займати багато часу,
- розраховує якусь допоміжну інформацію.

2. Запит:

- може використовувати дані, отримані під час препроцессинга,
- запускається на вимогу для пари (s, t) ,
- повинен виконуватися дуже швидко (в реальному часі).

Можливо розглянути в цьому ключі два приклади:

Алгоритм Дейкстри: препроцессинг - нічого не робити, запит - виконання алгоритму Дейкстри;

Повний перебір: препроцесінг - порахувати таблицю відстаней розміром $n \times n$ (займе близько 5 років часу і 1 петабайта пам'яті для карти Європи), запит - звернутися до елементу таблиці.

Обидва ці прикладу - крайні випадки. Нам потрібно щось більш гнучке: препроцесінг за годинник / хвилини, зростання кількості предпосчитаних даних лінійно від розміру графа і запити в реальному часі.

Алгоритм пошуку ALT

Абревіатура ALT розшифровується як A * + Landmarks + Triangle inequality: A * + орієнтири + нерівність трикутника.

1. Препроцесінг:

- взяти невелику кількість вершин (наприклад, 16), позначивши їх як орієнтири (англ. Landmarks),
- для кожного з орієнтирів порахувати найкоротші шляхи до всіх вершин,
- зберегти ці шляхи.

2. Запрос:

- використовуємо A *, якщо деяке ребро знаходиться на найкоротшому шляху між вихідною точкою і орієнтиром - по ньому йдемо у першу чергу (рис 2.1).

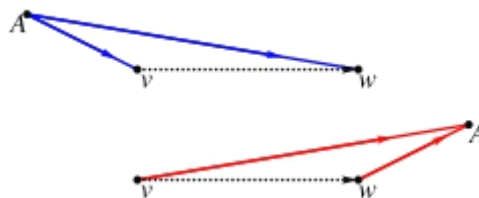


Рис. 2.1 – Алгоритм A *

Будемо використовувати нерівність трикутника для нижніх оцінок шляху (див. Рис. 1). нехай A — один з орієнтирів, тоді:

- $dist(v, w) \geq dist(A, w) - dist(A, v)$,

- $dist(v, w) \geq dist(v, A) - dist(w, A)$,
- $dist(v, w) \geq \max\{dist(A, w) - dist(A, v), dist(v, A) - dist(w, A)\}$.

Ця евристика добре працює, на дорожніх графах, для яких вірно наступне: як правило, найкоротший шлях зачіпає невелику кількість локальних доріг, потім велику автомагістраль і знову кілька локальних доріг.

Складнощі у виборі орієнтирів:

– хороший орієнтир для запиту $s \rightsquigarrow t$ повинен знаходитися "до" s (точно не буде загальних ребер на найкоротшому шляху) або "за" t (Чим гостріше кут, тим менше відхилення від попередньо порахували найкоротшого шляху до шуканого), нам потрібно вибрати такі орієнтири, які будуть непоганими для всіх запитів.

Виглядає логічним вибрати орієнтири на краю дорожньої мережі, тоді буде більше гострих кутів і орієнтири будуть краще.

Існують різні алгоритми вибору орієнтирів:

- Випадковий вибір (random)

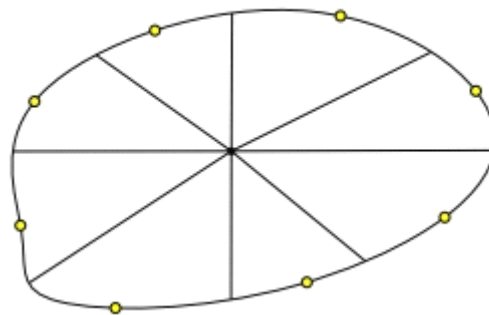


Рис. 2.2 – Орієнтири випадкового вибору

Як випливає з назви, орієнтири вибираються випадковим чином.

- Площинний (planar)

Розділимо карту на сектори однакової площі, візьмемо орієнтиром найбільш віддалену точку від центру в кожному секторі. Такій спосіб піходить, тільки тоді коли графік має віднос правильну форму. На практиці зазвичай використовується оптимізована версія цього алгоритму.

– Выборчий (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 поддереву T_r з коренем в v содержит орієнтир, размер v дорівнює 0.

Починаючи з максимальною за розміром вершини, підемо вниз по дереву T_r і знайдемо лист з максимальним розміром. Приймемо 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, $\ell(v, w) = dist(L, w) - dist(L, v)$, 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 относительно 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) .

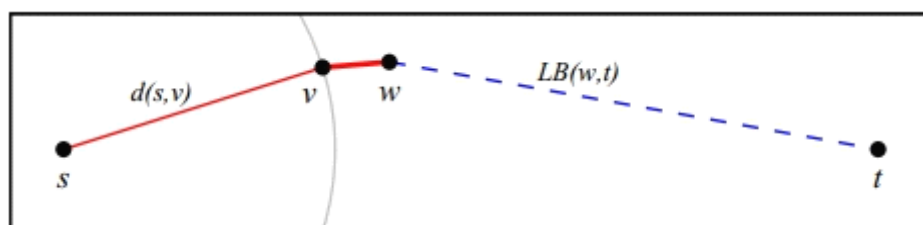


Рис. 2.3 – вибір вершини

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 маленьке охоплення.

Во время обработки 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.

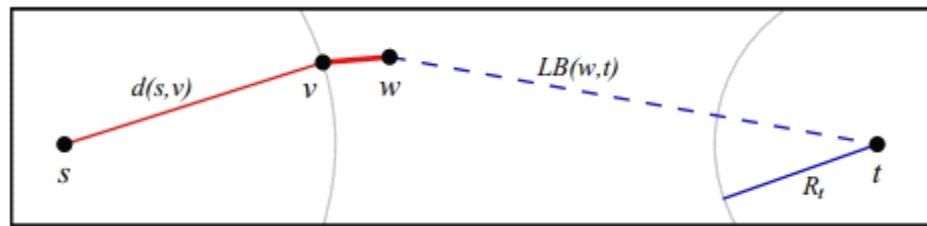


Рис. 2.4 – Вибір наступної вершини

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:

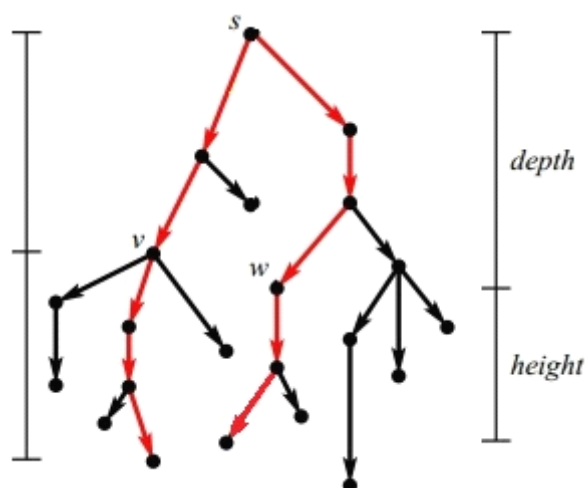


Рис. 2.5 – Препроцесінг

- 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 отладки (вершини с большим охопленням обробляються зазначеним вище алгоритмом - їх набагато менше, тому обробка буде швидка).

– Скорочення області пошуку

Зауважимо, що нам потрібні тільки вершини з маленьким охопленням $r(v) < \varepsilon, \varepsilon = const$. 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 $P_{s \rightsquigarrow t} = (s, s', \dots, v, \dots, t', t)$ ε - мінімальним, якщо виконуються умови: $dist(s, v) \geq \varepsilon$,

– $dist(s', v) < \varepsilon$,

– $dist(v, t) \geq \varepsilon$,

– $dist(v, t') < \varepsilon$.

Таким чином, алгоритм буде виглядати так:

– знайдемо оцінку охоплення $r'(v)$, використовуючи тільки ε — 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 2ε :

- 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' решту вершини (з охопленням $r(v) < \varepsilon$, вже оброблені),
- встановимо $\varepsilon \leftarrow 3\varepsilon$.

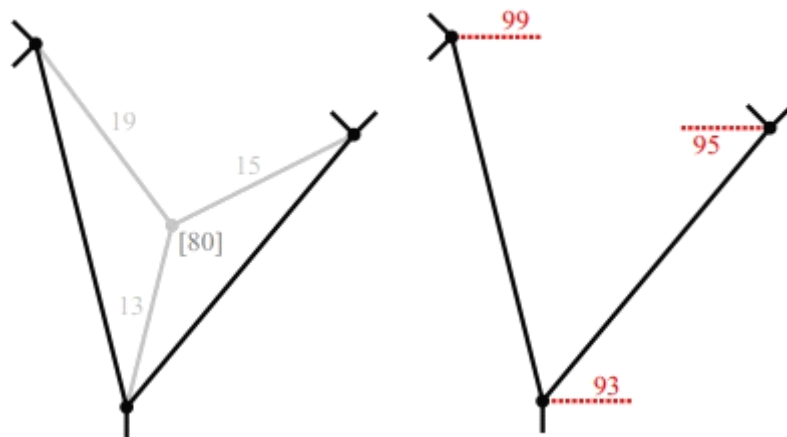


Рис. 2.6 - Частково оброблені дерева

- 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 "відкинутій" 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 в x величину $depth(v) = d(v) + in-penalty(x)$, где $d(v)$ — довжина шляху від кореня до вершини в дереві.

Щоб визначити висоту вершини, нам потрібно поняття "помилкових листя". Для кожної вершини v в дереві найкоротших шляхів додамо нащадка v' — помилковий лист, і ребро $(v, v') = out-penalty(v)$. В деякому розумінні, v' виступатиме в ролі всіх ребер, спочатку інцидентних вершині v , 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.

Уявімо 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.3

На цьому кроці ми будемо шукати пропускалися (англ. Bypassable) вершини. назвемо вершину v 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 $u \neq w$, то є uv 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 односторонні- і двосторонні- 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 подпуть $u \rightsquigarrow w$. Крім того, будь-який найкоротший шлях, що містить (u, v) , буде закінчуватися або в v , або в прилеглій вершині з низьким охопленням. Тоді, коректної верхньої оцінкою охоплення $(u, v) \in \bar{r}(u, v) = \ell(u, v) + out-penalty(v)$. аналогічно, $\bar{r}(v, w) = \ell(v, w) + in-penalty(v)$. 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 двосторонньою лінією, тому що крім оцінок, зазначених вище, можна додати: $\bar{r}(w, v) = \ell(w, v) + out-penalty(v)$, $\bar{r}(v, u) = \ell(v, u) + in-penalty(v)$.

Алгоритм Дейкстри 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 охват $\delta = const$ 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 запустимо пошук точних значень охоплення для кожної вершини цього підграфа. Так як дерева найкоротших шляхів будуть містити тільки вершини з G_δ , 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 підраховували точне значення охоплення.

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 в ряді випадків доводиться змінювати цьому правилу, наприклад в разі забороненої для польотів зони (наприклад, військового об'єкта, або ж 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 обльоту забороненого об'єкта, заданого вершинами багатокутника на координатній сітці (рис. 2.7).

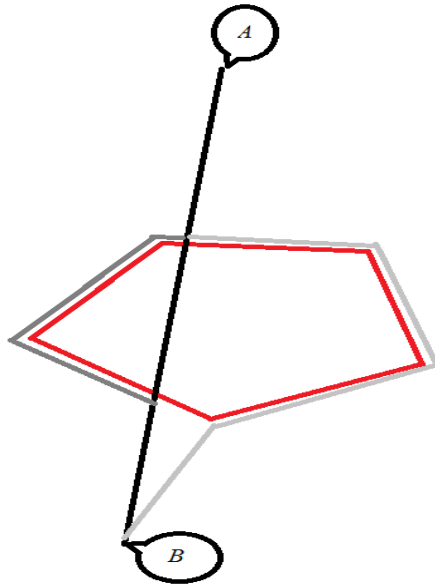


Рис. 2.7 – Обхід забороненої зони

2.3. Експериментальна перевірка

Рішення завдання маршрутизації є важливим етапом підготовки польоту інформаційних безпілотних літальних апаратів, який покликаний підвищити ефективність їх цільового функціонування [1]. Одна з поширених трактувань завдання маршрутизації зводиться до формування послідовності обльоту точок, певним чином розташованих на земній поверхні [2-5]. При цьому мінімізується тривалість польоту за маршрутом, оскільки це забезпечує підвищення оперативності збору інформації. Рішення має вигляд замкнутого маршруту польоту, що зв'язує всі задані точки, причому через кожен точку БПЛА пролітає тільки один раз, тобто маршрут не має петель. Таким чином, мова йде про класичну замкнutoї задачі алгоритму A^* . Специфіка згаданих робіт проявляється в тому, що саме розуміють автори під елементами вихідної матриці алгоритму A^* і, відповідно, який фізичний

зміст критерію. В [2,3] колійна швидкість БПЛА вважається відомої і постійною протягом усього польоту. Таким чином, фактично мова йде про мінімізацію протяжності маршруту. При цьому матриця алгоритму A^* є симетричною, а завдання алгоритму A^* виявляється планарним. В [4,5] колійна швидкість визначається з урахуванням дії в зоні польоту постійного вітру, що має відому швидкість і напрямок. Відповідно мінімізується тривалість польоту з урахуванням дії вітру, а матриця алгоритму A^* виявляється несиметричною. Саме поняття точок, що пов'язуються маршрутом, може мати різний фізичний зміст. Наприклад, в [2] це геометричний центр певного району земної поверхні, в [4] це точка на земній поверхні, з якої камера, розміщена на БПЛА, може знімати представляє інтерес об'єкт під певним ракурсом. Таким чином, в рамках класичної постановки задачі алгоритму A^* , при зовнішній простоті її формулювання, вдається врахувати нюанси цільового функціонування різних БПЛА, які вирішують різноманітні завдання інформаційного забезпечення, включаючи фактори, пов'язані з впливом зовнішнього середовища. Слід зазначити, що крім розглянутої в перерахованих вище статтях замкнутої завдання алгоритму A^* , становлять практичний інтерес постановки задачі маршрутизації, які можна звести до варіантів постановки розімкнутої завдання алгоритму A^* .

Варіанти постановки задачі маршрутизації

Стосовно легких БПЛА завдання маршрутизації має ряд особливостей. З одного боку, повітряна швидкість таких апаратів відносно невелика, тому необхідно враховувати дію вітру в зоні польоту. З іншого боку, для легких БПЛА характерні порівняно невеликий радіус дії і нетривалий за часом політ. Це дозволяє вважати напрямок і швидкість вітру постійними для всієї зони і часу польоту. Будемо також вважати, що повітряна швидкість і висота польоту БПЛА при русі по маршруту постійні.

Передбачається, що курсовий автопілот забезпечує політ БПЛА з таким напрямком вектора повітряної швидкості, яке з урахуванням дії відомого

постійного вітру забезпечить політ апарату між сусідніми точками маршруту по прямій їх з'єднає. Розрахунок елементів матриці алгоритму A^* , тобто розрахунок часу польоту БПЛА в поле відомого постійного вітру між усіма парами точок виконується за формулами, отриманими в [5]. Значення критерію, тобто розрахункова тривалість польоту за маршрутом, визначається як сума тривалості польоту по прямій між усіма точками маршруту. При цьому передбачається, що в силу малого радіусу розвороту легкого БПЛА величина фактичної тривалості польоту за маршрутом з урахуванням розворотів по курсу несуттєво відрізняється від розрахункової тривалості польоту за маршрутом. Крім того, вважається, що зі збільшенням тривалості польоту за рахунок перехідного процесу при зміні курсу можна знехтувати в силу його незначності. Постановка завдання маршрутизації як замкнутої завдання алгоритму A^* полягає в наступному. При відомій повітряній швидкості БПЛА, а також швидкості і напрямку вітру в зоні польоту потрібно знайти найшвидшого, замкнутий, зв'язний і не має петель маршрут обльоту всіх n заданих своїми координатами на земній поверхні точок. Доповнимо цю постановку замкнутої завдання маршрутизації варіантами постановки задач, які в подальшому інтерпретуємо як варіанти розімкнутої завдання алгоритму A^* . При відомій повітряній швидкості БПЛА, а також швидкості і напрямку вітру в зоні польоту потрібно знайти найшвидшого, розімкнутий, зв'язний і не має петель маршрут обльоту всіх n заданих своїми координатами на земній поверхні точок, якщо:

- точки початку і кінця маршруту задані;
- точка початку маршруту задана, а точка кінця вибирається;
- точка кінця маршруту задана, а точка початку вибирається;
- точки початку і кінця маршруту не задані і вибираються.

Математична постановка задачі маршрутизації польоту як замкнутої

завдання алгоритму A* зводиться до знаходження $n(n-1)$ булевих змінних

$x_{ij} \in \{0, 1\}$, $(i, j = \overline{1, n}; i \neq j)$, які мінімізують час обльоту всіх заданих точок, тобто забезпечують

$$\sum_{i=1}^n \sum_{j=1}^n t_{ij} x_{ij} \rightarrow \min, \quad i \neq j.$$

Тут t_{ij} - це час перельоту з точки з номером i в точку з номером j .

Якщо $x_{ij} = 1$, то рухаючись по маршруту, БПЛА з точки i перелітає

безпосередньо в точку j . Якщо $x_{ij} = 0$, то при русі по маршруту,

безпосередній переліт з точки з номером i в точку з номером j не передбачений. При цьому повинні бути виконані обмеження:

$$\sum_{i=1}^n x_{ij} = 1 \quad (j = \overline{1, n}; i \neq j),$$

$$\sum_{j=1}^n x_{ij} = 1 \quad (i = \overline{1, n}; i \neq j).$$

Обмеження відображають ту умову, що БПЛА повинен побувати в кожній з заданих точок рівно один раз. Як відомо [6], постановка (1) - (3) є завданням про призначення, рішення якої містить n змінних рівних одиниці,

а решта $n(n-2)$ є нульовими. Рішення може складатися з декількох простих

вершино-непересічних циклів, так званих підциклів, що проходять через меншу ніж n число точок. Тому, для отримання маршруту, що зв'язує n точок, необхідно доповнити обмеження (1) - (3) умовами, що забезпечують зв'язність шуканого циклу. Такі умови можна записати в вигляді різних обмежень. У ряді джерел, наприклад, в [6], використовуються обмеження виду:

$$u_i - u_j + nx_{ij} \leq n - 1, \quad u_i \geq 0 \quad (i, j = \overline{1, n}; i \neq j).$$

Розглянемо тепер розімкнуте завдання маршрутизації. Нехай стартова точка має номер s , а фінішна точка має номер f . Математична постановка

розімкнутої завдання маршрутизації польоту зводиться до знаходження (-1)

$(n-2) + 1$ булевих змінних $\varepsilon \{0, 1\}$, $(i, j = \overline{1, n}; i \neq j, i \neq f, j \neq s)$, які

мінімізують час обльоту всіх заданих точок, тобто забезпечують

$$\sum_{i=1}^n \sum_{j=1}^n t_{ij} x_{ij} \rightarrow \min, \quad i \neq j, \quad i \neq f, \quad j \neq s,$$

при виконанні обмежень

$$\sum_{i=1}^n x_{ij} = 1 \quad (j = \overline{1, n}; i \neq j, i \neq f, j \neq s),$$

$$\sum_{j=1}^n x_{ij} = 1 \quad (i = \overline{1, n}; i \neq j, i \neq f, j \neq s).$$

Природно, обмеження (6) - (7) доповнюються умовою виключення

підциклів. Рішення даного завдання містить $(n-1)$ змінних рівних одиниці, а

решта $(n-1)(n-3) + 1$ змінні є нульовими. Дана постановка справедлива і

для варіантів, коли не задана або точка старту, або точка фінішу, або і те й інша одночасно.

У роботах [2-5] не обговорювалися детально питання програмно алгоритмічного забезпечення вирішення задачі маршрутизації і виникають

при цьому обчислювальні труднощі, хоча відомо [6], що рішення задачі алгоритму A^*B силу її NP-повноти представляє самостійну серйозну проблему. Обчислювальні експерименти показали, що усунення підциклів введенням обмежень типу (4) при кількості точок більше 10 призводить до стрімкого зростання часу рішення. У даній роботі в якості альтернативи був використаний метод з послідовним винятком підциклів [7]. Суть методу полягає в тому, що завдання вирішується без врахування обмежень (4), а якщо отримується рішення містить підцикли, то вводяться додаткові обмеження, які забороняють ці підцикли, після чого рішення повторюється. Процес триває до того моменту, поки отримане рішення не буде містити підциклів. Збіжність такої процедури доведена [8]. Працездатність запропонованого підходу продемонстрована на прикладі розрахунку замкнутого маршруту обльоту 40 точок, розташованих випадковим чином на майданчику розміром 100×100 кілометрів. Координати цих точок наведені в таблиці 1. Повітряна швидкість БПЛА при розрахунках приймалася рівною 19,44 м / с, а швидкість південно-західного вітру становила 11,1 м / с. Ці значення повітряної швидкості і значення параметрів постійного вітру були використані і в інших прикладах. Координати точок маршруту обльоту наведені в таблиці 1.

Таблиця 1.

Координати точок маршруту обльоту ($n = 40$)

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
x	35	59	92	76	38	8	53	93	57	1	16	31	17	26	69	45	23	15	54	8
y	83	55	29	75	57	5	78	13	47	34	79	53	60	65	75	8	91	83	100	44
№	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
x	11	0	82	8	26	43	18	15	87	55	85	35	40	24	18	42	90	49	34	37
y	96	77	87	40	80	91	26	14	58	14	62	51	8	12	24	5	94	49	90	11

Найдений оптимальний маршрут об'їзду має вигляд: 1 → 7 → 2 → 9 → 38 → 5 → 32 → 12 → 14 → 13 → 33 → 40 → 34 → 6 → 28 → → 35 → 27 → 10 → 24 → 20 → 25 → 18 → 11 → 22 → 21 → 17 → 39 → 26 → 19 → → 37 → 36 → 16 → 30 → 8 → 3 → 29 → 31 → 15 → 4 → 23 → 1. Розташування 40 точок зв'язує їх оптимальної вторинної маршрут об'їзду показані на малюнку 2.8

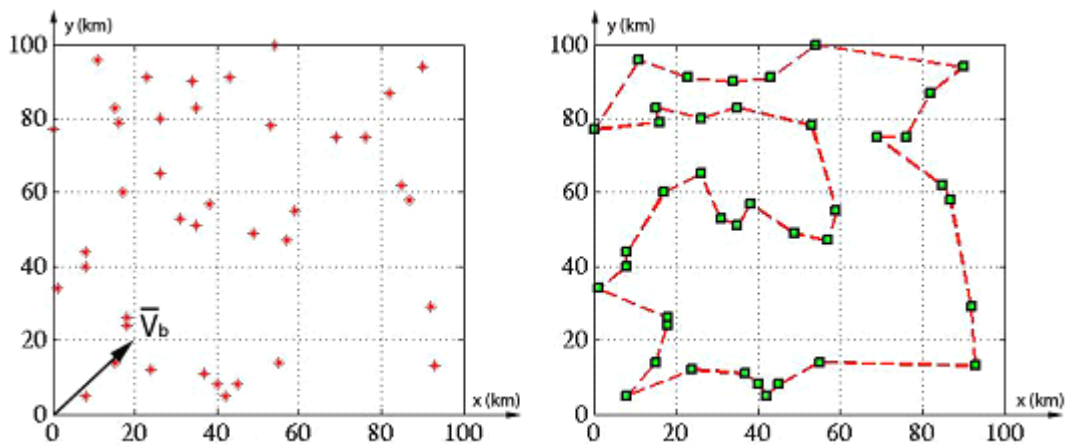


Рис 2.8 – Графік розподілу 40 точок

Для вирішення цієї та інших завдань маршрутизації, обговорюваних в статті, був використаний комп'ютер з процесором Intel (R) i3-4160 CPU @ 3.60GHz і оперативною пам'яттю 4,00 ГБ. Час розрахунку маршруту для 40 точок склало 61,12 секунд. Розглянемо знаходження розімкнутого маршруту об'їзду 15 точок, розташованих випадковим чином. Нехай точка початку

маршруту має номер $= 2$, а точка закінчення має номер $= 6$. Координати

точок наведені в таблиці 2.

Таблиця 2.

Координати точок маршруту при $n =$

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x	54	54	52	22	11	40	37	63	93	19	70	53	86	39	74
y	65	72	99	11	6	45	76	77	97	14	9	53	48	67	52

15

В результаті розрахунку було знайдено наступного оптимальний розімкнутий маршрут: $2 \rightarrow 8 \rightarrow 9 \rightarrow 3 \rightarrow 7 \rightarrow 14 \rightarrow 1 \rightarrow 12 \rightarrow 15 \rightarrow 13 \rightarrow 11 \rightarrow 4 \rightarrow 5 \rightarrow 10 \rightarrow 6$. Розташування 15 точок і зв'язує їх оптимальний розімкнутий маршрут показані на малюнку 2.9.

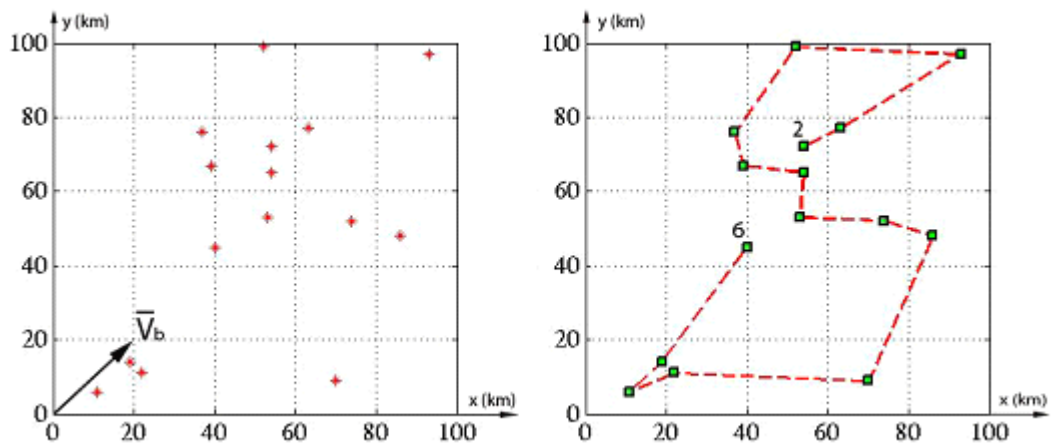


Рис 2.9 – Графік розподілу 15 точок

Час розрахунку цього маршруту склав 3.06 секунд. Нехай точка початку маршруту задана і має номер $= 2$, а точка закінчення не задана і

вибирається. У цьому випадку завдання маршрутизації вирішується ($- 1$)

раз для всіх можливих варіантів точки закінчення маршруту. Координати точок, що пов'язуються маршрутом, наведені в таблиці 3.

Таблиця 3.

Координати точок маршруту (n = 15)

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x	79	33	44	77	86	51	59	20	75	79	53	11	68	19	15
y	95	67	83	17	99	88	15	41	83	32	9	14	50	50	5

Результат розрахунку оптимального розімкнутого маршруту мають вигляд: 2 → 14 → 8 → 12 → 15 → 11 → 7 → 4 → 10 → 13 → 3 → 6 → 9 → 1 → 5. Розташування 15 точок і зв'язує їх оптимальний розімкнутий маршрут показані на рис. 2.10.

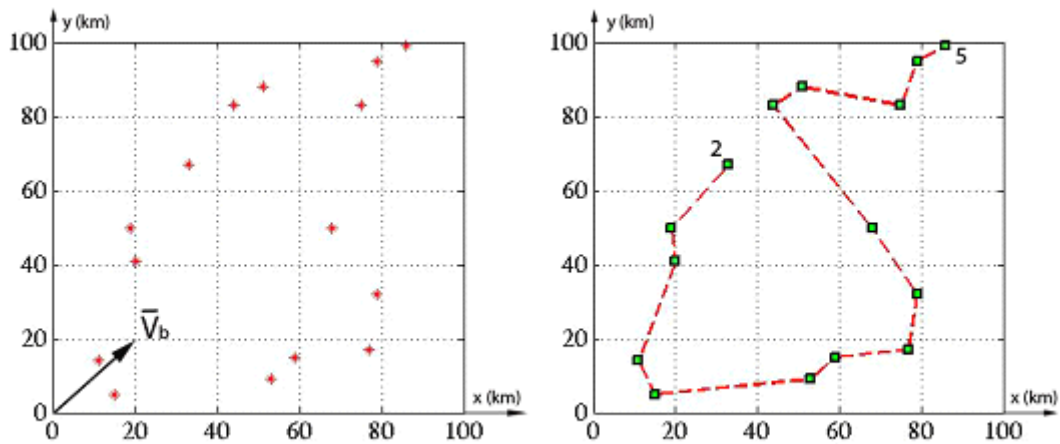


Рис. 2.10 – Графік розподілу 15 точок з точкою початку

Таким чином, сформульований ряд представляє практичний інтерес завдань знаходження маршруту обльоту легким безпілотним літальним апаратом заданих на земній поверхні точок, що знаходяться в полі постійного вітру.

Розділ 3
Практична частина

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

ArduPilot Mega (APM) - Польотний контролер, сумісний з Arduino (ArduPilot Mega).

Польотний контролер ArduPilot Mega є повноцінним рішенням БПЛА (безпілотних літаючих апаратів UAV), які дозволяють окрім радіокерованого дистанційного пілотування - автоматичне керування заздалегідь створеного маршруту, тобто політ по точкам. Має можливість двосторонньої передачі телеметричних даних з борту на наземну станцію (телефон, планшет, ноутбук, DIY) і ведення журналу у вбудовану пам'ять.

Він заснований на автопілоті APM 2.x, які розробляються спільнотою DIY Drones і базується на open-source проектах, що дозволяє перетворити будь-який апарат в автономний засіб і ефективно використовувати його не тільки в розважальних цілях, але і для виконання професійних проектів.

APM ArduCopter є простим в налаштуванні та із простою платформою для мультимоторних систем і вертольотів. Сьогодні на ринку його особливості виходять далеко за рамки основного ручного управління мультимоторних систем. На відміну від ручного управління мультимоторних систем, ArduCopter є комплексним рішенням БПЛА, пропонуючи як класичне радіоуправління, так і автономний політ, в тому числі по точкам, планування місій і телеметрії, які відображаються на наземній станції.

Система Arducopter знаходиться на передових позиціях повітряної робототехніки і призначена для тих людей, які хочуть спробувати сучасні технології і нові стилі польоту.

Планування всіх місій доступно через опційний двосторонній зв'язок. Точки, зміна режимів польоту, навіть тюнінг може бути зроблений з вашого ноутбука, в той час як апарат перебувати в повітрі!

Автоматичний взліт і посадка. Просто введіть апарат в цей режим і дивіться як ArduCopter сам виконає свою місію повністю автономно.

Версії плат:

- АРМ 1.0 - вже застаріла
- АРМ 2.0 - вже застаріла
- АРМ 2.5-2.8 - найбільш активно підтримується в даний момент.
- Mini АРМ Pro - «Китайський» клон в міні-корпусі.
- РХ4 - нова версія з 32-бітовим процесором. Поки в стадії бети, але дуже перспективний варіант.
- Pixhawk - плата другого покоління заснована на автопілоті РХ4 від інституту комп'ютерного зору Швейцарії за підтримки 3DRobotics працює на кодї ArduPilot і підтримує нові цифрові датчики.

Особливості

- Є всі можливі режими польоту (carefree, утримання висоти тощо)
- Розвинені функції GPS (утримання позиції, повернення додому)
- Можливість автономного польоту по точкам, обльоту території та ін
- Підтримка OSD, модемів, Bluetooth, датчиків струму і напруги
- Автономне ведення польотних логів
- Відмінна і величезна документація

На рис. 3.1 - 3.2 наведені схеми підключення.

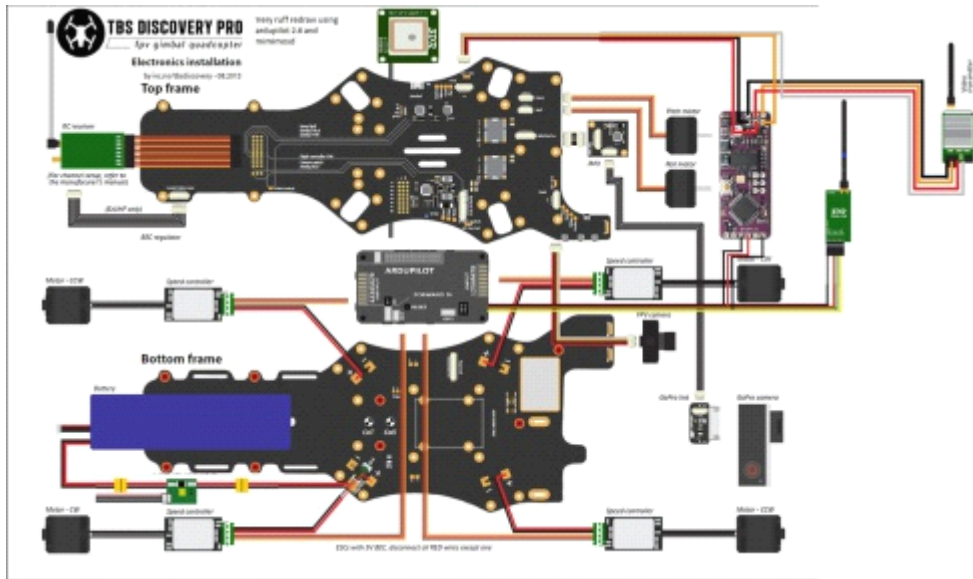


Рис. 3.1 - Схема підключення контролера АРМ 2.6

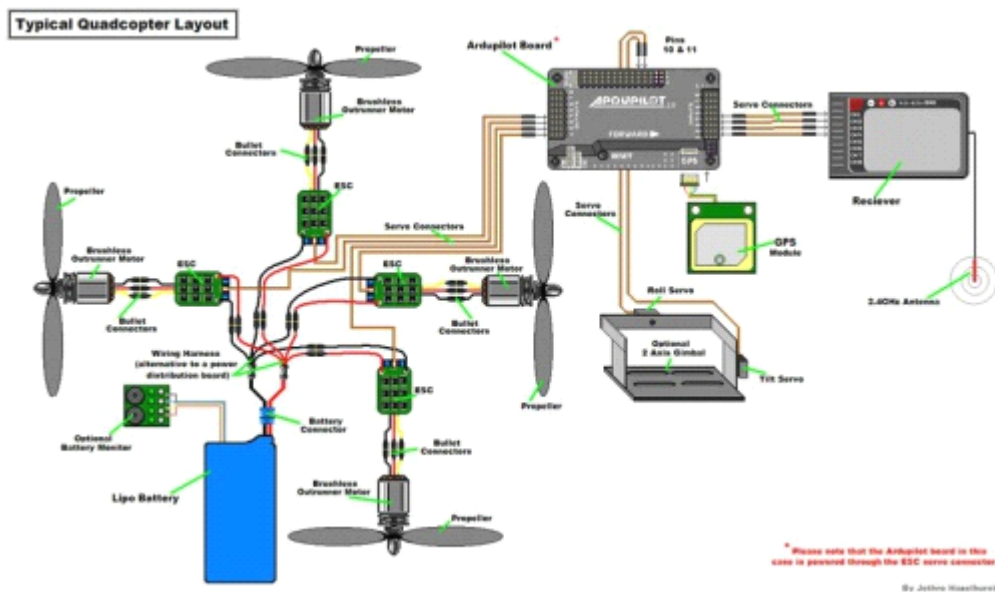


Рис. 3.2 - Схема підключення контролера АРМ 2.6

Починаючи з версії 2.0 з плати був видалений вбудований стабілізатор напруги 5В. Для забезпечення надійності роботи АРМ необхідне окреме джерело для живлення авіоніки (приймач, автопілот, телеметрія, OSD) і окремий для сервоприводів.

Ідеальним рішенням для живлення авіоніки є зв'язка UBEC з виходом 6В з додатковим LDO-регулятором, що знижує в свою чергу напругу до 5В. Низький рівень обрізки по напрузі не створює надмірного нагріву лінійного регулятора при струмі порядку 500мА, яких зазвичай з великим запасом досить на повний комплект авіоніки. Це рішення відмінно тримає навантаження, точно показує заряд акумулятора, захисний діод по входу запобігає спалювання UBEC при помилковому підключенні.

Живлення сервоприводів, підключених до вихідної планки АРМ 2-2.5 можна забезпечити двома способами:

- якщо ви використовуєте дорогий регулятор оборотів, має бути вбудоване імпульсне джерело живлення, і цього може виявитися цілком достатньо, щоб забезпечити живлення сервоприводів. При цьому і ESC і сервоприводи просто підключаються до планки виходів.

- якщо у вас використовується ESC з лінійним регулятором напруги (BEC) то при інтенсивній роботі сервоприводів або підключування однієї з них може статися перегрів ESC в результаті якого пропаде живлення на всіх сервоприводах і апарат стане некерованим. Тому, якщо при інтенсивній роботі сервоприводів ESC істотно нагрівається, слід його червоний провід в шлейфі управління відрізати (або вийняти з трьохконтактного роз'єму і заізолювати) а живлення подати від додаткового UBEC в будь-який вільний штекер планки виходів. У цьому випадку підійде будь-який потужний UBEC. У разі використання високовольтних сервоприводів (з живленням 7,2 В) можливо подавати на планку виходів цю напругу в якості живлення серв.

Головний екран (рис. 3.3). Він же екран польотних даних, також містить вікно HUD (Heads-up Display).

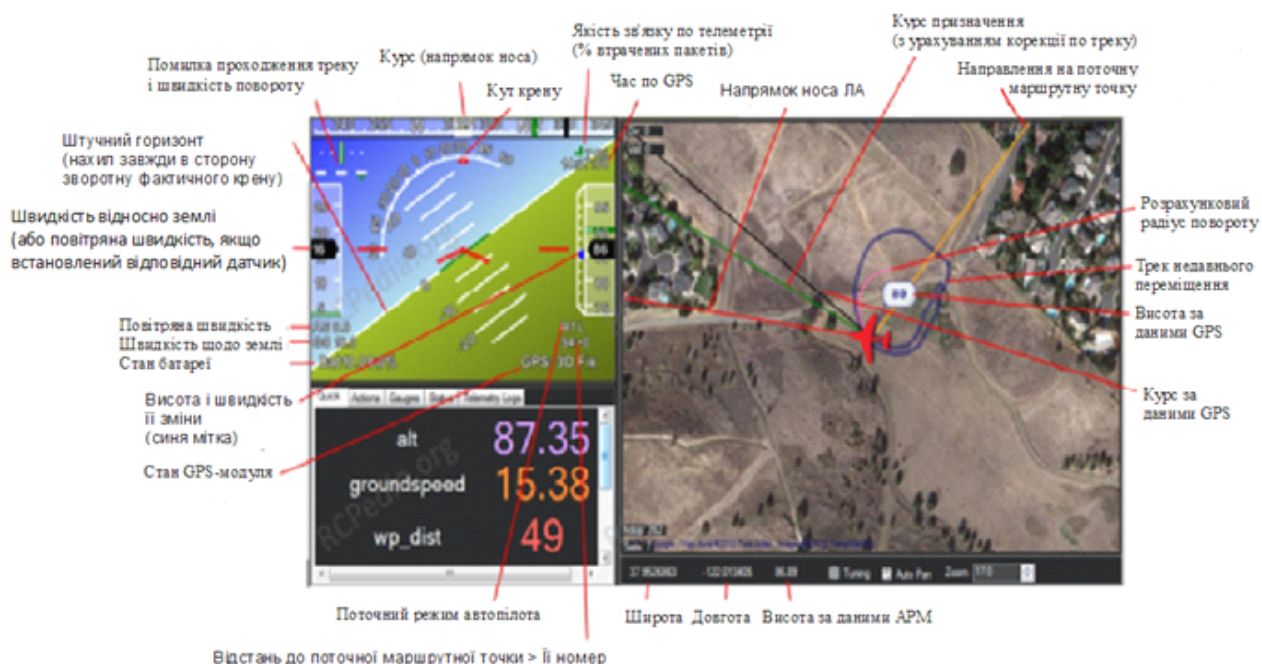


Рис. 3.3 - Головний екран

На ньому відображаються такі параметри польоту:

- **Червона лінія** - фактичний курс, напрям носа (переду) ЛА.
- **Зелена лінія** - курс призначення з урахуванням корекції по треку.
- **Чорна лінія** - курс, визначений за даними, отриманими від GPS (зазвичай відрізняється від фактичного).
- **Жовта лінія** - напрям на поточну маршрутну точку, до якої виповнюється політ.
- **Desired turn radius** - розрахунковий радіус повороту
- **GPS reported altitude** - висота за даними GPS
- **Latitude** - широта
- **Longitude** - довгота
- **Track of recent travel** - Трек недавнього переміщення

Вид вікна HUD більш крупно зображено на рис 3.4.

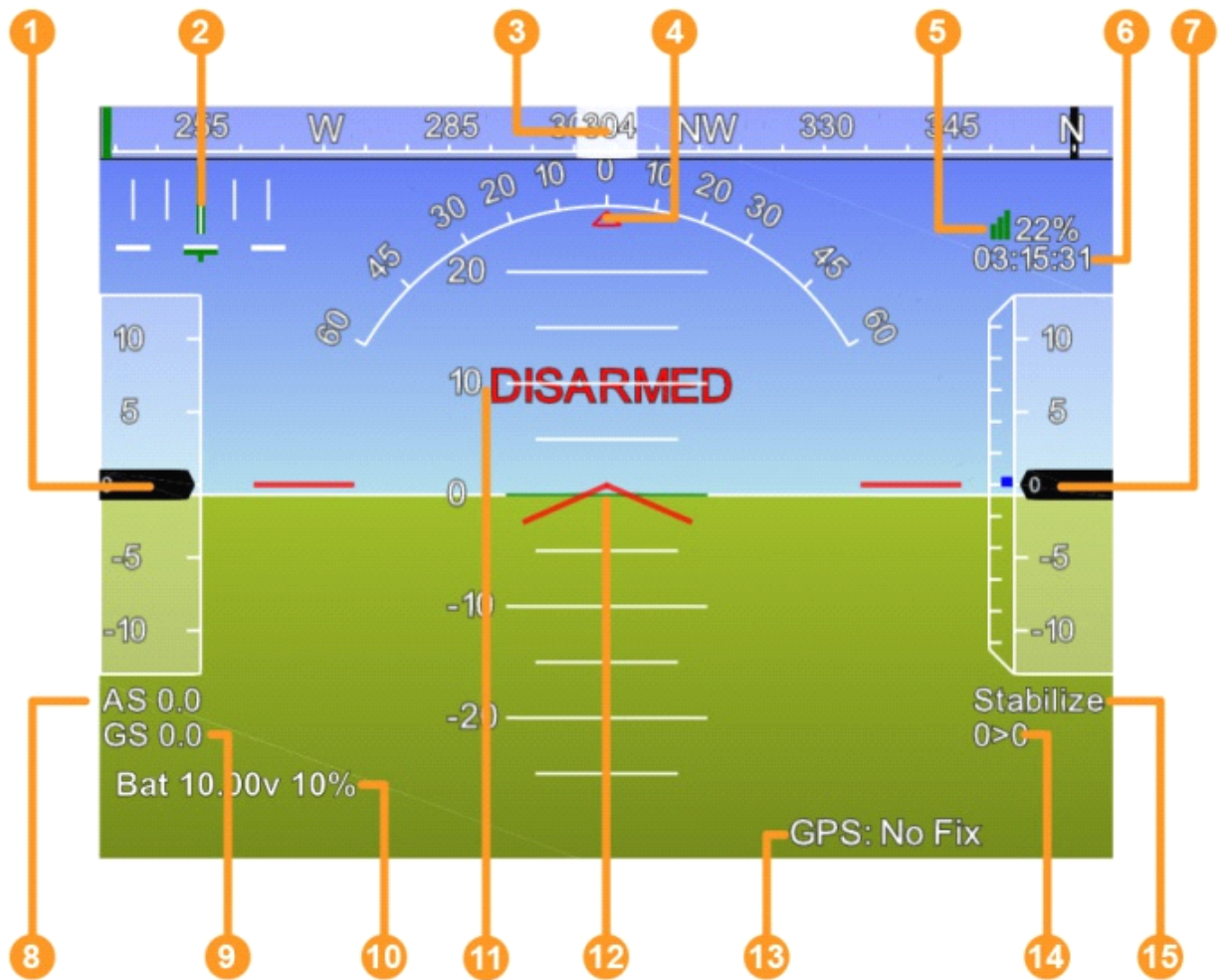


Рис. 3.4 – Вікно HUD

1. Air speed або Ground speed – швидкість відносно землі (або швидкість щодо повітря, якщо встановлений відповідний датчик);
2. Crosstrack error and turn rate – помилка слідування треку та швидкість повороту;
3. Heading direction – курс (напрямок носа ЛА);
4. Bank angle – кут крену;
5. Wireless telemetry connection – якість передачі телеметрійних даних (% втрачених пакетів);
6. GPS time – час за даними, які були отримані від GPS;
7. Altitude – висота за даними АРМ (синя метка, яка показує швидкість змінення висоти);

8. Air speed – повітряна швидкість (відносно повітря), показується тільки при наявності датчика;
9. Ground speed – швидкість відносно землі;
10. Battery status – стан батареї;
11. Artificial Horizon – штучний горизонт, (завжди реверсивен по відношенню до крену);
12. Pitch angle – Кут тангажу;
13. GPS status – стан GPS-модуля;
14. Distance to current waypoint > Current waypoint number - відстань до до поточної маршрутної точки та її номер;
15. Current autopilot mode – поточний режим автопілоту.

На карті буде показуватися поточна позиція тільки після того, як відбудеться визначення координат по GPS. Або в будь-якому випадку - при використанні симулятора польоту.

Слід в будь-який момент розуміти, який режим автопілота включений. Для того, щоб польотний контролер коректно сприймав команди MAVlink від APM Planner, він повинен працювати в одному з «автопілотні» режимах: Stabilize, Fly-by-Wire, Auto і т.п. Якщо ж контролер знаходиться в режимі Manual, то всі команди будуть ігноруватися.

Подвійний клік на вікні HUD дозволить його відчинити на весь екран на другому екрані.

Для того, щоб використовувати інші картки замість Google Maps, потрібно натиснути Ctrl-F і завантажити власні картки, які повинні бути підготовлені програмою Global Mapper.

Однією з особливостей APM Planner є можливість управляти БПЛА не вдаючись до допомоги пульта і не готуючи заздалегідь маршрут польоту. У вікні карти в будь-якому місці можна натиснути правою кнопкою миші і

вибрати "Fly To Here", тобто «лети сюди». ЛА полетить в задану точку і зависне в очікуванні наступної команди. Це і є режим "Guided Mode", який володіє і іншими цікавими можливостями (рис. 3.5).

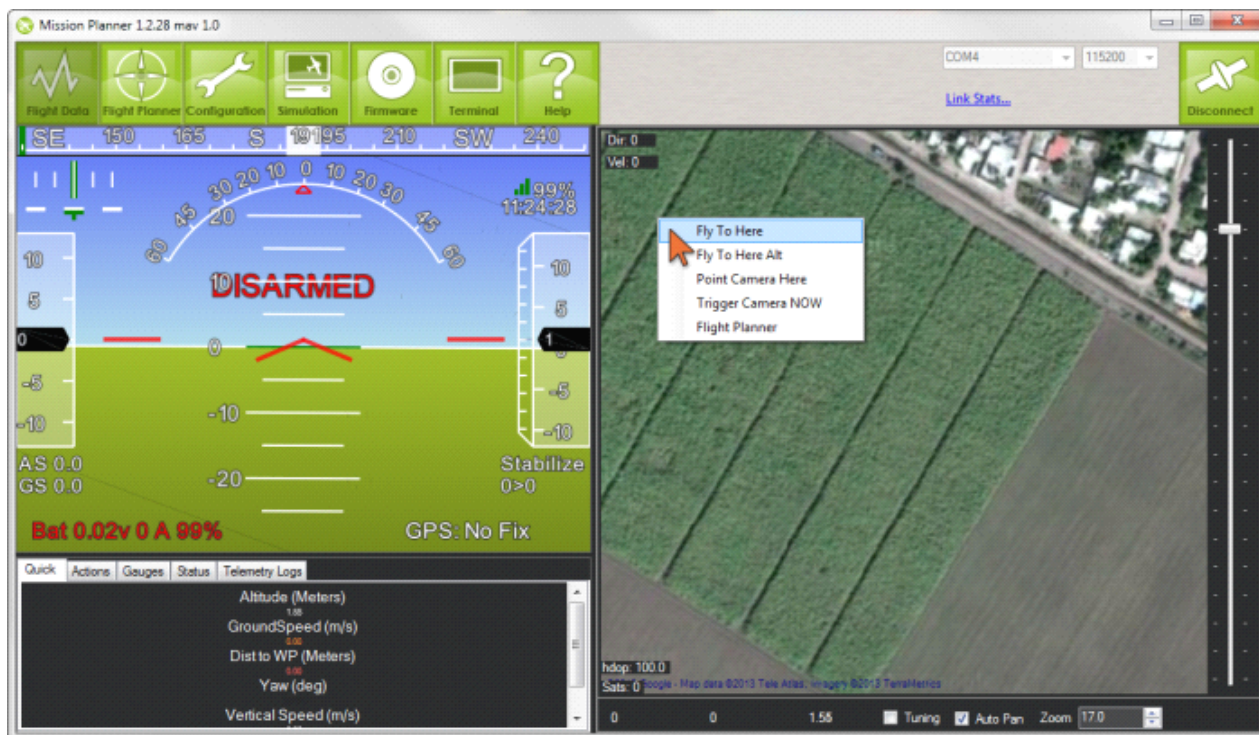


Рис. 3.5 – Режим «Guided Mode»

Увага: "Guided Mode" - це самостійний режим польоту. Вибравши його, польотний контролер буде залишатися в цьому режимі до тих пір, поки не буде обраний будь-якої інший режим. Наприклад, якщо вибрати точку на карті і дати команду "Fly To Here", то після прибуття в неї ЛА буде зависати над нею доти, поки з АРМ Planner не поступить наступної команди, а саме: або команди переміститися в іншу точку, або перемикання в інший режим. Більш того, якщо перемикання відбудеться в режим Auto, то місія (політ за маршрутом) відновиться з тієї точки, де вона була перервана.

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'єкта, або ж аеропорту). У таких випадках пропонується використовувати один з варіантів пошуку найкоротшого шляху, що обходять заборонений об'єкт по найкоротшій траєкторії.

Таким чином, завдання дослідження полягає в пошуку найкоротшого шляху обльоту забороненого об'єкта, заданого вершинами багатокутника на координатній сітці (рис. 3.6).

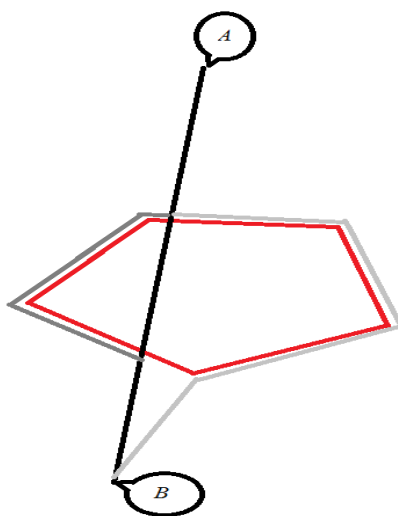


Рис. 3.6 – Обхід забороненої зони

Перегортаючи журнал інформація екрану HUD буде змінюватися і місце розташування апарата буде оновлюватись як це відбувається в польоті. Індивідуальні значення можна побачити через вкладку Status і відобразити на графіку натиснувши на "Tuning". Потім подвійним клацанням на даних

викликайте вікно з додатковими параметрами, які вам буде потрібно відобразити. Це покаже записані дані в процесі польоту (рис 3.7).



Рис. 3.7 – Відображення записаних даних

Створення 3D шляху польоту. Ви можете створити KMZ файли виконуючи наступне:

- Відкрийте 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 (рис 3.8 – 3.9).



Рис. 3.8 – Необхідні дії для відображення записаних даних у Google Earth

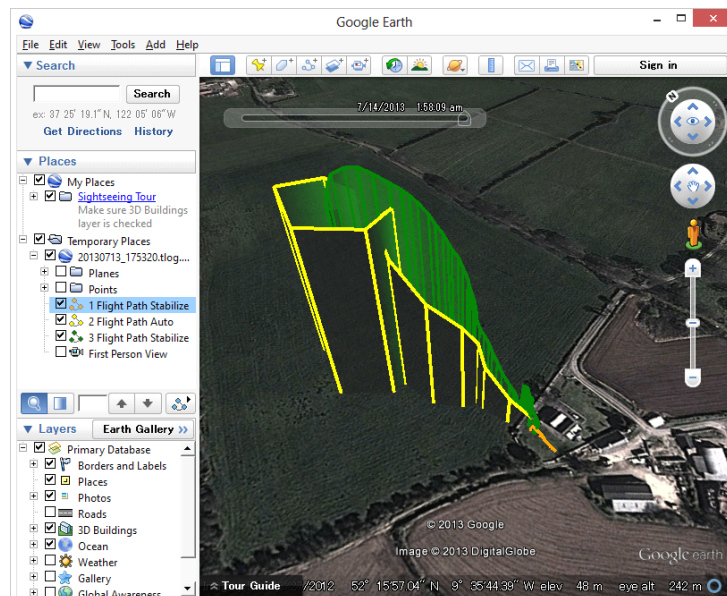


Рис. 3.9 – Відображення записаних даних у Google Earth

Різні режими польоту будуть відображені різними кольорами доріжок. Ви можете змінити деталі відображення шляху і кольору роблячи 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 редагування шляху
- Колір можна 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 точки з tlog слідуючи тим самим 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 переключившись 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 відкрийте 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, такими як "RC_CHANNELS" і "RAW_IMU".

Змінити величину графіка можна за допомогою колеса миші, вибираючи область графіка затисніть ліву клавішу або клацнувши правою кнопкою на графіку виберіть "Set Scale To Defaul" для скидання величини (рис. 3.10).

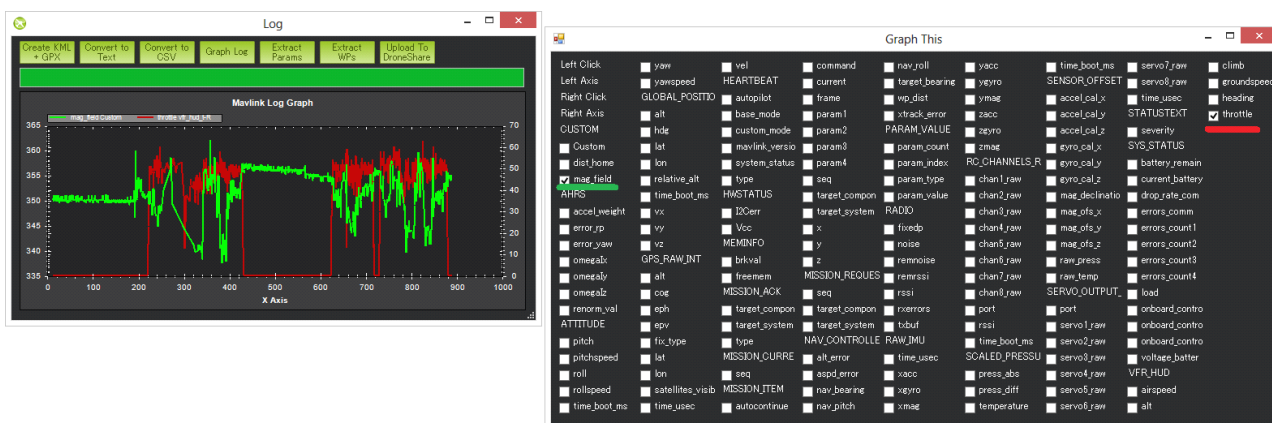


Рис. 3.10 – Зміна величин графіка

Режим пошуку найкоротшого шляху (рис. 3.11).



Рис. 3.11 – Режим пошуку найкоротшого шляху

Підключивши модуль `sway_comm.cpp` (`init.ini` - `use sway_comm`) потрапляємо у режим автоматичного визначення безпілотних зон і пошуку шляхів обльоту.

- Встановлюйте прапорець кілька разів що б вибрати маршрут обльоту.

- Змінити величину графіка можна за допомогою колеса миші, вибираючи область графіка затисніть ліву клавішу або клацнувши правою кнопкою на графіку виберіть "Set Scale To Defaul" для скидання величини.

Таким чином, сформульований ряд представляє практичний інтерес завдань знаходження маршруту обльоту легким безпілотним літальним апаратом заданих на земній поверхні точок. Завдання маршрутизації зведені до різних варіантів замкнутої і розімкнутої завдання алгоритму A*. Зокрема, для розімкнутої завдання розглянуті варіанти, коли є, або відсутня інформація про точки початку і кінця маршруту. При цьому мінімізується тривалість польоту за маршрутом, оскільки це забезпечує підвищення оперативності вирішення цільової завдання.

Обговорювані завдання маршрутизації математично формалізовані як завдання лінійного цілочисельного програмування з булевими змінними. Кожна з розглянутих постановок має особливості в запису обмежень і цільової функції. В обчислювальному плані складання маршруту зведено до послідовного розв'язування задач цілочисельного лінійного програмування з додаванням умов, що виключають підцикли.

Таким чином, основним результатом роботи є розробка уніфікованого підходу до математичної формалізації і чисельному розв'язанню низки актуальних і практично затребуваних завдань маршрутизації польоту легкого БПЛА в поле постійного вітру.

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:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_d$$

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_d – витрати праці на підготовку документації.

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 0.9)
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:

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

Витрати праці на розробку алгоритму рішення задачі:

Витрати на складання програми по готовій блок-схемі:

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

– за умови комплексного налагодження завдання:

Витрати праці на підготовку документації:

де t_{dp} – трудомісткість підготовки матеріалів і рукопису.

t_{do} – трудомісткість редагування, печатки й оформлення документації

4.2. Витрати на створення програмного забезпечення

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

Заробітна плата виконавців визначається за формулою:

де t – загальна трудомісткість, людино-годин;

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

грн.

Вартість машинного часу, необхідного для налагодження програми на ЕОМ:

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

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

грн.

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

АСУП.

Очікуваний період створення ПЗ:

де B_k – число виконавців;

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

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 2 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 ARМ Planner, в сільському господарстві. У даній темі в якості оптимізації дрона буде розглянута програма ARМ 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.

Були проаналізовані різноманітні програмні продукти-скетчі на виявлення недоліків традиційного підходу до програмування дронів, а саме:

- Програма-скетч Ardupilot_copter (v. 3.5.2);
- Програма-скетч AeroQuad (v. 3.2);
- Програма-скетч Phantom 3 (v. 1.9).

На підставі аналізу різноманітних програмних продуктів, було вдосконалено:

- Оновлена база даних зон з обмеженням польотів;
- Виправлена помилка, що приводить до автоматичного перезапуску батареї в холодних умовах експлуатації;

- Покращена стабілізація;
- Підвищен захист даних дрону;
- З’явилася можливість встановлювати «критичні зони».

В мережі інтернет є багато різноманітних програмних продуктів для програмування дронів, але, на думку автора, для кожної конкретної задачі необхідно, обмежена кількість режимів польоту, низький рівень стабілізації літального апарату та інші. Рішенням цих недоліків є можливість доопрацювання своє налаштування необхідних замовнику характеристик. На жаль, часто системи управління дроном мають істотні недоліки, наприклад: низький рівень надійності існуючої системи управління дроном. Незважаючи на те, що системи, що дозволяють організувати на їх основі власні розробки, все ж існують, вони дозволяють створити надійну і відповідальну власним вимогам систему одночасно.

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		
Інвестиції на ПЗ	21459	-	-	-	-	-	21459	4292
Витрати до впровадження ПЗ	-	15300	15000	15700	15400	18100	79500	15900
- 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
Витрати після впровадження ПЗ	-	4500	4700	4850	5000	5150	24200	4840
- 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 (4,9 роки) складе 439 грн, тобто відповідає умовам ефективності, тому що $NPU > 0$.

Середній термін окупності капіталовкладень складе 4,9 року.

Індекс прибутковості за 4,9 роки складе 1,02, тобто $ID > 1$, проект слід прийняти.

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

Висновки

В ході проведеної роботи були вивчені різні алгоритми пошуку найкоротшого шляху. У процесі вивчення були запропоновані та протестовані нові способи вирішення задач пошуку найкоротшого шляху для розрахунку траєкторії безпілотного літального апарату.

В результаті були проаналізовані та протестовані найбільш придатні алгоритми пошуку найкоротшого шляху, алгоритми роботи пошуку геоданих та об'єднання цих двох сутностей в одному програмному продукті.

У роботі над проектом було проведено аналіз існуючих механізмів їх структури і компонентів, що відповідають за пошук найкоротшого шляху, аналіз та відображення геоданих в програмі APM Planner, так само аналіз в досягненні кращого результату генерації найкоротшого шляху. Крім того, було проведено огляд і порівняльний опис методів пошуку найкоротшого шляху для безпілотних літальних апаратів.

При впровадженні даної системи в механізм побудови траєкторії безпілотних літальних апаратів було впроваджено алгоритми пошуку, що автоматизують пошук заборонених зон та генерують найкоротший шлях для обльоту цих зон.

Список використаних джерел

1. Салычев О.С. Автопилот БПЛА с инерциальной интегрированной системой – основа безопасной эксплуатации беспилотных комплексов. http://www.uav.ru/articles/TeKnol_Autopilot.pdf
2. Чистяков Н.В. Что такое ДПЛА // <http://dpla.ru/Articles/WhatIsDPLA.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. – 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-perspektivy-razvitiya-rossijskih-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/OPK/Soderjanie/OPK-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 – 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. – 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.
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 Reade. – 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. –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 - 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 екрана (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:

Додаток А

Модифікований текст

```
#include <QtGui>
#include <QMetaObject>
#include "waypointitem.h"
#include "opmapwidget.h"

namespace mapcontrol
{

    OPMaPWidget::OPMaPWidget(QWidget *parent, Configuration *config) :
    QGraphicsView(parent),
        configuration(config),
        UAV(0),
        GPS(0),
        Home(0),
        followmouse(true),
        compass(0),
        showuav(false),
        showhome(false),
        diagTimer(0),
        showDiag(false),
        diagGraphItem(0)
    {
        setSizePolicy(QSizePolicy::Preferred, QSizePolicy::Preferred);
        core=new internals::Core;
        map=new MapGraphicItem(core,config);
        mscene.addItem(map);
        this->setScene(&mscene);
        this->adjustSize();

        connect(map,SIGNAL(zoomChanged(double,double,double)),this,SIGNAL(zoomChanged(double,double,double)));
        connect(map->core,SIGNAL(OnCurrentPositionChanged(internals::PointLatLng)),this,SIGNAL(OnCurrentPositionChanged(internals::PointLatLng)));
        connect(map->core,SIGNAL(OnEmptyTileError(int,core::Point)),this,SIGNAL(OnEmptyTileError(int,core::Point)));
    }
}
```

```

        connect(map->core,SIGNAL(OnMapDrag()),this,SIGNAL(OnMapDrag()));
        connect(map-
>core,SIGNAL(OnMapTypeChanged(MapType::Types)),this,SIGNAL(OnMapTypeChanged(MapType::Types)));
        connect(map->core,SIGNAL(OnMapZoomChanged()),this,SIGNAL(OnMapZoomChanged()));
        connect(map->core,SIGNAL(OnMapZoomChanged()),this,SLOT(emitMapZoomChanged()));
        connect(map-
>core,SIGNAL(OnTileLoadComplete()),this,SIGNAL(OnTileLoadComplete()));
        connect(map->core,SIGNAL(OnTileLoadStart()),this,SIGNAL(OnTileLoadStart()));
        connect(map-
>core,SIGNAL(OnTilesStillToLoad(int)),this,SIGNAL(OnTilesStillToLoad(int)));
        ShowDiagnostics(showDiag);
        this->setMouseTracking(followmouse);
        SetShowCompass(true);

    }
    void OPMMapWidget::SetShowDiagnostics(bool const& value)
    {
        showDiag=value;
        if(!showDiag)
        {
            if(diagGraphItem!=0)
            {
                delete diagGraphItem;
                diagGraphItem=0;
            }
            if(diagTimer!=0)
            {
                delete diagTimer;
                diagTimer=0;
            }
        }
        else
        {
            diagTimer=new QTimer();
            connect(diagTimer,SIGNAL(timeout()),this,SLOT(diagRefresh()));
            diagTimer->start(500);
        }
    }

    void OPMMapWidget::SetUavPic(QString UAVPic)
    {
        if(UAV!=0)
            UAV->SetUavPic(UAVPic);
        if(GPS!=0)

```

```

        GPS->SetUavPic(UAVPic);
    }

    UAVItem* OPMAPWidget::AddUAV(int id)
    {
        UAVItem* newUAV = new UAVItem(map, this);
        newUAV->setParentItem(map);
        UAVS.insert(id, newUAV);
        QGraphicsItemGroup* waypointLine = new QGraphicsItemGroup(map);
        waypointLines.insert(id, waypointLine);
        return newUAV;
    }

    void OPMAPWidget::AddUAV(int id, UAVItem* uav)
    {
        uav->setParentItem(map);
        QGraphicsItemGroup* waypointLine = new QGraphicsItemGroup(map);
        waypointLines.insert(id, waypointLine);
        UAVS.insert(id, uav);
    }

    void OPMAPWidget::DeleteUAV(int id)
    {
        UAVItem* uav = UAVS.value(id, NULL);
        UAVS.remove(id);
        if (uav)
        {
            delete uav;
            uav = NULL;
        }
        QGraphicsItemGroup* wpLine = waypointLines.value(id, NULL);
        waypointLines.remove(id);
        if (wpLine)
        {
            delete wpLine;
            wpLine = NULL;
        }
    }

    /**
     * @return The reference to the UAVItem or NULL if no item exists yet
     * @see AddUAV() for adding a not yet existing UAV to the map
     */

```

```

UAVItem* OPMAPWidget::GetUAV(int id)
{
    return UAVS.value(id, 0);
}

const QList<UAVItem*> OPMAPWidget::GetUAVS()
{
    return UAVS.values();
}

QGraphicsItemGroup* OPMAPWidget::waypointLine(int id)
{
    return waypointLines.value(id, NULL);
}

void OPMAPWidget::SetShowUAV(const bool &value)
{
    if(value && UAV==0)
    {
        UAV=new UAVItem(map, this);
        UAV->setParentItem(map);
        // FIXME XXX The map widget is here actually handling
        // safety and mission logic - might be worth some refactoring

connect(this, SIGNAL(UAVLeftSafetyBouble(internals::PointLatLng)), UAV, SIGNAL(UAVLeftSafet
yBouble(internals::PointLatLng)));

connect(this, SIGNAL(UAVReachedWayPoint(int,WayPointItem*)), UAV, SIGNAL(UAVReachedWayPoint
(int,WayPointItem*)));
    }
    else if(!value)
    {
        if(UAV!=0)
        {
            delete UAV;
            UAV=0;
        }
    }

    if(value && GPS==0)
    {
        GPS=new GPSItem(map, this);
        GPS->setParentItem(map);
    }
    else if(!value)

```

```

    {
        if(GPS!=0)
        {
            delete GPS;
            GPS=0;
        }
    }
}

void OPMapWidget::SetShowHome(const bool &value)
{
    if(value && Home==0)
    {
        Home=new HomeItem(map, this);
        Home->setParentItem(map);
    }
    else if(!value)
    {
        if(Home!=0)
        {
            delete Home;
            Home=0;
        }
    }
}

void OPMapWidget::resizeEvent(QResizeEvent *event)
{
    if (scene())
        scene()->setSceneRect(
            QRect(QPoint(0, 0), event->size()));
    QGraphicsView::resizeEvent(event);
    if(compass)
        compass->setScale(0.1+0.05*(qreal)(event-
>size().width())/1000*(qreal)(event->size().height())/600);

}

QSize OPMapWidget::sizeHint() const
{
    return map->sizeHint();
}

void OPMapWidget::showEvent(QShowEvent *event)
{
    connect(&mscene, SIGNAL(sceneRectChanged(QRectF)), map, SLOT(resize(QRectF)));
}

```

```

        map->start();
        QGraphicsView::showEvent(event);
    }
    OPMMapWidget::~OPMapWidget()
    {
        delete UAV;

        foreach(UAVItem* uav, this->UAVS)
        {
            delete uav;
        }

        delete Home;
        delete map;
        delete core;
        delete configuration;
        foreach(QGraphicsItem* i, this->items())
        {
            delete i;
        }
    }
    void OPMMapWidget::closeEvent(QCloseEvent *event)
    {
        core->OnMapClose();
        event->accept();
    }
    void OPMMapWidget::SetUseOpenGL(const bool &value)
    {
        useOpenGL=value;
        if (useOpenGL)
            setViewport(new QGLWidget(QGLFormat(QGL::SampleBuffers)));
        else
            setupViewport(new QWidget());
        update();
    }
    internals::PointLatLng OPMMapWidget::currentMousePosition()
    {
        return currentmouseposition;
    }

    void OPMMapWidget::mouseMoveEvent(QMouseEvent *event)
    {
        QGraphicsView::mouseMoveEvent(event);
        QPointF p=event->pos();
        p=map->mapFromParent(p);
    }

```

```

        currentmouseposition=map->FromLocalToLatLng(p.x(),p.y());
    }
    ////////////////////////////////////WAYPOINT////////////////////////////////////
WayPointItem* OPMapWidget::WPCreate()
{
    WayPointItem* item=new WayPointItem(this->CurrentPosition(),0,map);
    ConnectWP(item);
    item->setParentItem(map);
    return item;
}
void OPMapWidget::WPCreate(WayPointItem* item)
{
    ConnectWP(item);
    item->setParentItem(map);
}
void OPMapWidget::WPCreate(int id, WayPointItem* item)
{
    Q_UNUSED(id);
    static internals::PointLatLng lastPos;

    ConnectWP(item);
    item->setParentItem(map);

//    QGraphicsItemGroup* wpLine = waypointLines.value(id, NULL);
//    if (!wpLine)
//    {
//        wpLine = new QGraphicsItemGroup(map);
//        waypointLines.insert(id, wpLine);
//    }

//    if (!lastPos.IsEmpty())
//    {
//        wpLine->addToGroup(new TrailLineItem(lastPos, item->Coord(), Qt::red,
map));
//        lastPos = item->Coord();
//    }

// Add waypoint line
//    trail->addToGroup(new TrailItem(position,altitude,color,this));
//    if(!lasttrailline.IsEmpty())
//        trailline->addToGroup((new
TrailLineItem(lasttrailline,position,color,map)));
//    lasttrailline=position;

```



```

    }
    WayPointItem* OPMaPWidget::WPCreate(internals::PointLatLng const& coord,int const&
altitude)
    {
        WayPointItem* item=new WayPointItem(coord,altitude,map);
        ConnectWP(item);
        item->setParentItem(map);
        return item;
    }
    WayPointItem* OPMaPWidget::WPCreate(internals::PointLatLng const& coord,int const&
altitude, QString const& description)
    {
        WayPointItem* item=new WayPointItem(coord,altitude,description,map);
        ConnectWP(item);
        item->setParentItem(map);
        return item;
    }
    WayPointItem* OPMaPWidget::WPInsert(const int &position)
    {
        WayPointItem* item=new WayPointItem(this->CurrentPosition(),0,map);
        item->SetNumber(position);
        ConnectWP(item);
        item->setParentItem(map);
        emit WPInserted(position,item);
        return item;
    }
    void OPMaPWidget::WPInsert(WayPointItem* item,const int &position)
    {
        item->SetNumber(position);
        ConnectWP(item);
        item->setParentItem(map);
        emit WPInserted(position,item);
    }
    WayPointItem* OPMaPWidget::WPInsert(internals::PointLatLng const& coord,int const&
altitude,const int &position)
    {
        WayPointItem* item=new WayPointItem(coord,altitude,map);
        item->SetNumber(position);
        ConnectWP(item);
        item->setParentItem(map);
        emit WPInserted(position,item);
        return item;
    }
    WayPointItem* OPMaPWidget::WPInsert(internals::PointLatLng const& coord,int const&

```

```

altitude, QString const& description,const int &position)
{
    WayPointItem* item=new WayPointItem(coord,altitude,description,map);
    item->SetNumber(position);
    ConnectWP(item);
    item->setParentItem(map);
    emit WPInserted(position,item);
    return item;
}
void OPMAPWidget::WPDelete(WayPointItem *item)
{
    emit WPDeleted(item->Number());
    delete item;
}
void OPMAPWidget::WPDeleteAll()
{
    foreach(QGraphicsItem* i,map->childItems())
    {
        WayPointItem* w=qgraphicsitem_cast<WayPointItem*>(i);
        if(w)
            delete w;
    }
}
QList<WayPointItem*> OPMAPWidget::WPSelected()
{
    QList<WayPointItem*> list;
    foreach(QGraphicsItem* i,mscene.selectedItems())
    {
        WayPointItem* w=qgraphicsitem_cast<WayPointItem*>(i);
        if(w)
            list.append(w);
    }
    return list;
}
void OPMAPWidget::WPReNumber(WayPointItem *item, const int &newnumber)
{
    item->SetNumber(newnumber);
}

void OPMAPWidget::ConnectWP(WayPointItem *item)
{
    connect(item,SIGNAL(WPNumberChanged(int,int,WayPointItem*)),this,SIGNAL(WPNumberChanged(
int,int,WayPointItem*)));
}

```

```

connect(item, SIGNAL(WPValuesChanged(WayPointItem*)), this, SIGNAL(WPValuesChanged(WayPoint
Item*)));

connect(this, SIGNAL(WPInserted(int,WayPointItem*)), item, SLOT(WPInserted(int,WayPointItem
*)));

connect(this, SIGNAL(WPNumberChanged(int, int,WayPointItem*)), item, SLOT(WPRenumbered(int, i
nt,WayPointItem*)));
    connect(this, SIGNAL(WPDeleted(int)), item, SLOT(WPDeleted(int)));
}
void OPMaPWidget::diagRefresh()
{
    if(showDiag)
    {
        if(diagGraphItem==0)
        {
            diagGraphItem=new QGraphicsTextItem();
            mscene.addItem(diagGraphItem);
            diagGraphItem->setPos(10,100);
            diagGraphItem->setZValue(3);
            diagGraphItem->setFlag(QGraphicsItem::ItemIsMovable, true);
            diagGraphItem->setDefaultTextColor(Qt::yellow);
        }
        diagGraphItem->setPlainText(core->GetDiagnostics().toString());
    }
    else
        if(diagGraphItem!=0)
        {
            delete diagGraphItem;
            diagGraphItem=0;
        }
}

////////////////////////////////////
void OPMaPWidget::SetShowCompass(const bool &value)
{
    if(value && !compass)
    {
        compass=new
QGraphicsSvgItem(QString::fromUtf8(":/markers/images/compas.svg"));
        compass->setScale(0.1+0.05*(qreal)(this->size().width())/1000*(qreal)(this-
>size().height())/600);
        // compass->setTransformOriginPoint(compass-
>boundingRect().width(),compass->boundingRect().height());
        compass->setFlag(QGraphicsItem::ItemIsMovable, true);
    }
}

```

```

        mscene.addItem(compass);
        compass->setTransformOriginPoint(compass->boundingRect().width()/2, compass-
>boundingRect().height()/2);
        compass->setPos(55-compass->boundingRect().width()/2, 55-compass-
>boundingRect().height()/2);
        compass->setZValue(3);
        compass->setOpacity(0.7);

    }
    if(!value && compass)
    {
        delete compass;
        compass=0;
    }
}
void OPMMapWidget::SetRotate(qreal const& value)
{
    map->mapRotate(value);
    if(compass && (compass->rotation() != value)) {
        compass->setRotation(value);
    }
}
void OPMMapWidget::RipMap()
{
    new MapRipper(core, map->SelectedArea());
}

#define deg_to_rad      ((double)M_PI / 180.0)
#define rad_to_deg      (180.0 / (double)M_PI)
#define earth_mean_radius  6371    // kilometers

// *****
// return the bearing from one point to another .. in degrees

double OPMMapWidget::bearing(internals::PointLatLng from, internals::PointLatLng to)
{
    double lat1 = from.Lat() * deg_to_rad;
    double lon1 = from.Lng() * deg_to_rad;

    double lat2 = to.Lat() * deg_to_rad;
    double lon2 = to.Lng() * deg_to_rad;

    //    double delta_lat = lat2 - lat1;
    double delta_lon = lon2 - lon1;

```

```

    double y = sin(delta_lon) * cos(lat2);
    double x = cos(lat1) * sin(lat2) - sin(lat1) * cos(lat2) * cos(delta_lon);
    double bear = atan2(y, x) * rad_to_deg;

    bear += 360;
    while (bear < 0) bear += 360;
    while (bear >= 360) bear -= 360;

    return bear;
}

// *****
// return a destination lat/lon point given a source lat/lon point and the bearing and
// distance from the source point

internals::PointLatLng OPMaPWidget::destPoint(internals::PointLatLng source, double
bear, double dist)
{
    double lat1 = source.Lat() * deg_to_rad;
    double lon1 = source.Lng() * deg_to_rad;

    bear *= deg_to_rad;

    double ad = dist / earth_mean_radius;

    double lat2 = asin(sin(lat1) * cos(ad) + cos(lat1) * sin(ad) * cos(bear));
    double lon2 = lon1 + atan2(sin(bear) * sin(ad) * cos(lat1), cos(ad) - sin(lat1) *
sin(lat2));

    return internals::PointLatLng(lat2 * rad_to_deg, lon2 * rad_to_deg);
}
}

```

Додаток Б

ВІДГУК

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

Тема дипломного проекту - підвищення ефективності програмного забезпечення дронів з використанням програми APM Planner при моніторингу в сільському господарстві.

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 та способи вирішення задач пошуку найкоротшого шляху для розрахунку траєкторії безпілотного літального апарату в вільно розповсюджене програмне забезпечення АРМ 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 АРМ 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 Є.Є. Кудальцеву кваліфікацію «Інженер з комп'ютерних систем».

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

/Алексєєв М.О./

Додаток В

РЕЦЕНЗІЯ

**на дипломний проект студента групи 122м-16-1 Кудальцева Є.Є.
на тему: «Підвищення ефективності програмного забезпечення
дронів з використанням програми APM 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 щонадальше 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 вільно розповсюджене програмне забезпечення.

В дипломному проекті в результаті проведеної роботи були проведені 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, скоротит терміни автоматичного розрахунку траєкторії з урахуванням заборонених зон і заощадить час роботи користувача користувачів.

Студент Кудальцев Є.Є. 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 Є.Є. Кудальцеву кваліфікацію «Інженер з комп'ютерних систем».

Рецензент

Додаток Г

Перелік документів на диску

ім'я файлу	опис
Пояснювальна записка	
Пояснювальна записка.docx	Пояснювальна записка
програма	
project.zip	Архів. Містить вихідні коди програми.
презентація	
Презентація.pptx	Презентація дипломного проекту