

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

**Навчально-науковий інститут аерокосмічних технологій
Кафедра авіа- та ракетобудування**

«На правах рукопису»
УДК 629.735

До захисту допущено:
В. о. завідувача кафедри
_____ Олександр БОНДАРЕНКО
«__» _____ 2021 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Літаки і вертольоти»

зі спеціальності 134 «Авіаційна та ракетно-космічна техніка»

**на тему: «Оптимізація аеродинамічних характеристик планера на
основі статистичних даних польотів та моделі атмосфери»**

Виконав:

студент II курсу, групи АЛ-01мп
Тисячук Олег Вячеславович _____

Керівник:

Професор, к.т.н., професор кафедри АРБ
Сухов Віталій Вікторович _____

Консультант :

Асистент, магістр, асистент кафедри АРБ,
Предаченко Костянтин Олегович _____

Рецензент:

Доцент, к.т.н., доцент кафедри СКЛА
Сердюк Анатолій Анатолійович _____

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

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут аерокосмічних технологій
Кафедра авіа- та ракетобудування

Рівень вищої освіти – другий (магістерський)

Спеціальність – 134 «Авіаційна та ракетно-космічна техніка»

Освітньо-професійна програма «Літаки і вертольоти»

ЗАТВЕРДЖУЮ

В. о. завідувача кафедри

_____ Олександр БОНДАРЕНКО

«__» _____ 2021 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Тисячуку Олегу Вячеславовичу

1. Тема дисертації «Оптимізація аеродинамічних характеристик планера на основі статистичних даних польотів та моделі атмосфери», науковий керівник дисертації Сухов Віталій Вікторович, д.т.н., професор, затверджені наказом по університету від «25» жовтня 2021 р. № 3576-с

2. Термін подання студентом дисертації 06 грудня 2021 р.

3. Об'єкт дослідження: Треки польотів планерів

4. Предмет дослідження: Методика формування аеродинамічних характеристик планера на основі статистичних даних польотів та моделі атмосфери

5. Перелік завдань, які потрібно розробити

5.1. Аналітичний огляд сучасних робіт по оптимізації аеродинамічних характеристик планерів.

5.2. Аналіз особливостей польоту планера за маршрутом.

5.3. Збір та аналіз статистичних даних.

5.4. Розробка оптимізаційної моделі на основі зібраних даних.

5.5. Розробка геометрії крила планера на основі оптимізаційної моделі.

5.6. Визначення аеродинамічних характеристик запропонованого планера.

5.7. Порівняння отриманих аеродинамічних характеристик з вихідними.

5.8. Стартап-проект.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу

6.1. Аналітичний огляд сучасних робіт по оптимізації аеродинамічних характеристик планерів.

6.2. Аналіз особливостей польоту планера за маршрутом.

6.3. Збір та аналіз статистичних даних.

6.4. Розробка оптимізаційної моделі на основі зібраних даних.

6.5. Результати роботи алгоритму оптимізації.

6.6. Порівняння отриманих аеродинамічних характеристик з вихідними.

7. Орієнтовний перелік публікацій

7.1. Стаття у фаховому виданні.

7.2. Доповідь на науково-технічній конференції з публікацією тез.

8. Дата видачі завдання 05.10. 2020 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Аналітичний огляд сучасних робіт по оптимізації аеродинамічних характеристик планерів	до 20.12.2020 р.	
2.	Аналіз особливостей польоту планера за маршрутом.	до 31.05.2021 р.	
3.	Збір та аналіз статистичних даних.	до 15.09.2021 р.	
4.	Розробка геометрії крила планера на основі оптимізаційної моделі	до 30.10.2021 р.	
5.	Визначення аеродинамічних характеристик запропонованого планера.	до 10.11.2021 р.	
6.	Порівняння отриманих аеродинамічних характеристик з вихідними.	до 20.11.2021 р.	
7.	Розробка стартап-проекту.	до 06.12.2021 р.	
8.	Оформлення пояснювальної записки та ілюстративного матеріалу.	до 06.12.2021 р.	
9.	Перевірка на плагіат.	до 06.12.2021 р.	

Студент

Олег ТИСЯЧУК

Науковий керівник

Віталій СУХОВ

Реферат

Магістерська дисертація: «Оптимізація аеродинамічних характеристик планера на основі статистичних даних польотів та моделі атмосфери», 94 сторінки, 34 рисунки, 24 таблиць, 18 посилань.

Актуальність: серед усіх типів повітряних суден планери є найдосконалішими з точки зору аеродинаміки, оскільки володіють рекордними показниками аеродинамічної якості (порядку 40-60 одиниць). Для проектування нових зразків планерів використовують найдосконаліші методи аеродинамічних розрахунків, а при виготовленні використовують дорогі та трудомісткі технології з використанням найновіших матеріалів. Тому, в проектуванні конкурентного, сучасного планера необхідно використовувати нові підходи та моделі, оскільки стандартні методи аеродинамічного проектування та оптимізації аеродинамічних характеристик (далі АХ) не дають необхідного результату. Дана дисертація націлена на розробку такого методу та створення відповідної оптимізаційної моделі, яка дозволить суттєво покращити АХ планера.

Мета роботи: оптимізація аеродинамічних характеристик планера на основі статистичних даних треків польотів планеристів.

Об'єкт дослідження: треки польотів планерів.

Предмет дослідження: методика формування оптимальних геометричних характеристик планера на основі статистичних даних польотів та моделі атмосфери.

Методи дослідження: визначення аеродинамічних характеристик планера за «Теорією несучої лінії»; збір та обробка статистичних даних польотів планерів, зібраних з відкритих джерел; створення узагальненої моделі атмосфери на основі статистичних даних; оптимізація АХ планера на основі створеної моделі за допомогою генетичного алгоритму.

Наукова новизна одержаних результатів: створена математична модель атмосфери при польоті планера за маршрутом, на основі якої була створена оптимізаційна модель АХ планера; написана комп'ютерна програма,

яка вирішує оптимізаційну задачу знаходження оптимальних АХ планера з використанням генетичного алгоритму; на основі отриманої оптимізаційної моделі був проведений аеродинамічний розрахунок планера стандартного класу, АХ якого були порівняні з аналогом.

Практичне значення одержаних результатів: в рамках роботи над даною магістерською дисертацією були отримані математичні, оптимізаційна модель, а також створені комп'ютерні програми, які їх реалізують. Разом, це набір інструментів, які дозволяють з достатньою точністю проводити аеродинамічне проектування та оптимізацію вихідних АХ нових планерів.

Апробація результатів дисертації: підтвердження результатів дисертації проведене у розділі 7 цієї роботи. Були отримані позитивні тенденції до збільшення середньої швидкості проходження маршруту.

Публікації:

- Тези на III науково-практичній конференції студентів та молодих вчених «Авіа-ракетобудування: перспективи та напрямки розвитку».
- Стаття «До питання оптимізації геометрії крила планеру з урахуванням статистичних даних».

Ключові слова: планер, аеродинамічні характеристики, геометрична параметризація, оптимізація, Теорія несучої лінії, Генетичний алгоритм, модель атмосфери, політ за маршрутом, статистика.

Abstract

Master's thesis: "Optimization of the aerodynamic characteristics of the glider on statistical data and atmosphere model", 94 pages, 18 links, 34 drawings, 24 tables.

Relevance: Among all types of aircraft, gliders are the most advanced in terms of aerodynamics, as they have record values of glide ratio (about 40-60 units). Design process of new glider models claim the most advanced methods of aerodynamic calculations while the manufacturing process claim expensive and time-consuming technologies with use of advanced materials. Therefore, in the design of a competitive, modern glider it is necessary to use new approaches and models, as standard methods of aerodynamic design and optimization of aerodynamic characteristics do not give the desired result. This dissertation is aimed at developing such a method and creating an appropriate optimization model that will significantly improve the aerodynamic characteristics of the glider.

Purpose of the study: optimization of aerodynamic characteristics of the glider based on statistical data from glider flight tracks.

The object of study: tracks of glider flights.

Subject of research: the method of formulating of the optimal geometric characteristics of the glider based on statistical data from the flight tracks and the model of the atmosphere.

Methods of research: determination of aerodynamic characteristics of the glider according to the "Lifting Line Theory"; collection and processing of glider flight statistics collected from open sources; the creation of a generalized model of the atmosphere based on statistical data; optimization of aerodynamic characteristics of glider based on the created model using a genetic algorithm.

The results and their novelty: a mathematical model of the atmosphere during the glider flight along the route has been created, which was the base for the optimization model of the aerodynamic characteristics of created glider; computer program that solves the optimization problem of finding the optimal aerodynamic characteristics of glider using a genetic algorithm has been written; on the basis of the received optimization model the aerodynamic calculation of the glider of a standard class which AH were compared with analog was carried out.

Recommendations for the use of work: as part of the work on this master's dissertation, mathematical and optimization models were obtained, as well as computer programs were created to implement them. Together, this is a set of tools that allow you to perform aerodynamic design and optimization of aerodynamic characteristics of new gliders with sufficient accuracy.

Approbation of the results of the dissertation: confirmation of the results of the dissertation has been carried out in section 7 of this work. There were positive trends to increase the average speed of the cross-country flight.

Publications:

Thesis at the III scientific-practical conference of students and young scientists "Aircraft and Rocket Engineering: perspectives and directions of development"

Publication "ON THE ISSUE OF OPTIMIZATION OF THE GEOMETRY OF THE PLANE'S WING BASED ON STATISTICAL DATA".

Key words: sailplane, aerodynamic characteristics, geometry parameterization, optimization, Lifting Line Theory, Genetic algorithm, atmosphere model, cross-country flight, statistics.

Зміст

Перелік умовних позначень	9
Вступ.....	10
1. АНАЛІТИЧНИЙ ОГЛЯД СУЧАСНИХ РОБІТ ПО ОПТИМІЗАЦІЇ АЕРОДИНАМІЧНИХ ХАРАКТЕРИСТИК ПЛАНЕРІВ.	11
1.1 Традиційні методи проектування.	11
1.2 Методи контролю ламінарного примежового шару.....	11
1.3 Компонування та місцева аеродинаміка.	12
1.4. Проектування аеродинамічних профілів.	14
1.5. Оптимізація на основі моделі атмосфери.	15
Висновки по розділу	17
2. АНАЛІЗ ОСОБЛИВОСТЕЙ ПОЛЬОТУ ПЛАНЕРА ЗА МАРШРУТОМ.....	18
2.1. Моделювання польоту планера за маршрутом.	18
2.2. Продуктивність набору висоти у потоках.	22
Висновки по розділу	24
3. ЗБІР ТА АНАЛІЗ СТАТИСТИЧНИХ ДАНИХ.....	25
Висновки по розділу	28
4. РОЗРОБКА ОПТИМІЗАЦІЙНОЇ МОДЕЛІ НА ОСНОВІ ЗІБРАНИХ ДАНИХ.	29
4.1. Розрахунок аеродинамічних характеристик планера методом LLT	29
4.2. Генератор геометрії крила.	31
4.3. Генетичний алгоритм	33
Висновки по розділу	38
5. РОЗРОБКА ГЕОМЕТРІЇ КРИЛА ПЛАНЕРА НА ОСНОВІ ОПТИМІЗАЦІЙНОЇ МОДЕЛІ.	39
5.1. Результати роботи генетичного алгоритму.	39
5.2. Створення просторової моделі крила.	41
Висновки по розділу	42
6. ВИЗНАЧЕННЯ АЕРОДИНАМІЧНИХ ХАРАКТЕРИСТИК ЗАПРОПОНОВАНОГО ПЛАНЕРА.	43
Висновки по розділу	49

Подп. и дата
Взам. инв. №
Инв. № дубл.
Подп. и дата
Инв. № подл

АЛО1мп0107.14.00.00.00 ПЗ					
Лит	Изм.	№ докум.	Подп.	Дата	
Разраб.	Олег Тисячук				
Пров.	Сухов В.В.				
Т. контр.					
Н. контр.	Поваров С.А.				
Утв.	Бондаренко О.М.				
Оптимізація аеродинамічних характеристик планера на основі статистичних даних польотів та моделі атмосфери			Лит	Лист	Листов
				7	88
КПІ ім. Ігоря Сікорського Каф. АРБ гр. АЛ-01мп					

7. ПОРІВНЯННЯ ОТРИМАНИХ АЕРОДИНАМІЧНИХ ХАРАКТЕРИСТИК З ВИХІДНИМИ	50
Висновки по розділу	53
8. СТАРТАП ПРОЕКТ	54
8.1. Опис ідеї проекту	54
8.2. Технологічний аудит ідеї проекту	55
8.3. Аналіз ринкових можливостей запуску стартап-проекту	57
8.4. Розроблення ринкової стратегії проекту	66
8.5. Розроблення маркетингової програми стартап-проекту	68
Висновки по розділу	71
ВИСНОВКИ	72
.Список використаних джерел	74
Додаток 1	75
Додаток 2	78

Перелік умовних позначень

ПЗ – пояснювальна записка;

ТЗ – технічне завдання;

ЛА – літальний апарат;

АХ – аеродинамічні характеристики

ГА – генетичний алгоритм;

ПШ – примежовий шар;

EASA – European Aviation Safety Agency (Європейська Агенція Авіаційної Безпеки);

КМ – композиційні матеріали;

ГО – горизонтальне оперення;

ВО – вертикальне оперення;

CS – Certification Specification (Сертифікаційна Специфікація);

FAI – Міжнародна авіаційна федерація;

СУ – силова установка.

Вступ

Важко переоцінити внесок планеризму у світову авіацію, оскільки саме з планерів почалося покорення неба апаратами, важчими за повітря. Так планери Отто Лілієнталя піднялися в небо за довго до першого польоту братів Райт. Згодом, з розвитком моторної авіації, планери почали використовуватись як шкільна парта для пілотів, оскільки вони вчили пілотів відчувати машину і безпомилково розраховувати захід на посадку. У сучасному світі планери і досі використовують для навчання та тренування пілотів; планеризм також існує як спортивна дисципліна. Пілоти які мають досвід у пілотуванні планерів досі високо цінуються у авіакомпаніях, а сучасні планери поєднують у своїй конструкції передові рішення.

Серед усіх типів повітряних суден планери є найдосконалішими з точки зору аеродинаміки, оскільки володіють рекордними показниками аеродинамічної якості (порядку 40-60 одиниць). Ця особливість витікає з того факту, що планери не використовують силову установку (далі СУ) для горизонтального польоту, а натомість використовують силу гравітації та енергію атмосферних мас. В результаті кожна лишня одиниця аеродинамічної якості має величезну цінність, оскільки від неї залежить чи зможе пілот завершити маршрут, чи доведеться передчасно здійснювати посадку в непідготовленому місці. Саме через це для проектування нових зразків планерів вимагає найдосконаліших методів аеродинамічних розрахунків, а виробництво – використання дорогих та трудомістких технологій зі впровадженням найновіших матеріалів. Тому, в проектуванні конкурентного, сучасного планера необхідно використовувати нові підходи та моделі, оскільки стандартні методи аеродинамічного проектування та оптимізації АХ не дають необхідного результату. Існує багато методів та теорій для максимально ефективного використання уже створеного планера при поточних погодних умовах, тоді як дана робота націлена на створення для конструкторів інструментів проектування максимально ефективного планера.

1. АНАЛІТИЧНИЙ ОГЛЯД СУЧАСНИХ РОБІТ ПО ОПТИМІЗАЦІЇ АЕРОДИНАМІЧНИХ ХАРАКТЕРИСТИК ПЛАНЕРІВ.

1.1 Традиційні методи проектування. Особливістю аеродинамічного проектування планерів є той факт, що основним методом збільшення аеродинамічної якості планера є збільшення площі ламінарного примежового шару на поверхні планера. Через це, ключовою вимогою до методу проектування є точне визначення положення переходу ламінарного ПШ в турбулентний. В експериментальній аеродинаміці для цього використовують аеродинамічні труби з низьким початковим ступенем турбулізації потоку, або літаючі лабораторії. Обидва варіанти є досить дорогими та трудомісткими, що збільшують загальну ціну етапу проектування. В чисельній аеродинаміці положення переходу визначають на основі математичних моделей турбулентності (k - ϵ , k - ω , e^n). Такі методи є дешевшими, але менш точними і потребують підтвердження експериментом. Гарно себе зарекомендував панельно-вихоровий метод з використанням моделі турбулентності e^n . Верифікація даного методу проектування планера була виконана в бакалаврському дипломному проекті [1].

Наведені вище методи аеродинаміки використовуються разом з аналітичними методами, викладеними у різноманітній літературі за цією темою. Гарним прикладом є книга за авторства Фреда Томаса «Fundamentals of Sailplane Design» [2], в якій викладені основи аналітичного аеродинамічного проектування планера, які є класичними для будь-якого ЛА, а також основи оптимізації АХ планерів, яка є досить специфічною, у порівнянні з класичними ЛА.

1.2 Методи контролю ламінарного примежового шару. Як згадувалось вище, основною методом збільшення аеродинамічної якості планера є збільшення площі ламінарного ПШ. Це забезпечується так званими засобами контролю примежового шару. У статті «Refinement of Glider Aerodynamic Design using CFD» [3], яка вивчає методи покращення АХ планера, розглядається вплив на АХ так званого «всмоктуючого пристрою» (див. рис. 1.1), який встановлюється на обшивці фюзеляжу перед коренем крила. Це багатообіцяючий засіб, який винайшли та запатентували у Північно-західному Університеті, який комбінує одразу два

принципи. По-перше, всмоктування через щілину низькоенергетичного примежового шару дозволяє попередити його турбулізацію. При цьому високоенергетичний ПШ зміщується нижче по потоку, в сторону проти негативного градієнту тиску, до моменту турбулізації. Це також попереджає явище відриву потоку. По-друге, знижується тиск в точці застою на передній кромці крила, що знижує опір тиску крила.

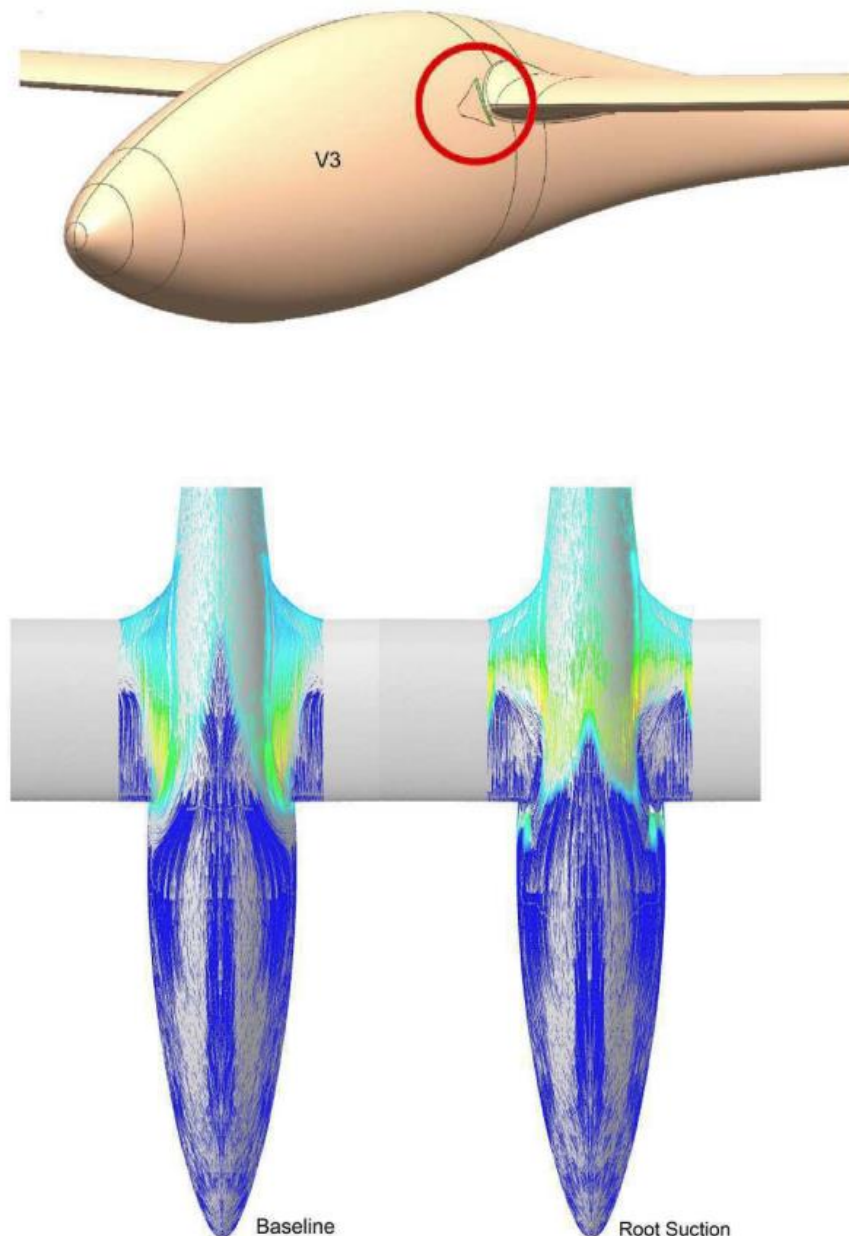


Рис. 1.1. Результати застосування методів контролю ламінарного ПШ.

Візуалізацію роботи даного пристрою можна побачити на рис. 1.1

1.3 Компонування та місцева аеродинаміка. В проектуванні планерів також застосовують методи оптимізації місцевої аеродинаміки та компоновання

агрегатів. В [3] наведено порівняння двох варіантів взаємного розміщення крила та фюзеляжу – високоплан та середньоплан, а також вплив на АХ.

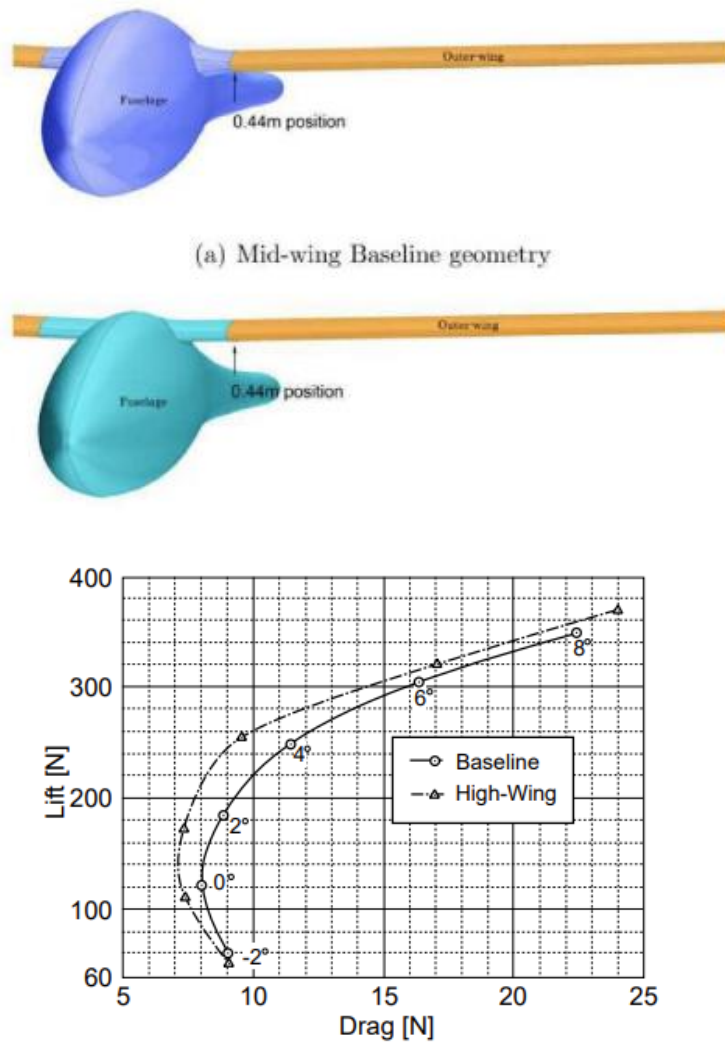


Рис. 1.2. Вплив компоновання крила та фюзеляжу на АХ.

За результатом проведеного дослідження, автор зробив висновок що високоплан є більш ефективним варіантом.

На рис 1.3. зображене порівняння аеродинамічного опору фюзеляжу для різних його форм. Таким чином, змінюючи кривизну і форму фюзеляжу можна досягти меншого опору.

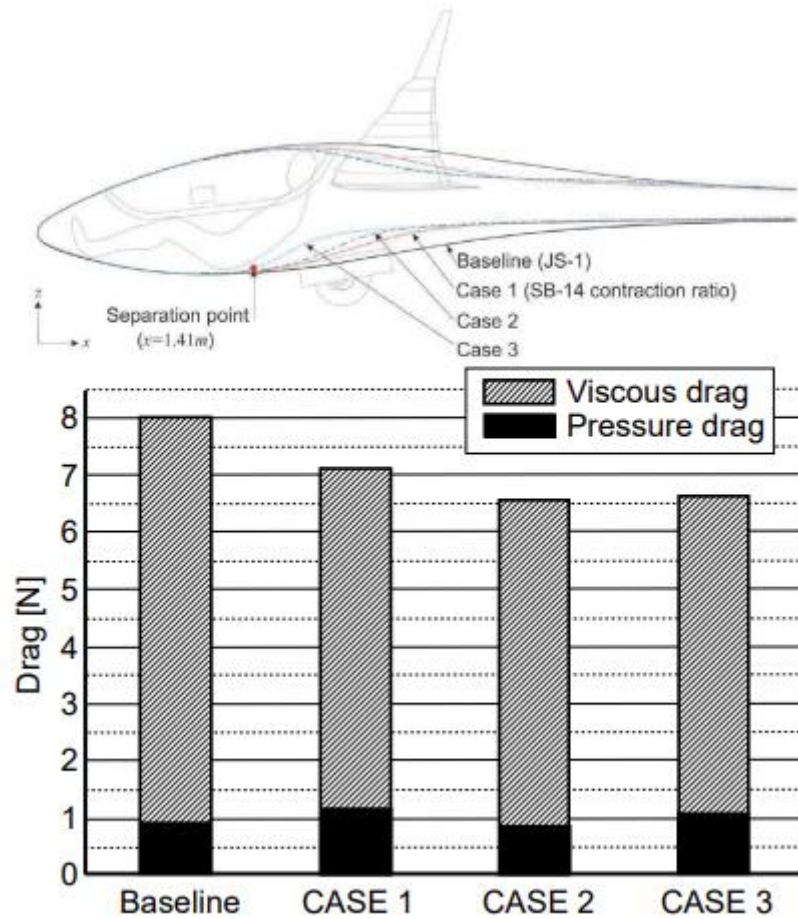


Рис. 1.3. Дослідження опору фюзеляжу в залежності від його форми.

1.4. Проектування аеродинамічних профілів. Одним із найважливіших елементів в проектуванні планера є вибір профіля крила. Від цього напряму залежать більшість аеродинамічних характеристик. У планеробудуванні використовують особливий клас профілів – ламінарні. Такий профіль крила забезпечує більшу площу ламінарного обтікання що суттєво знижує аеродинамічний опір крила. Найбільш відомим розробниками ламінарних аеродинамічних профілів є Вортманн, Ешплер.

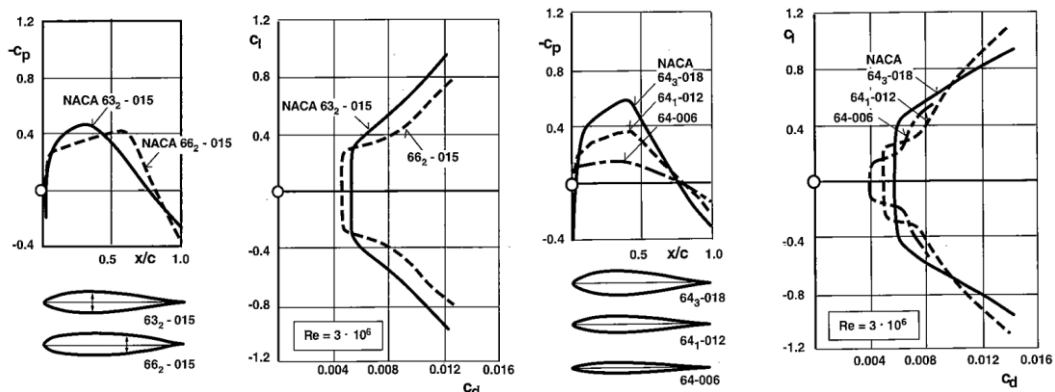


Рис. 1.4. Планерні профілі.

Великий внесок в розробку ламінарних аеродинамічних профілів внесли науковці з Делфтського технічного університету. Їхня серія профілів DU користується широкою популярністю серед сучасних виробників планерів. Їх профілі використовує південноафриканська компанія Jonker Sailplanes, німецькі Schempp-Hirth та Schleier. Перспективним напрямком у використанні ламінарних профілів є імплементація їх у вітряних електростанціях.

Як об'єкт оптимізації, аеродинамічні профілі становлять великий інтерес, адже невелике покращення характеристик профіля значною мірою відобразиться на характеристиках всього планера.

Складність оптимізації саме ламінарних профілів полягає у тому, що для її якісної реалізації необхідно точно обчислювати положення точки турбулізації ПШ навколо крила. За знаходження цієї точки відповідають моделі турбулентності газів та рідин, які є досить складними з математичної точки зору. На сьогоднішній день найбільше розповсюдження набули моделі турбулентності $k - \epsilon$, $k - \omega$, а також модель e^N [4]. Перші дві є досить точними, але потребують значних обчислювальних потужностей. Часто для швидкого аналізу турбулізації ПШ навколо профіля крила використовують модель e^N . Вона є простішою за моделі сімейства k - та швидше обчислюється на персональних комп'ютерах. Найвідомішою програмою, яка використовує цю модель для знаходження точки переходу ламінарного ПШ в турбулентний являється XFOIL. На базі цієї програми, написаної багато років тому розробили велику кількість різноманітних надбудов. Прикладом такої програми є Xflr5 який використовує XFOIL для обчислення в'язкого обтікання профілів і використовує ці результати для імплементації методів продувок тивимірних об'єктів – крил.

Розробка і оптимізація ламінарних профілів з часом стає все дорожче, оскільки ріст характеристик на декілька відсотків коштує все дорожче, якщо розглядати затрати на проведені дослідження.

1.5. Оптимізація на основі моделі атмосфери. Ще одним методом оптимізації АХ планерів, є підбір параметрів планера таким чином, щоб він максимально ефективно себе показував при певному узагальненому стані атмосфери. Ця особливість витікає зі специфіки польоту планера за маршрутом, яка

більш детально буде розглядатися в наступному розділі. Основи цього методу викладені в [2].

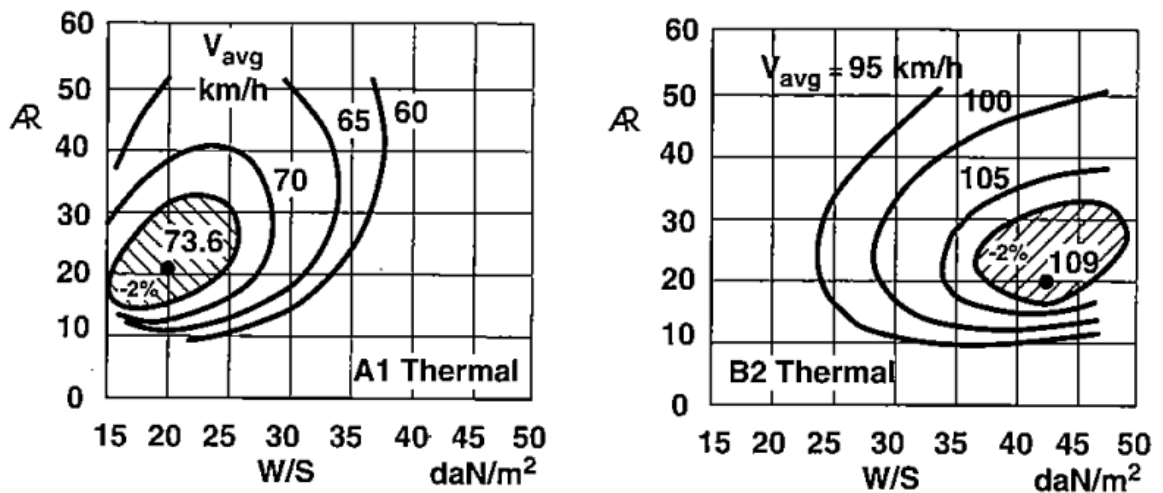


Рис. 1.5. Ефективність планера у різних потоках.

На рис. 1.5. зображені діаграми, які відображають ефективність планерів з різним подовженням крила та різним навантаженням на крило у різних типах термальних потоків. За цими діаграмами можна визначити геометричні характеристики планера, який би максимально ефективно перетворював енергію висхідних потоків теплого повітря у висоту. Проте, як буде розглядатися далі у цій роботі, це є одною зі складових польоту планера, тому сконцентрувавшись тільки на наборі висоти, втрачається ефективність на іншому етапі – переході. Як показує практика, більша середня швидкість у планера, який летить швидше, а не випарює краще.

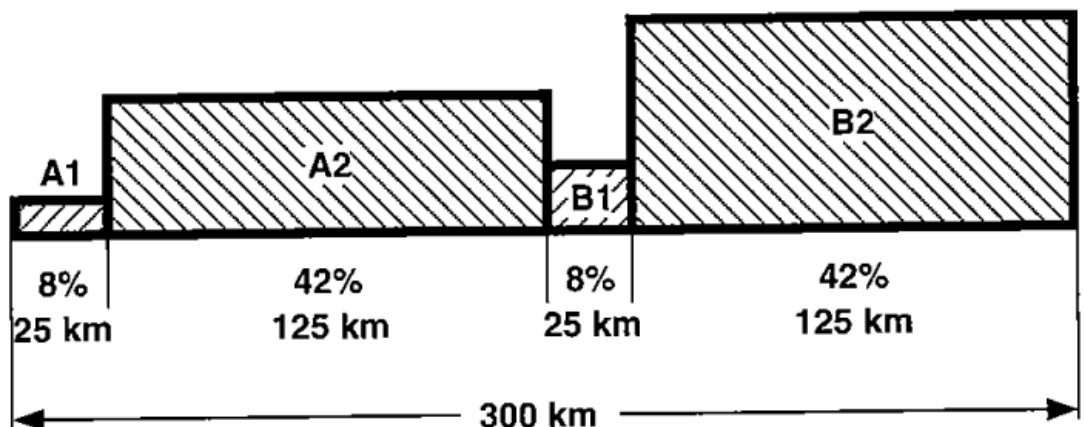


Рис. 1.6. Модель атмосфери при польоті за маршрутом.

Дана методика була раніше доступна тільки обмеженому колу людей з доступом до даних льотних випробувань планерів, але тепер, з розвитком технологій ця можливість відкрилась назагал. Цьому посприяло розповсюдження серед пілотів-планеристів польотних комп'ютерів які ведуть реєстрацію положення планера в просторі за допомогою системи GPS, а також мають можливість отримувати дані з бортових приладів. Дані, отримані з таких комп'ютерів дозволяють пілоту поліпшувати власну стратегію проходження маршруту та вміння керувати планером.

Висновки по розділу

В даному розділі були проаналізовані методи оптимізації аеродинамічних характеристик планерів, які застосовуються на сьогоднішній день. Були розглянуті методи поліпшення місцевої аеродинаміки, методи керуванням ламінарним примежовим шаром, а також метод оптимізації на основі моделі атмосфери.

В результаті був вибраний метод оптимізації за моделлю атмосфери, дані про яку буде отримано в розділі 4 цієї роботи з відкритих джерел.

2. АНАЛІЗ ОСОБЛИВОСТЕЙ ПОЛЬОТУ ПЛАНЕРА ЗА МАРШРУТОМ

Стандартний політ планера за маршрутом (англ. cross country flight) суттєво відрізняється від польоту класичного ЛА. Зазвичай політ ЛА складається з наступних фаз: зліт, набір висоти, крейсерський політ, зниження, приземлення, тоді як політ планера складається з наступних: зліт, набір висоти (у складі аеропотяга, з лебідки чи самостійний старт з використанням власної силової установки), відчеплення (вимкнення СУ), серія випарювань та переходів, зниження та посадка (рис. 2.1). В даному розділі буде розглянуті особливості таких польотів з точки зору оптимізації аеродинамічних характеристик.

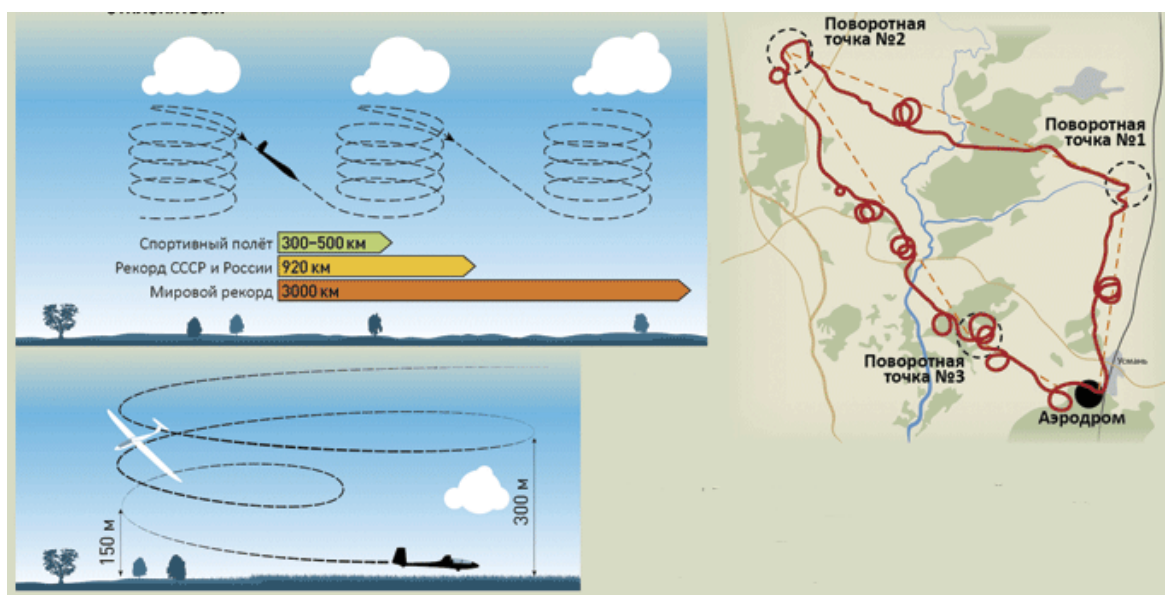


Рис. 2.1. Структура польоту планера за маршрутом.

2.1. Моделювання польоту планера за маршрутом. Процес оптимізації враховує вплив зміни окремих геометричних параметрів на загальну ефективність планера. Тому політ за маршрутом повинен бути виражений в математичній формі, яка об'єднує багато геометричних та конструктивних параметрів. Такі моделі є дещо спрощеними відображеннями реальних процесів, які протікають в процесі польоту планера та не можуть врахувати незліченну кількість варіацій подій та процесів, викликаних метеорологічними явищами та особливостями місцевого рельєфу. Основною задачею таких моделей є не оцінка дійсних швидкостей на маршруті, а створення базису для порівняння різних конструкторських та

аеродинамічних рішень при проектуванні, дозволяючи кількісно оцінити вплив окремих параметрів (подовження крила, вибір аеродинамічного профілю, тощо).

У найпростішій формі, як було зазначено вище, політ за маршрутом складається з серії наборів висоти у висхідних потоках та переходах (плануваннях) від потоку до потоку. В межах одного висхідного потоку, набір висоти (випарювання) має відбуватися в усталеній спіралі. Найпростіша математична модель маршруту планера припускає що сила потоку залишається незмінною з висотою. Також вона припускає відсутність затримки при переході від набору висоти до планування до наступного потоку, тобто час на пошук потоку не враховується. Сегмент такого маршруту зображений на рис. 2.2 [2].

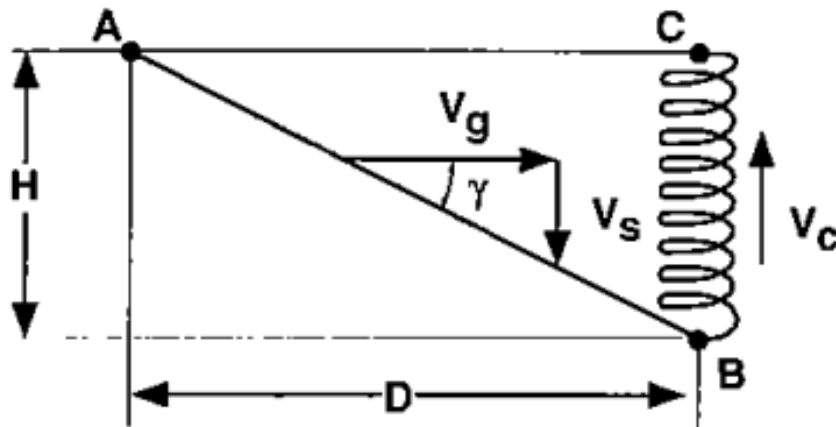


Рис. 2.2. Ідеалізований сегмент маршруту планера:

A-B – перехід (планування); B-C – набір висоти; V_c – швидкість набору висоти; V_s – швидкість зниження; V_g – швидкість на переході; H – висота, набрана в потоці; D – дистанція, пройдена при переході; γ – кут планування.

Середня швидкість на маршруті $V_{\text{сеп}}$ визначається як і середня шляхова швидкість часом t , необхідним для проходження шляху D до наступного потоку для набору висоти, втраченої при плануванні. Час t є сумою часу t_c , затраченого на набір висоти і t_g – часу, витраченому при переході.

$$V_{\text{сеп}} = \frac{D}{t} = \frac{D}{t_c + t_g} \quad (1)$$

Виразивши час через швидкості набору висоти та зниження, отримаємо наступне співвідношення:

$$t_c = \frac{H}{V_c};$$

$$t_g = \frac{H}{V_s};$$

$$D = \frac{V_g}{V_s} \cdot H;$$

$$\frac{V_{\text{сер}}}{V_g} = \frac{V_c}{V_c + V_s} \quad (2)$$

Швидкість набору висоти V_c є функцією від сили потоку V_T (власна вертикальна швидкість повітря всередині потоку) та V_{S_c} – швидкості зниження планера в спіралі:

$$V_c = V_T - V_{S_c} \quad (3).$$

Виходячи з цього, можна стверджувати що середня швидкість маршруту оптимізується шляхом досягнення наступних умов:

- зниження власної швидкості зниження V_{S_c} в спіралі;
- зниження власної швидкості зниження V_s при переході;
- збільшення горизонтальної швидкості на переході V_g .

Оскільки

$$\gamma \approx \tan \gamma = \frac{V_s}{V_g},$$

то кут планування γ може бути використаним замість власної швидкості зниження планера при переході V_s . Тоді необхідно визначити таке співвідношення величин V_{S_c} , γ та V_g , щоб середня швидкість $V_{\text{сер}}$ була максимальною. Звідси впливає основна складність оптимізації наведених параметрів. Необхідно забезпечити при великій швидкості планування V_g малий кут планування. Також складність при оптимізації додають зміни погодних умов.

На рис. 2.3. зображений графічний метод знаходження V_{S_c} , γ та V_g , при яких середня швидкість $V_{\text{сер}}$ буде найвищою.

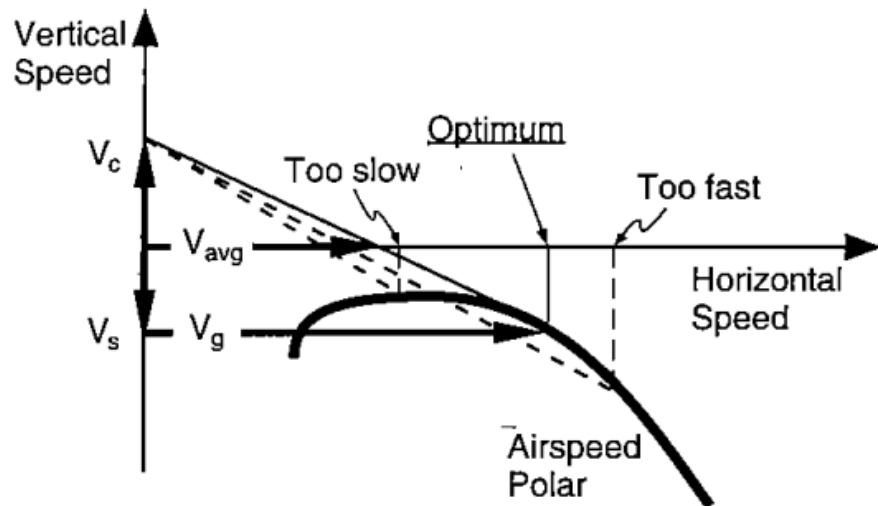


Рис. 2.3. Графічне знаходження максимальної середньої швидкості.

На рисунку зображена поляра швидкостей $V_s(V_g)$ разом з прямою лінією, проведеною з точки $(0, V_c)$ до робочої точки $(V_g, V_s(V_g))$. Для заданої величини швидкості переходу V_g буде справедливе наступне співвідношення:

$$\frac{V_{\text{сеп}}}{V_g} = \frac{V_c}{V_c + V_s}$$

Звідси випливає, що максимальну середню швидкість на маршруті можна досягти за умови, якщо пряма, проведена з точки $(0, V_c)$ буде дотичною до графіка поляри швидкостей. У разі зміни таких характеристик, як навантаження на крило, поправки на вітер поляра швидкостей буде зміщуватись.

2.2. Продуктивність набору висоти у потоках.

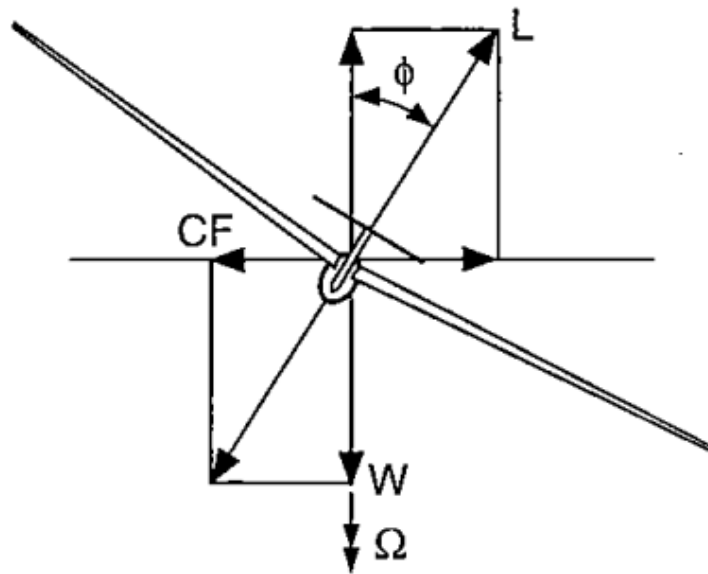


Рис. 2.4. Умова рівноваги планера у спіралі.

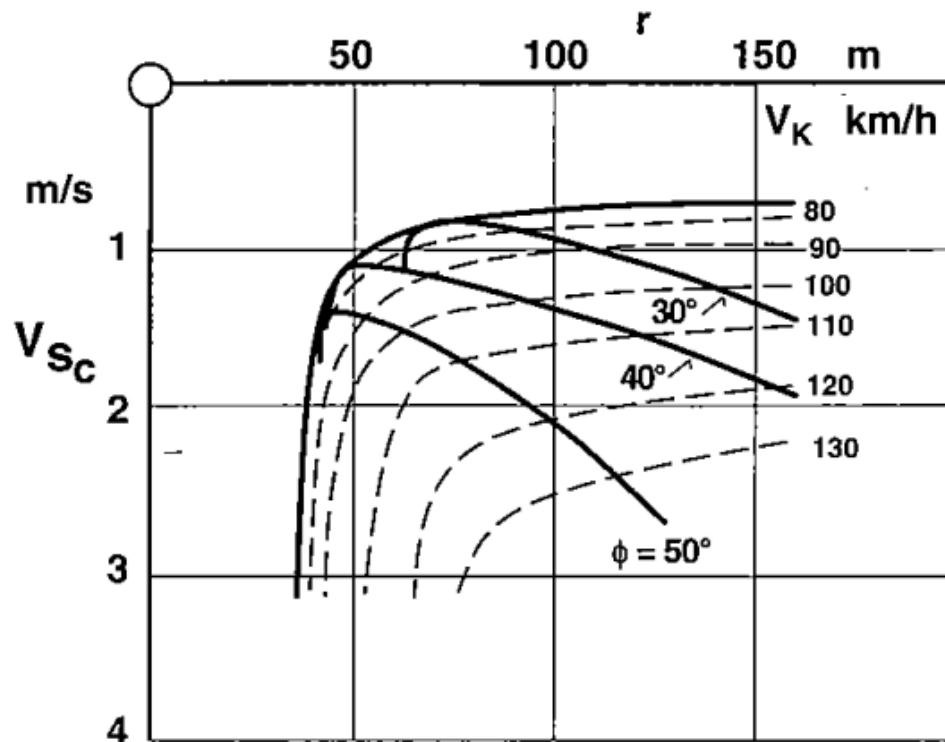


Рис. 2.5. Залежність швидкості зниження від радіусу спіралі для різних швидкостей V_k та кутів крену φ .

В природі зустрічається велика кількість термічних потоків. Різні географічні та кліматичні умови так само як і локальні метеорологічні феномени призводять до формування потоків різної сили, ширини та структури. Проте, при визначенні конструкції планера, детальне моделювання усіх можливих термічних потоків не є

необхідним кроком – достатньо загального опису декількох найбільш поширених типів потоків.

Раніше основними джерелами для дослідження термічних потоків були публікації результатів льотних випробувань. На даний момент, з розвитком сучасних технологій доступ до таких даних став простішим, а самих даних – більше. Тим не менше, не існує однастайності у поглядах щодо найбільш точної моделі термічних потоків. Історично, перші моделі термічних потоків опирались на модель Кармайкла [5], яка розділяла потоки на вузький-сильний, вузький-слабкий та широкий (див. рис. 2.6).

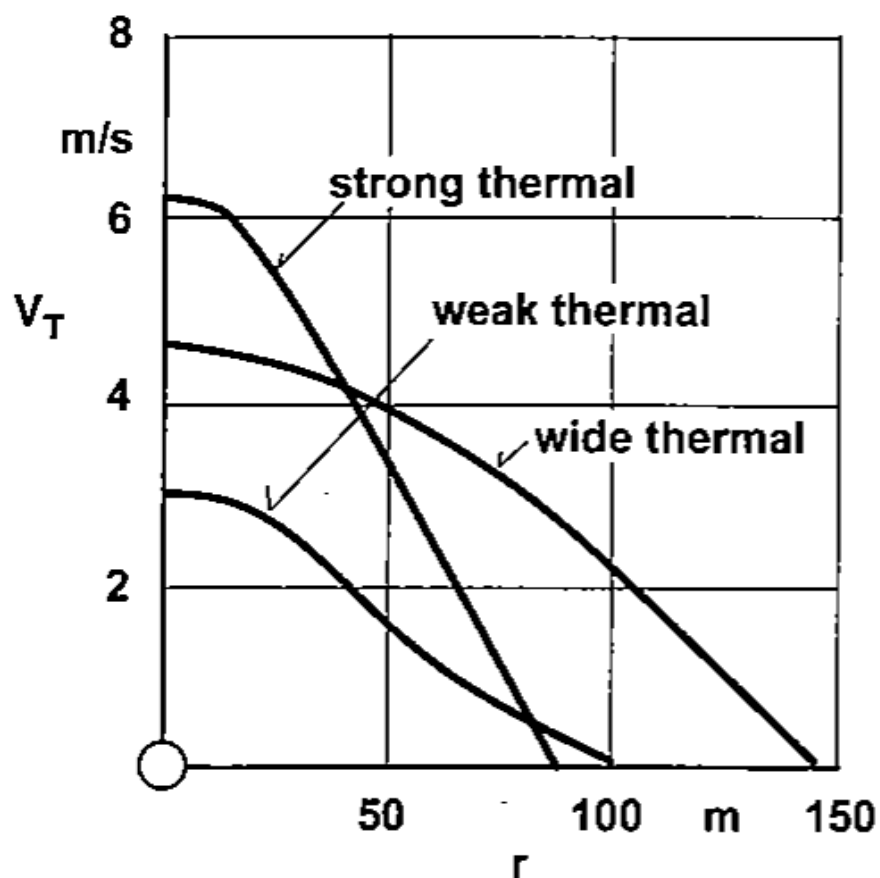
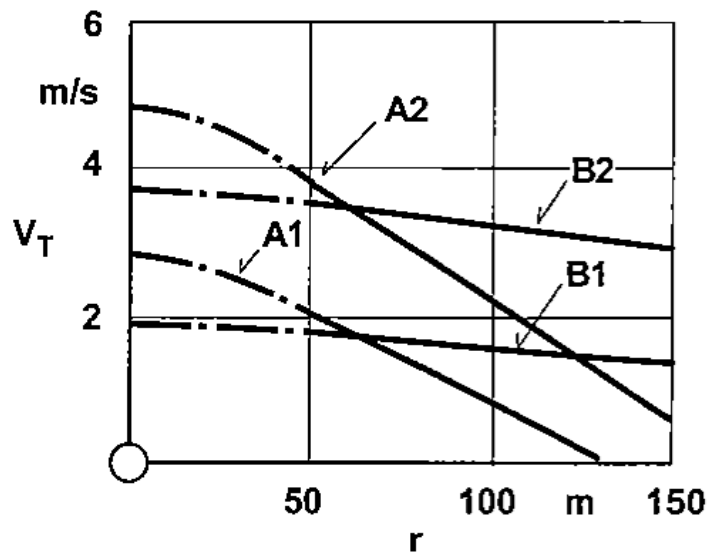


Рис. 2.6. Стандартні профіль за моделлю Кармайкла [5].

Хоча ця модель давала можливість отримувати доволі гарні результати, багато дослідників побачили необхідність в більш досконалий підхід. Базуючись на системному аналізі і результатах багатьох льотних випробувань, Хорстманн [6] запропонував удосконалену класифікацію термічних потоків по типам. Така класифікація є більш реалістичною, особливо при моделюванні атмосфери за типових Європейських погодних умов. Ця модель базується на типових потоках,

які відрізняються загальною силою потоку, та шириною. Профілі потоків за моделлю Хортсмана зображені на рис. 2.7. Кожен тип потоку має власний радіальний розподіл підйомної сили.



	A1	A2	B1	B2
V_T [m/s] at $r = 60\text{m}$	1.75	3.50	1.75	3.50
Gradient [$\frac{\text{cm}}{\text{s}}/\text{m}$]	2.5	3.20	0.45	0.60

Рис. 2.7. Профілі стандартних термічні потоків в моделі Хортсмана [2, 6]:

V_T – сила потоку; r – відстань від центру потоку.

Висновки по розділу

В даному розділі були розглянуті особливості польоту планера за маршрутом, досліджені математичні закони, які описують фази польоту планера. Зроблено висновок про важливість врахування стану атмосфери на маршруті, оскільки це напряму впливає на вибір швидкісного режиму, а також на загальну протяжність самого маршруту. Досліджена література, в якій описані математичні моделі атмосфери.

3. ЗБІР ТА АНАЛІЗ СТАТИСТИЧНИХ ДАНИХ

В рамках даної магістерської роботи було прийняте рішення для збору статистичних даних використовувати відкриті джерела.

Ресурс www.weglide.org надає відкриту базу даних треків польотів планеристів.

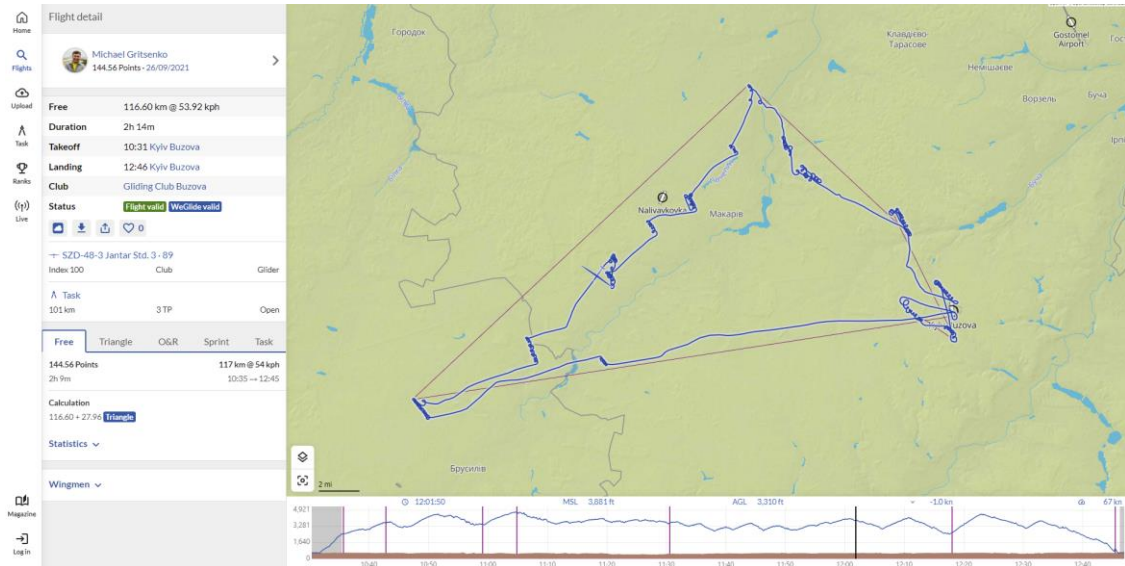


Рис. 3.1. Ресурс weglide.org.

Треки польотів зберігаються в файлах з розширенням `.igs`. Це файл зі стандартизованою структурою даних, в якому зберігаються координати, часові відмітки та інша інформація про політ планера. Цей стандарт також використовується в такій технології як `flarm`.

```

1 AFLA3AV
2 HFDTE060713
3 HFFXA500
4 HFFLTPilotincharge:MARKUS UHLIG
5 HPCM2Crew2:
6 HFGTYGliderType:SZD-48-3 JANTAR
7 HFGIDGliderID:D-7746
8 HFDTM100GPSDatum:WGS84
9 HFRFWFirmwareVersion:Flarm-IGC05.09
10 HFRHWHardwareVersion:Flarm-IGC06
11 HFFTYFRTType:Flarm-IGC
12 HFGPSu-blox:LEA-4P,16,8191
13 HFRSPPressAltSensor:Intersema MS5534B,8191
14 HFCCLCompetitionClass:Club
15 HFCIDCompetitionID:
16 IO23638FXA3940SIU
17 C220613091337000000000002Task
18 C0000000N000000000ETAKEOFF
19 C5038650N01307717EGrossrueckerswa
20 C5116433N01430417EKlix
21 C5135200N01314750EFalkenberg Bf
22 C5038650N01307717EGrossrueckerswa
23 C0000000N000000000ELANDING
24 B1223475038652N01307638EV008850000099900
25 F12235402141225042924
26 B1223555038652N01307638EV008910000099900
27 LFLA122357ID 2 DD8579
28 LFLA122357 STEALTH OFF
29 LFLA1223570B27.02.2008alps20080225_
30 LFLA122357EE0UTjjRUBCA?A?A?A?A?HGrrrrstu
31 LFLA122357EE1srrstuwvrsA?A?A?rssrA?rssrVV
32 LFLA122357EE2A?vwA?rsA?EDssh`aA?rersrsA?A?
33 LFLA122357EE3A?A?
34 F1223570214122504293124

```

Рис. 3.2. Структура файлу igc-формату.

В рамках даної дисертації була написана програма на мові Python для автоматизованого завантаження потрібних файлів з ресурсу і для аналізу даних, які у них містяться. Вихідний код програми наведений в додатку 1.

З множини всіх треків, які є у відкритому доступі була сформована наступна вибірка: тип судна: SZD-48-3 Jantar Standard 3, пілоти досвідчені (не юніори), довжина маршруту не менше 200 км. Ці критерії були обрані з наступних міркувань:

- планер SZD-48-3 є досить поширеним, його аеродинамічні та геометричні характеристики добре відомі
- досвідченість пілота зменшує шанс неефективної побудови маршруту; в рамках даної роботи висувається припущення що пілот обирає оптимальний маршрут, та оптимальні режими польоту

В результаті було отримано 84 файли. Результат обробки даних зображений на рис. 2.2.

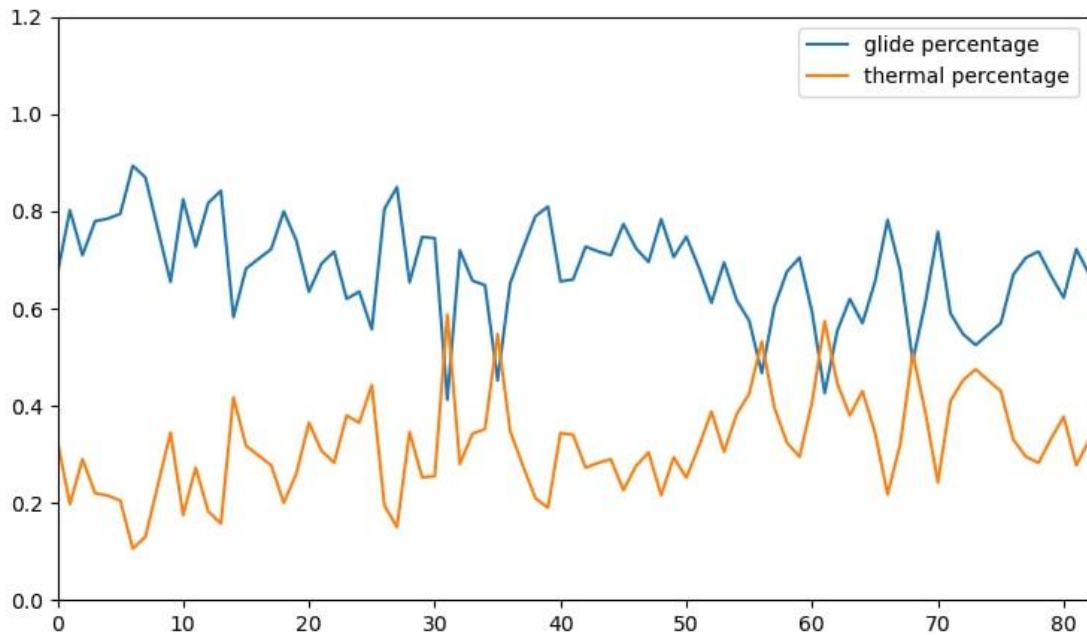


Рис. 3.3. Відношення часу при випарюванні і на переході.

В результаті було визначено, що відношення часу на переході до часу в потоках становить 69:31. Це дозволить сформувати більш точний критерій оптимізації.

Наступним кроком було визначення розподілу швидкостей по відносному часу. Тобто був побудований графік, по осі X якого був розподіл швидкостей, а по осі Y – відсоток часу, який планер провів на цій швидкості. Формула, по якій була обчислена ця залежність наведена нижче:

$$P(V_i) = \frac{t(V_i)}{T} \cdot 100\% \quad (4),$$

де V_i - швидкість, $t(V_i)$ – час, затрачений планером на політ на цій швидкості, T – загальний час польоту.

Таким чином було отримано залежність, графік якої зображений на рис. 3.4.

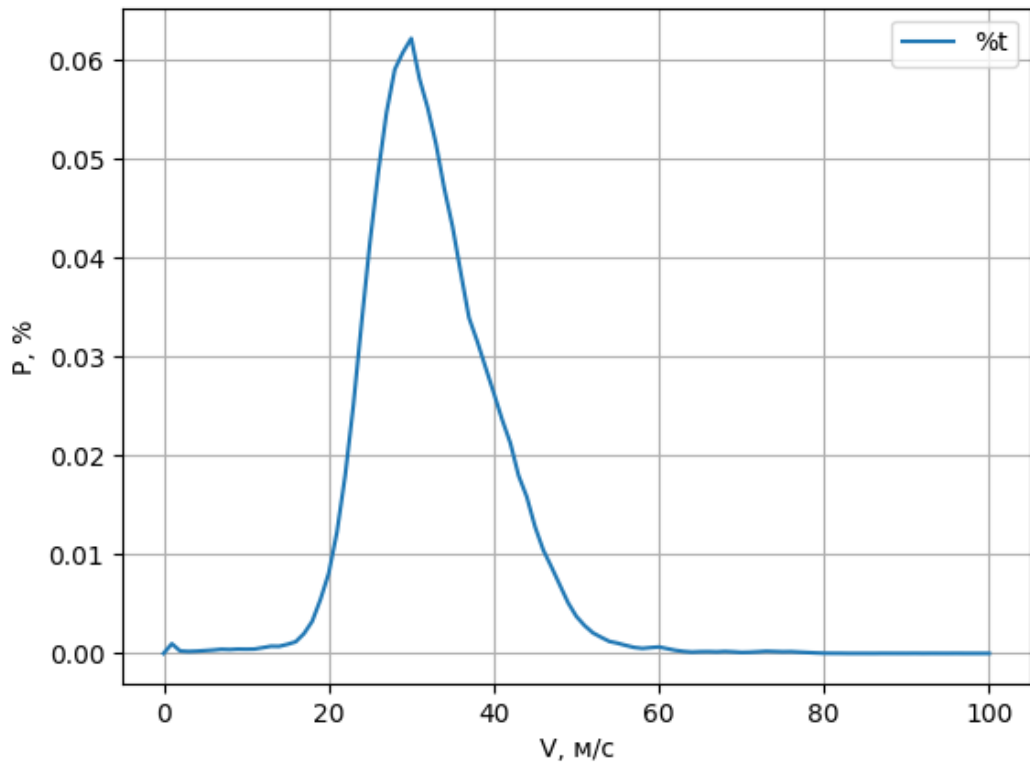


Рис. 3.4. Залежність $P(V_i)$.

Ця залежність є ключовою для моделювання критерію оптимізації, яке детально описане в наступному розділі.

Висновки по розділу

В процесі роботи над даним розділом магістерської дисертації було знайдено джерело відкритих даних треків польотів планерів – ресурс weglide.com. З цього ресурсу за допомогою API-запитів було завантажено масив файлів. Вибірка файлів була наступною:

- тип судна: SZD-48-3 Jantar Standard 3;
- висока досвідченість пілотів
- довжина маршруту не менше 200 км.

В результаті було отримано 84 файли.

За результатами обробки масиву інформації, були зроблені висновки що відношення часу на переході до часу в потоках становить 69:31. Також були отримано дані щодо розподілу швидкостей протягом маршруту.

4. РОЗРОБКА ОПТИМІЗАЦІЙНОЇ МОДЕЛІ НА ОСНОВІ ЗІБРАНИХ ДАНИХ.

4.1. Розрахунок аеродинамічних характеристик планера методом LLT

Для розрахунку аеродинамічних характеристик крила планера в цій роботі використана математична модель, відома під назвою "Теорія несучої лінії" (Lifting Line Theory, далі LLT [7, 8]). LLT модель схематизує крило П-подібним вихром змінної по розмаху циркуляції, кінцеві вихрові джгути простягаються в нескінченність назад по потоку, та неперервною вихровою пеленою, що заповняє простір між П-подібним контуром, з перемінною циркуляцією (відповідно зміні циркуляції вихрового джгута в прилягаючому перерізі, щоб задовольнити умову накладену теоремою Гельмгольца). Швидкості індуковані цією системою вихорів розраховуються інтегруванням формули Біо-Савара Лапласа (див. 5, 6).

$$\vec{V} = \int_{-\infty}^{\infty} \frac{\Gamma}{4\pi} \frac{d\vec{l} \times \vec{r}}{|\vec{r}|^3} \quad (5)$$

$$\vec{V} = \frac{\Gamma}{4\pi} \int_{-\infty}^{\infty} \frac{\sin \theta}{r^2} dl = \frac{\Gamma}{2\pi h} \quad (6)$$

Розв'язок інтегрально-диференційного рівняння, що пов'язує індуковані вихровою системою швидкості з підйнятною силою перерізу крила (вираз для теореми Жуковського) дозволяє знайти розподіл циркуляції по вихровому джгуті, який задовольняє рівняння Лапласа та граничні умови. Відповідні формули наведені нижче.

Основне інтегрально диференціальне рівняння теорії несучої лінії:

$$\alpha(y_0) = \frac{\Gamma(y_0)}{\pi V_{\infty} c(y_0)} + \alpha_{L=0}(y_0) + \frac{1}{4\pi V_{\infty}} \int_{-b/2}^{b/2} \frac{(d\Gamma/dy)dy}{y_0 - y} \quad (7)$$

Основне інтегрально диференціальне рівняння теорії несучої лінії в представленні через синусні ряди Фур'є:

$$\alpha(\theta_0) = \frac{2b}{\pi c(\theta_0)} \sum_1^N A_n \sin n\theta_0 + \alpha_{L=0}(\theta_0) + \sum_1^N nA_n \frac{\sin n\theta_0}{\sin \theta_0} \quad (8)$$

З вказаного вище очевидно, що LLT оперує нев'язкою нестисненою рідиною, таким чином явища в'язкого тертя і відриву потоку не моделюються. Для врахування явища в'язкого тертя використано стороннє програмне забезпечення XFOIL [9] що розраховує аеродинамічні характеристики 2D профіля крила для заданих умов обтікання (числа Маха, Рейнольдса та кут атаки/коефіцієнт підйімальної сили) які отримані на етапі роботи алгоритму LLT. Отримані значення опору тертя, індуктивного опору та підйімальної сили дозволяють оцінити аеродинамічні характеристики крила з достатньою точністю (на ділянці безвідривного обтікання). Перехід до характеристик апарату можна провести додавкою коефіцієнту опору фюзеляжу, втрат на інтерференцію і балансування достатньо простими виразами.

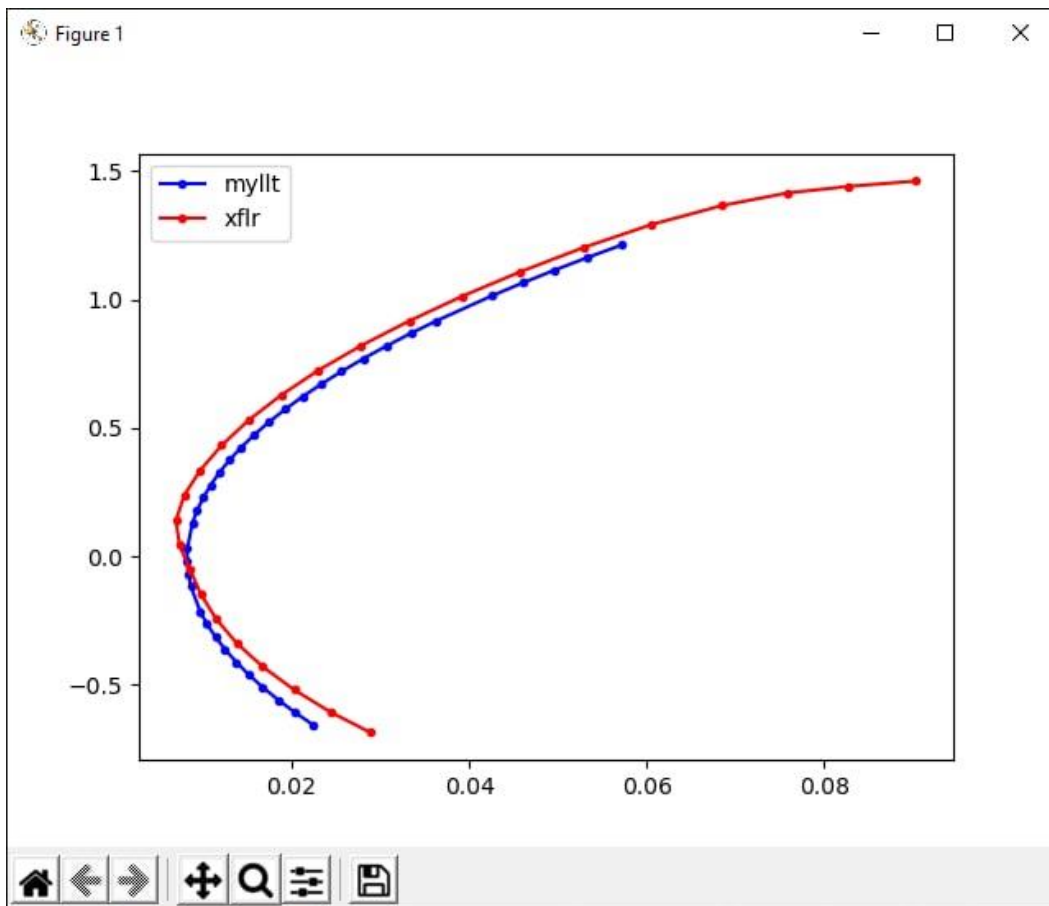


Рис. 4.1. Перевірка написаної програми.

Для вирішення задач, поставлених перед даною дисертацією, було створено комп'ютерну програму, написану мовою Python, яка реалізує метод LLT. Результати роботи програми було перевірено з еталоном. В даному випадку

еталоном виступає розрахунок, проведений в програмі Xflr5. Перевірка розрахунків програми Xflr5 була проведена в рамках бакалаврського дипломного проекту [1].

Алгоритм розрахунку АХ планера наведений на рис. 4.2.

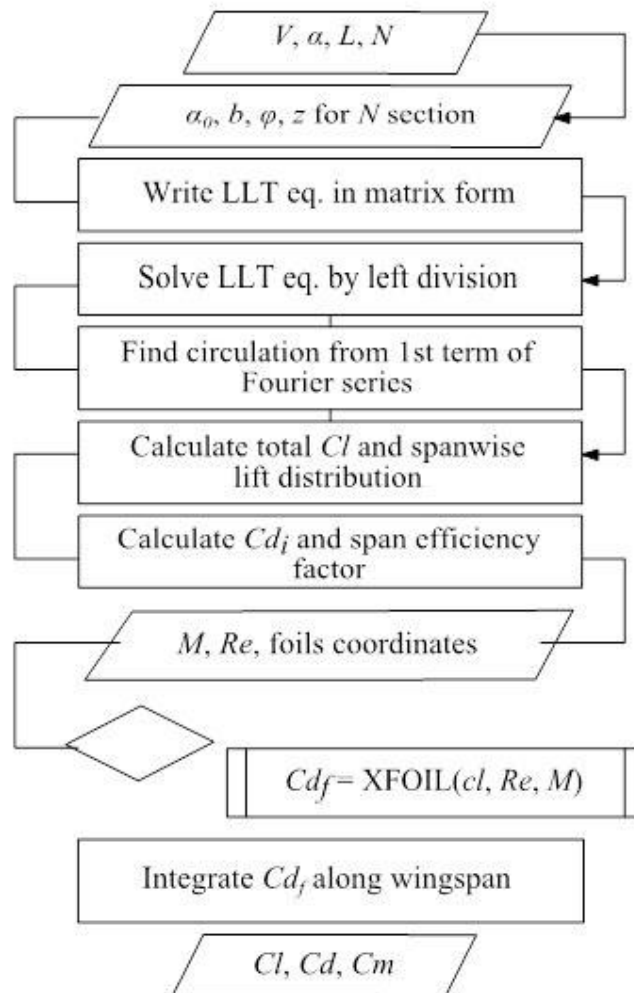


Рис. 4.2. Алгоритм розрахунку АХ планера

4.2. Генератор геометрії крила.

Для роботи алгоритму оптимізації необхідно створити функцію що параметризує створення геометрії крила. Така ф-я була написана мовою програмування Python. Результати її роботи наведені на рис. 4.2 – 4.3.

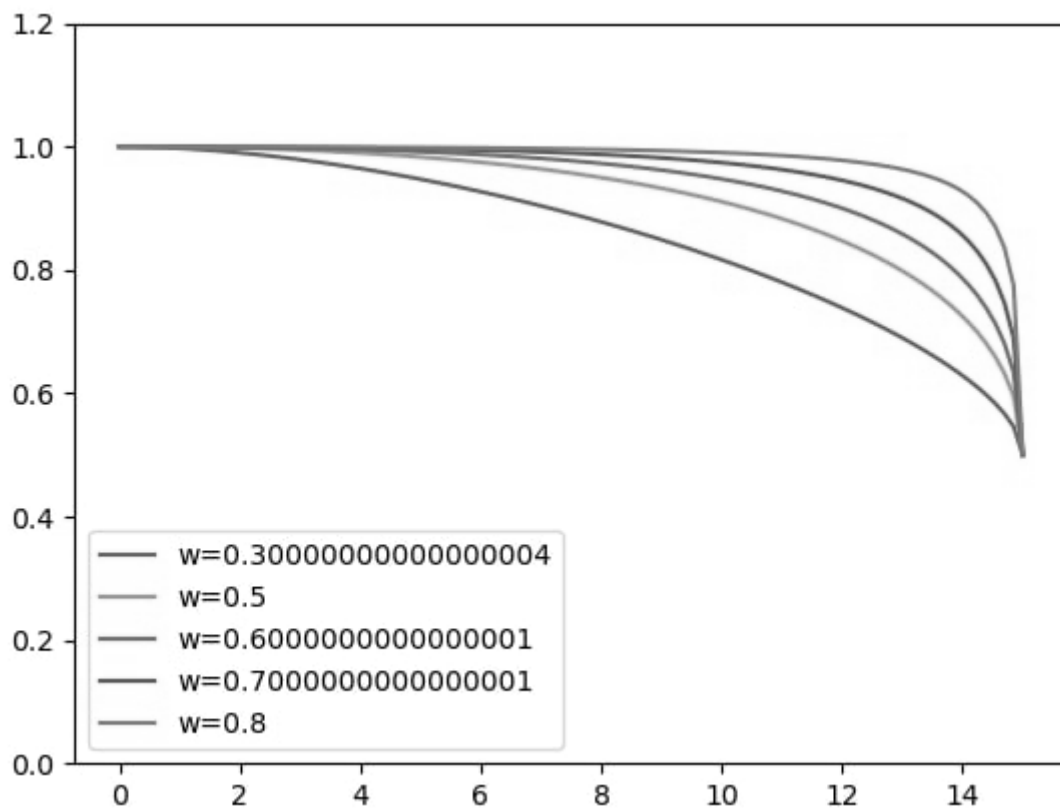


Рис. 4.3. Генерація сплайну – передньої кромки крила.

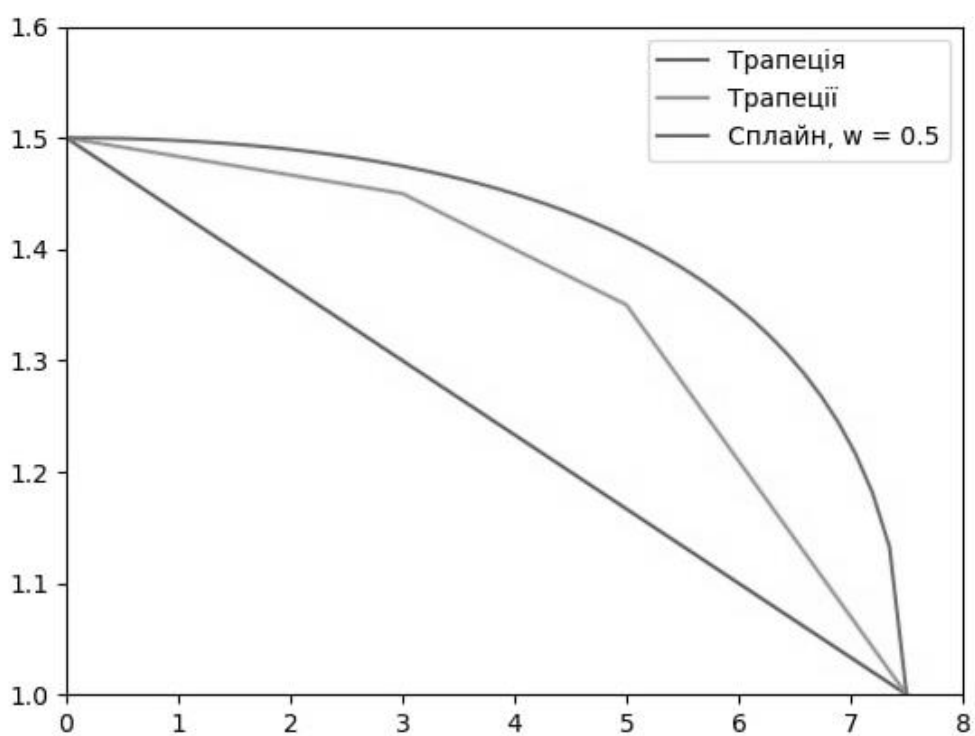


Рис. 4.4. Генерація різних типів крил (трапеція, декілька трапецій, сплайн).

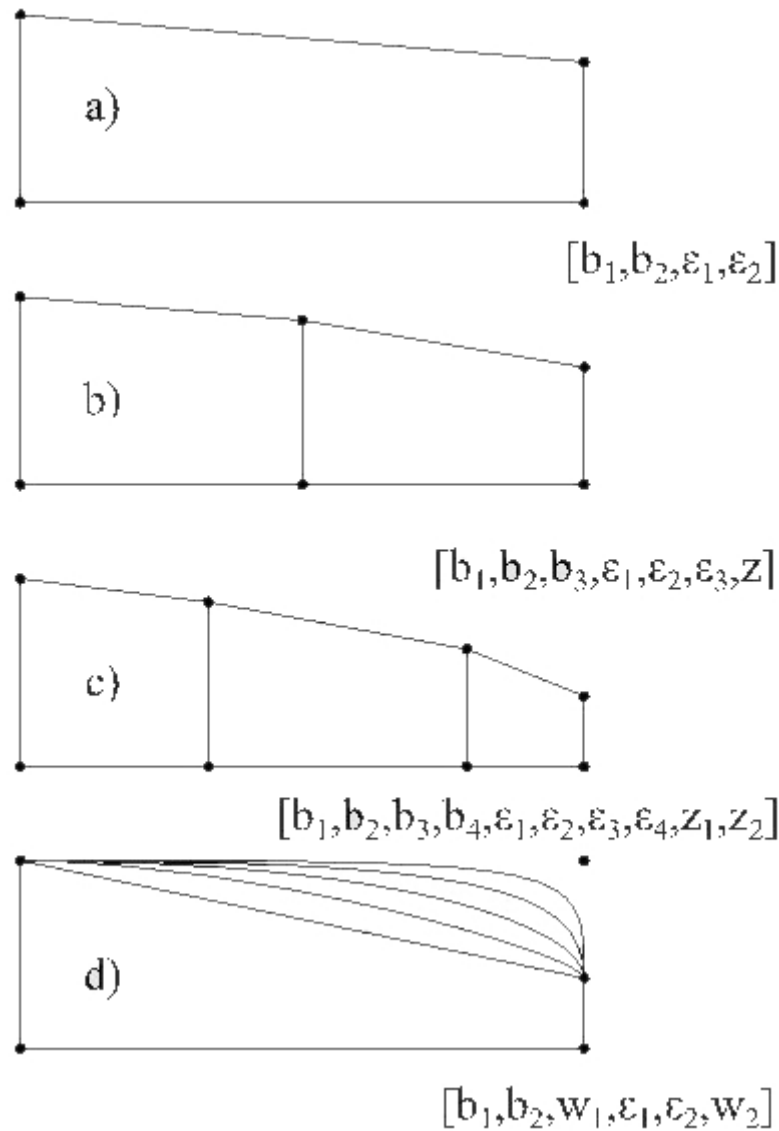


Рис. 4.5. Параметризація крила.

4.3. Генетичний алгоритм

Генетичний алгоритм (англ. genetic algorithm) - це евристичний алгоритм пошуку, що використовується для вирішення задач оптимізації та моделювання шляхом випадкового підбору, комбінування та варіації шуканих параметрів з використанням механізмів, аналогічних природному відбору в природі [10, 12]. Є різновидом еволюційних обчислень, за допомогою яких вирішуються оптимізаційні завдання з використанням методів природної еволюції, таких як успадкування, мутації, відбір і кросингвер. Відмінною особливістю генетичного алгоритму є акцент на використання оператора схрещування, який проводить

операцію рекомбінації рішень-кандидатів, роль якої аналогічна ролі схрещування в живій природі.

Завдання формалізується таким чином, щоб її рішення могло бути закодоване у вигляді вектора (генотипу) генів, де кожен ген може бути бітом, числом або якимось іншим об'єктом. У класичних реалізаціях генетичного алгоритму (ГА) передбачається, що генотип має фіксовану довжину. Однак є варіації ГА, вільні від цього обмеження.

Деяким, зазвичай випадковим чином, створюється безліч генотипів початкової популяції. Вони оцінюються з використанням «функції пристосованості», в результаті чого з кожним генотипом асоціюється певне значення («пристосованість»), яке визначає наскільки добре фенотип, який він описує, вирішує поставлене завдання.

При виборі функції пристосованості (або *fitness function* в англійській літературі) важливо стежити, щоб її рельєф був гладким.

З отриманої множини рішень («покоління») з урахуванням значення «пристосованості» вибираються рішення (зазвичай найкращі особини мають більшу ймовірність бути обраними), до яких застосовуються «генетичні оператори» (у більшості випадків «схрещування» - *crossover* і «мутація» - *mutation*), результатом чого є отримання нових рішень. Їх також обчислюється значення пристосованості, і потім проводиться відбір («селекція») кращих рішень у наступне покоління.

Цей набір дій повторюється ітеративно, так моделюється «еволюційний процес», що триває кілька життєвих циклів (поколінь), доки буде виконано критерій зупинення алгоритму. Таким критерієм може бути:

- перебування глобального, чи субоптимального рішення;
- вичерпання числа поколінь, відпущених на еволюцію;
- вичерпання часу, відпущеного на еволюцію.

Генетичні алгоритми служать, головним чином, для пошуку рішень у багатовимірних просторах пошуку.

Таким чином, можна виділити такі етапи генетичного алгоритму:

1. Задати цільову функцію (пристосованості) для особин популяції

2. Створити початкову популяцію

- (Початок циклу)

Розмноження (схрещування)

1. Мутація
2. Обчислити значення цільової функції всім особин
3. Формування нового покоління (селекція)
4. Якщо виконуються умови зупинки, то (кінець циклу), інакше (початок циклу).

4.4. Критерій оптимальності.

З огляду на проведені дослідження та зібрані дані, було вирішено визначити узагальнений критерій, який би враховував вплив атмосфери, фази польоту та аеродинамічні характеристики конкретного планера. В результаті було розглянуто залежність 4, зображену на рис. 2, вісь швидкостей якої було перераховано в коефіцієнт підйомної сили за наступною формулою:

$$C_{y_i} = \frac{2 \cdot mg}{\rho \cdot v_i^2 \cdot S} \quad (10),$$

де m – маса планера, S – площа крила планера, v_i – швидкість польоту.

В результаті отримуємо залежність

$$P_{C_y} = f(C_y) \quad (11),$$

де p_{C_y} – відсоток часу, який провів планер при заданому C_y (див. рис. 4.6. а).

Площу фігури під графіком цієї залежності, відносно осі швидкості, можна знайти за формулою (12). Аналогічно, фігуру під графіком поляри крила, відносно осі C_y можна знайти за формулою (13). Схематичне зображення шуканої величини зображено на рис. 4.6, б.

$$S_{Pi} = \int_{C_{yv_{min}}}^{C_{yv_{max}}} P(C_{y_i}) \cdot dC_y \quad (12)$$

$$S_{Ci} = \int_{C_{yv_{min}}}^{C_{yv_{max}}} C_x \cdot dC_y \quad (13)$$

З огляду на це було запропоновано обчислювати критерій, як площу під графіком поляри крила, приведеної до відсоткового розподілу, отриманого зі статистичних даних. Таким чином, шукана величина обчислюється за наступною формулою:

$$f(W_i) = \int_{C_{yVmin}}^{C_{yVmax}} p_{C_{y_i}} \cdot C_{y_i} \cdot dC_y \quad (14)$$

$$W_i = (b_{1_i}, b_{2_i}, \omega_{1_i}, e_{1_i}, e_{2_i}).$$

де W_i – вектор параметрів крила планера, які більш детально розглядались вище по тексту.

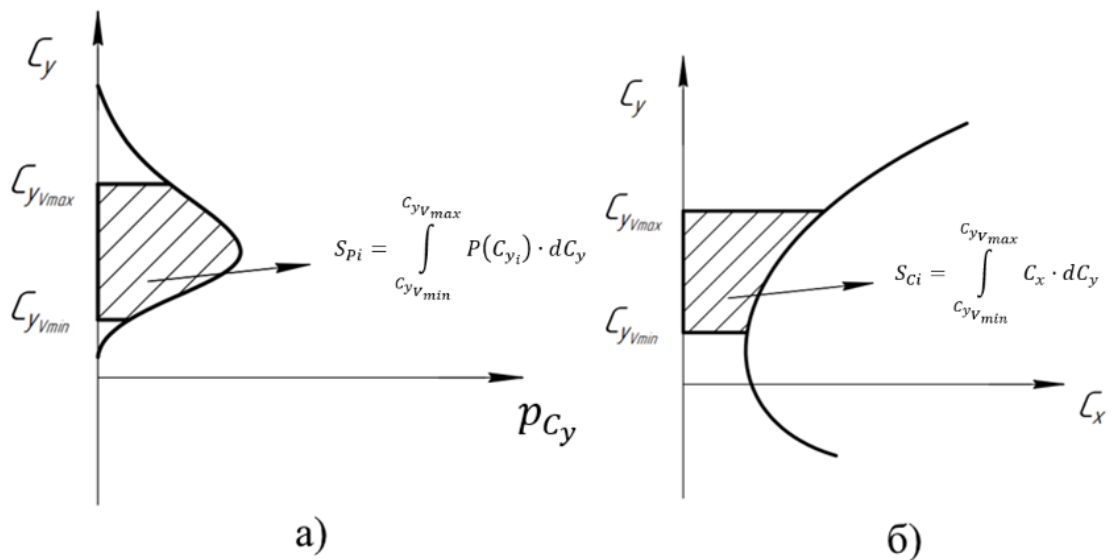


Рис. 4.6. Виведення критерію оптимізації:

a) – відсотковий розподіл коефіцієнта підйомної сили по часу; б) – площа під графіком поляри крила.

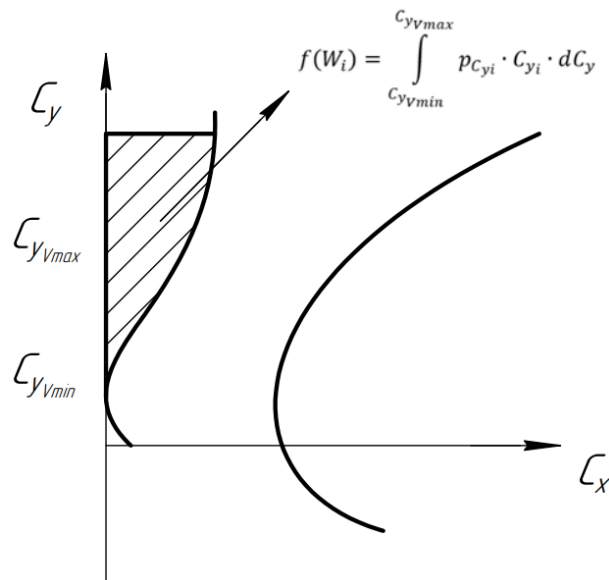


Рис. 4.7. Графічне зображення критерію оптимізації

В результаті, при порівнянні двох крил, менше значення виразу 14 для одного з них буде свідчити про більш високі аеродинамічні характеристики даного крила. Іншими словами, чим менша величина розрахованого критерію для конкретного крила, тим більшу середню швидкість він покаже при проходженні маршруту. Дану гіпотезу буде перевірено в наступних розділах даної дисертації.

Величина, розрахована за формулою 14 для автоматично згенерованого крила планера зображена на рис. 4.8 поруч з полярою цього крила, розрахованої за Теорією несучої лінії, програмну реалізацію якої було здійснено в рамках даної магістерської дисертації.

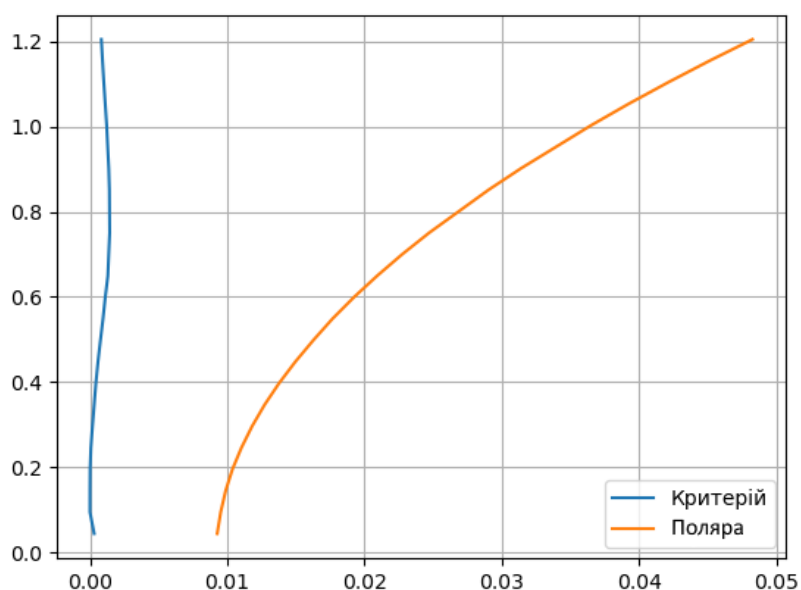


Рис. 4.8. Поляра крила та критерій оптимальності порашовані для згенерованого крила.

Висновки по розділу

В цьому розділі був проведений синтез оптимізаційної моделі на основі зібраних в попередньому розділі даних. Даний процес був розділений на чотири етапи:

1. розробка модуля розрахунку аеродинамічних характеристик планера методом LLT;
2. розробка програми-генератора геометрії крила, якій передувало дослідження можливих варіантів параметризації геометричних характеристик крила;
3. реалізація генетичного алгоритму оптимізації;
4. синтез критерію оптимізації.

Кожен з елементів, розглянутих в даному розділі був реалізований у вигляді програм чи окремих функцій, написаних мовою програмування Python 3 з використанням спеціалізованих наукових бібліотек SciPy та NumPy. Програмний код наведений у додатках.

5. РОЗРОБКА ГЕОМЕТРІЇ КРИЛА ПЛАНЕРА НА ОСНОВІ ОПТИМІЗАЦІЙНОЇ МОДЕЛІ.

В даному розділі будуть розглядатися результати роботи розробленого алгоритму і відповідної програми, на основі яких буде створено геометрію планера.

5.1. Результати роботи генетичного алгоритму.

Генетичний алгоритм, реалізований в рамках даної роботи має велику кількість налаштувань, таких як величина популяція, ймовірності виклику одного з «генетичних операторів», вибірка особин, до якої будуть застосовуватись оператори і т.д. Крім того, робота алгоритму займає досить багато часу та обчислювальних ресурсів комп'ютера. В сумі ці фактори призводять до накладення деяких обмежень на виконання програми, які певним чином впливають на результат. Так, на перших запусках величина початкової популяції, яка визначалася випадковим чином, обмежувалась п'ятьма елементами, з яких тільки три приймали участь у процесі «еволюції», а кількість ітерацій алгоритму обмежується величиною в 47.

В процесі роботи програма записує у лог-файл обчислені дані а також операції генетичного алгоритму, які вона виконала. Скріншот цього файлу зображений на рис. 5.1.

1	start logging			
2	breed_random			
3	breed_random			
4	breed_random			
5	breed_random			
6	breed_random			
7	breed_random			
8	breed_random			
9	breed_random			
10	population after populate():			
11	name: 0, crit: 0.0002510766840308263, params: [0.58, 0.35, 0.79, -3.0, 1.7]			
12	name: 0, crit: 0.0004206684969046473, params: [1.4, 0.69, 0.52, 1.4, 0.91]			
13	name: 0, crit: 0.0004523694859078406, params: [1.32, 1.01, 0.12, -1.82, -0.52]			
14	name: 0, crit: 0.00016046099164646537, params: [0.59, 0.31, 0.21, -0.76, -2.47]			
15	name: 0, crit: 0.0003037821530720207, params: [1.46, 0.6, 0.51, -0.63, -2.28]			
16	name: 0, crit: 0.0004766361140248574, params: [1.32, 0.93, 0.18, -0.29, -1.35]			
17	name: 0, crit: 0.0005200471456298202, params: [1.19, 0.54, 0.82, 1.37, -3.33]			
18	name: 0, crit: 0.0005321076152597319, params: [1.41, 0.37, 0.24, 1.91, -4.43]			
19				
20	breed_random			
21	breed_random			
22	breed_crossover			
23	breed_crossovered			
24	breed_interpolate			
25	breed_random			
26	breed_random			
27	breed_random			
28	breed_random			
29	breed_random			
30	breed_random			
31	==== iteration: 1			
32	population after evolve():			
33	name: 0, crit: 0.0005321076152597319, params: [1.41, 0.37, 0.24, 1.91, -4.43]			
34	name: 0, crit: 0.0005200471456298202, params: [1.19, 0.54, 0.82, 1.37, -3.33]			
35	name: 0, crit: 0.0004766361140248574, params: [1.32, 0.93, 0.18, -0.29, -1.35]			
36	name: 12, crit: 0.00043731737535274737, params: [1.13, 0.53, 0.87, -3.85, 1.38]			
37	name: 12, crit: 0.0003805479464406084, params: [1.36, 0.33, 0.68, -1.19, -3.89]			
38	name: 12, crit: 0.0004649701802273668, params: [1.37, 0.89, 0.09, -3.76, -0.88]			
39	name: 12, crit: 0.00035045311447273026, params: [1.4, 0.43, 0.8, -4.21, 1.22]			
40	name: 12, crit: 0.0005102491202306774, params: [1.46, 0.51, 0.15, -0.42, 1.81]			
41				
42	breed_random			
43	breed_random			
44	breed_interpolate			
45	breed_interpolate			
46	breed_random			
47	breed_crossover			
48	breed_random			
49	breed_random			
50	breed_random			
51	breed_random			
52	breed_random			
53	==== iteration: 2			
54	population after evolve():			
55	name: 0, crit: 0.0005321076152597319, params: [1.41, 0.37, 0.24, 1.91, -4.43]			
56	name: 0, crit: 0.0005200471456298202, params: [1.19, 0.54, 0.82, 1.37, -3.33]			
57	name: 23, crit: 0.0006888640471295661, params: [1.05, 0.48, 0.58, -0.17, -3.57]			

	b	z	e	
0	0,4000000	-7,5	-1,8	
1	0,4573014	-7,386058	-1,588888888888889	
2	0,5217001	-7,047694	-1,3777777777777778	
3	0,5904213	-6,495190	-1,166666666666667	
4	0,6604589	-5,745333	-0,9555555555555558	
5	0,7285840	-4,820907	-0,7444444444444446	
6	0,7912991	-3,749999	-0,5333333333333334	
7	0,8446525	-2,565151	-0,3222222222222223	
8	0,8836422	-1,302361	-0,11111111111111113	
9	0,9000000	0,0	0,1	
10	0,9000000	0,0	0,1	
11	0,8836422	1,3023613	-0,11111111111111113	
12	0,8446525	2,5651510	-0,3222222222222223	
13	0,7912991	3,7499999	-0,5333333333333334	
14	0,7285840	4,8209070	-0,7444444444444446	
15	0,6604589	5,7453333	-0,9555555555555558	
16	0,5904213	6,4951905	-1,166666666666667	
17	0,5217001	7,0476946	-1,3777777777777778	
18	0,4573014	7,3860581	-1,588888888888889	
19	0,4000000	7,5	-1,8	

Рис. 5.1. Лог-файли програми.

У файл записується наступна інформація:

- Номер ітерації алгоритму
- Поточна популяція (номер крила, обчислений критерій та параметри генератора)
- Список операцій, здійснених над популяцією.

Після закінчення роботи алгоритму, що в залежності від налаштувань займає досить багато часу (1.5 -4 години в рамках даного дослідження), з лог-файлу збирається інформація про кожну популяцію в один csv файл для зручності подальшої обробки.

На рисунку нижче зображений графік зміни критерію оптимальності. Як видно з нього, величина критерію має тенденцію сходиться в на певному значенні, що є позитивним результатом.

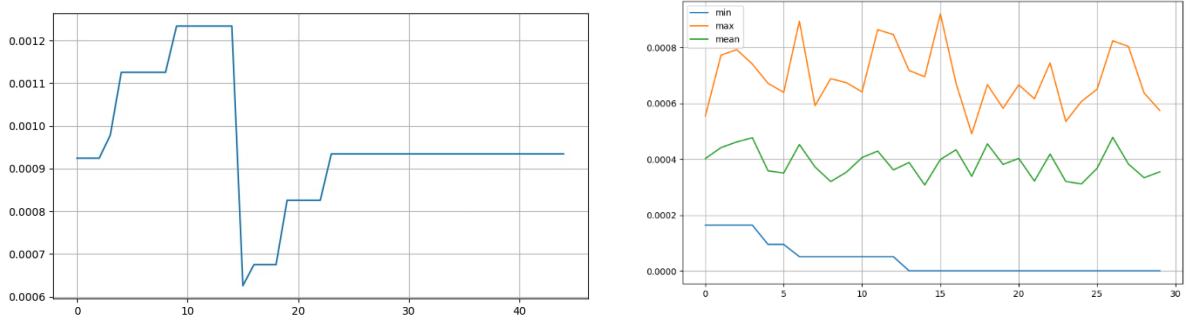


Рис. 5.2. Графіки критерію оптимальності на кожній з ітерацій програми..

Отримані дані дозволяють перейти до більш детальної розробки зовнішнього вигляду планера. Отже, отримано наступне:

Таблиця 1.

Синтезовані геометричні параметри

Розмах крила, м	15
Коренева хорда, м	0,93
Кінцева хорда, м	0,55
Кут крутки в корені, градус	0
Кут крутки на закінцівці	-1,3
Профіль	FX S 02-196

5.2. Створення просторової моделі крила.

Для подальшого дослідження аеродинамічних характеристик синтезованого крила необхідно створити більш детальну просторову модель, яка може слугувати вхідними даними для програм розрахунку АХ. В даному випадку, такою програмою слугуватиме Xflr5.

На рис. 5.3. зображений згенерований програмою графік, на якому зображений розподіл хорд вздовж крила.

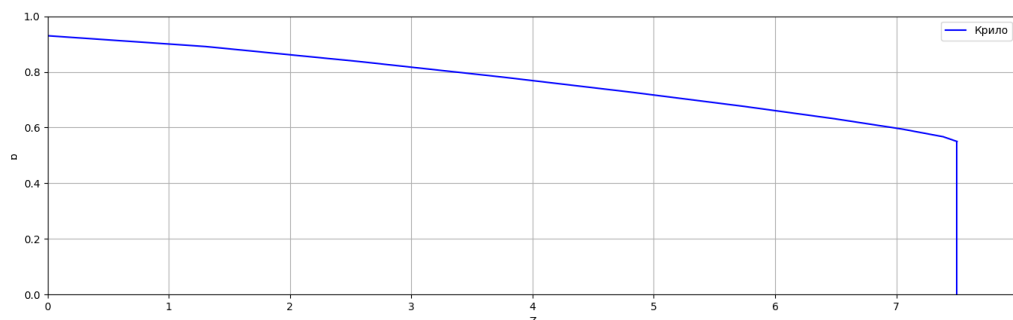


Рис. 5.3. Генерація оптимального крила.

Більш детальне схематичне креслення отриманого крила зображене на рис. 5.4.

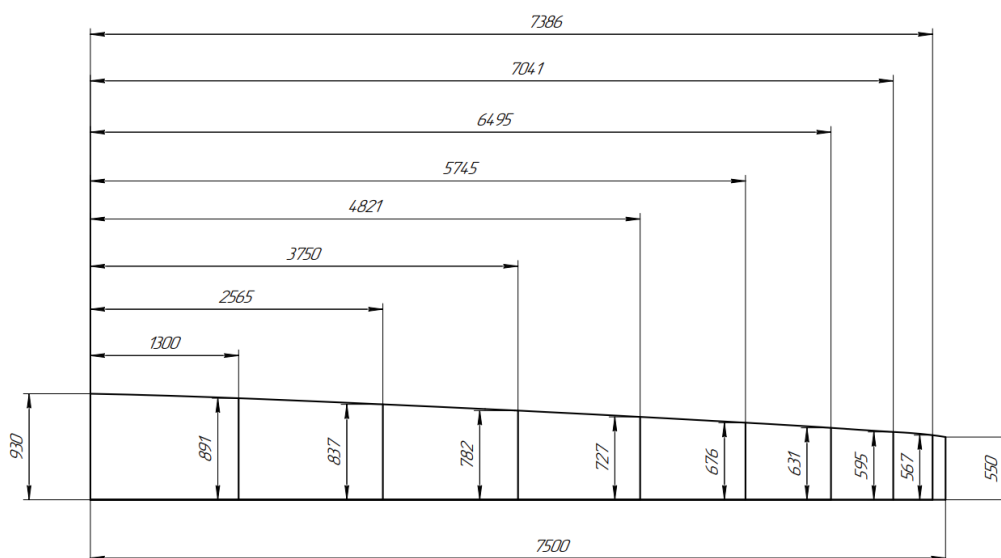


Рис. 5.4. Схематичне зображення крила

Маючи геометричні параметри крила, у програмі аеродинамічного аналізу Xflr5 було створено просторову модель синтезованого планерного крила. Скріншот програми зображений на рис. 5.5. Даний крок дасть змогу проаналізувати аеродинамічні характеристики крила в наступному розділі.

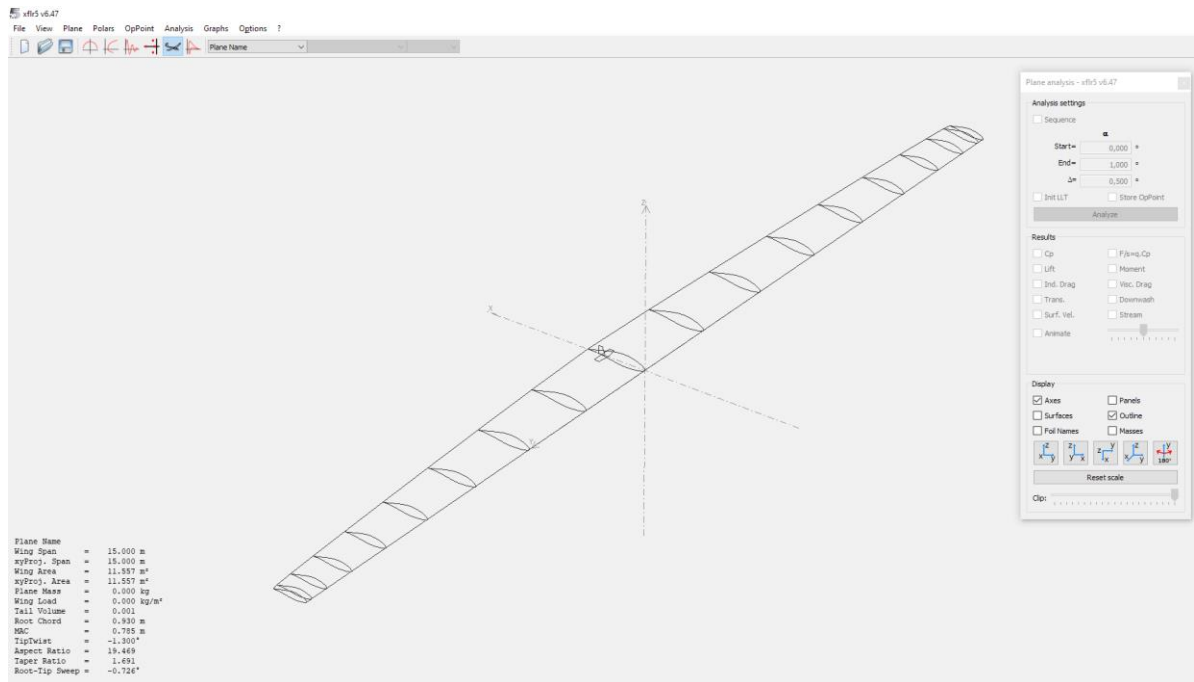


Рис. 5.5. Просторова модель планера в програмі Xflr-5.

Висновки по розділу

В даному розділі була побудована геометрія крила планера на основі даних, згенерованих програмою оптимізатором, математичну модель якої було описано в попередньому розділі. Були отримані параметри крила, за якими була побудована просторова модель в програмі Xflr5 для подальшого розрахунку аеродинамічних характеристик.

6. ВИЗНАЧЕННЯ АЕРОДИНАМІЧНИХ ХАРАКТЕРИСТИК ЗАПРОПОНОВАНОГО ПЛАНЕРА.

Дослідження аеродинамічних характеристик синтезованого планера були здійснені методом LLT в програмному забезпеченні Xflr-5.

Вхідними даними для роботи програми є:

1. просторова модель, описана в попередньому розділі;
2. швидкість потоку $V_{\infty} = 25$ м/с;
3. густина повітря $\rho = 1,225$ кг/м³;
4. кінематична в'язкість $\nu = 1,46 \cdot 10^{-5}$ м²/с;
5. діапазон кутів атаки $\alpha = -13^{\circ} \dots +12,5^{\circ}$ з кроком $0,5^{\circ}$.

Параметри середовища вибиралися відповідно до міжнародної стандартної атмосфери на рівні моря, а швидкість обрана довільно з діапазону дозволених керівництвом з льотної експлуатації.

Результатами роботи програми є залежності $C_{ya}(\alpha)$, $C_{xai}(\alpha)$, $C_{xaf}(\alpha)$, $m_{za}(\alpha)$.

Метод дослідження АХ за теорією несучої лінії не передбачує аналіз обтікання фюзеляжу ЛА, тому виникає необхідність розрахувати опір, який він створює аналітично.

З огляду на те, що аеродинамічна підйомна сила, створювана фюзеляжем є досить малою в порівнянні з підйомною силою крила, то в даній роботі вона не враховується, натомість враховується тільки аеродинамічну силу опору фюзеляжу. Як відомо, повний аеродинамічний опір фюзеляжу складається з опору тиску, опору тертя та хвильового опору:

$$X_{\text{фюз}} = X_f + X_{\text{хв}} + X_p, \text{ де}$$

X_f – опір тертя;

$X_{\text{хв}}$ – хвильовий опір;

X_p – опір тиску.

Запишемо відповідні їм аеродинамічні коефіцієнти:

$$C_x^{fus} = C_{xf} + C_{xв} + C_{xp}.$$

Оскільки планер здійснює політ при швидкостях в межах від приблизно 50 км/год до максимум 250, то хвильовим аеродинамічним опором можна знехтувати.

У такому разі отримаємо наступну залежність:

$$C_x^{fus} = C_{xf} + C_{xp} \quad (15).$$

Вираз для обчислення опору тертя фюзеляжу знайдений виходячи з методу заміни поверхні фюзеляжу на еквівалентну за площею пластину, який описаний в § 10.5 []. З огляду на те, що в даній роботі за референсну точку був обраний планер SZD-48-3, то припустимо що величини площ фюзеляжів цих планерів збігаються, а отже

$$S_{fus} = 9,98 \text{ м}^2.$$

Значення числа Рейнольдса для даного фюзеляжу

$$Re_{fus} = \frac{V \cdot l_{fus}}{\nu} = 1,17294 \cdot 10^7.$$

Згідно з §4,4 [], опір ламінарного примежового шару

$$C_{xf}^{lam} = \frac{1,3}{\sqrt{Re_{fus}}} = 3,796 \cdot 10^{-4},$$

опір ламінарного примежового шару

$$C_{xf}^{turb} = \frac{0,072}{\sqrt[5]{Re_{fus}}} = 2,776 \cdot 10^{-3}$$

З огляду на те що якість поверхні сучасних планерів є дуже висока і на поверхні відсутні турбулізатори, то виберемо відношення площі турбулентного примежового шару до ламінарного на рівні 1:1. Тобто, площі ламінарного і турбулентного примежових шарів будуть однаковими:

$$S_{lam} = S_{turb} = 4,99\text{м}^2$$

Тоді, сила опору тертя буде рівна

$$Q_f = C_{xf}^{lam} \cdot q \cdot S_{lam} + C_{xf}^{turb} \cdot q \cdot S_{turb} \quad (16),$$

де q – це швидкісний напір.

Коефіцієнт опору тертя, приведений до площі крила планера можна знайти за наступною формулою:

$$C_{xf} = \frac{Q_f}{q \cdot S} = 1,362 \cdot 10^{-3} \quad (17),$$

де S – площа крила.

Опір тиску фюзеляжу можна оцінити за допомогою дослідження, описаного в розділі 6-9 [11]. Згідно з ним, відношення опору тиску до загального опору для обтічного тіла при повністю турбулентному обтіканні можна знайти за наступною формулою:

$$\frac{C_{xp}^{turb}}{C_x^{turb}} = \frac{21}{\lambda_{fus}^2} \quad (18),$$

де C_x^{turb} – коефіцієнт повного опору для повністю турбулентного обтікання.

$$C_x^{turb} = C_{xf}^{turb} + C_{xp}^{turb} \quad (19).$$

Оскільки для безвідриного обтікання точка переходу з ламінарного в турбулентний примежовий шар не впливає на епюру тиску, то справедливе наступне співвідношення:

$$\frac{C_{xp}^{turb}}{C_x^{turb}} = \frac{C_{xp}}{C_x^{fus}} = \frac{21}{\lambda_{fus}^2} \quad (20).$$

Підставивши (15) в (19) отримаємо:

$$C_{xp} = \frac{\frac{21}{\lambda_{fus}^2} \cdot C_{xf}}{\lambda_{fus} - \frac{21}{\lambda_{fus}}} = 4,75 \cdot 10^{-4} \quad (21).$$

В результаті, даний опір додається до порахованого програмою Xflr5 тим самим враховується опір фюзеляжу.

Результати продувки просторової моделі крила в Xflr5 наведені на рис.

6.1.

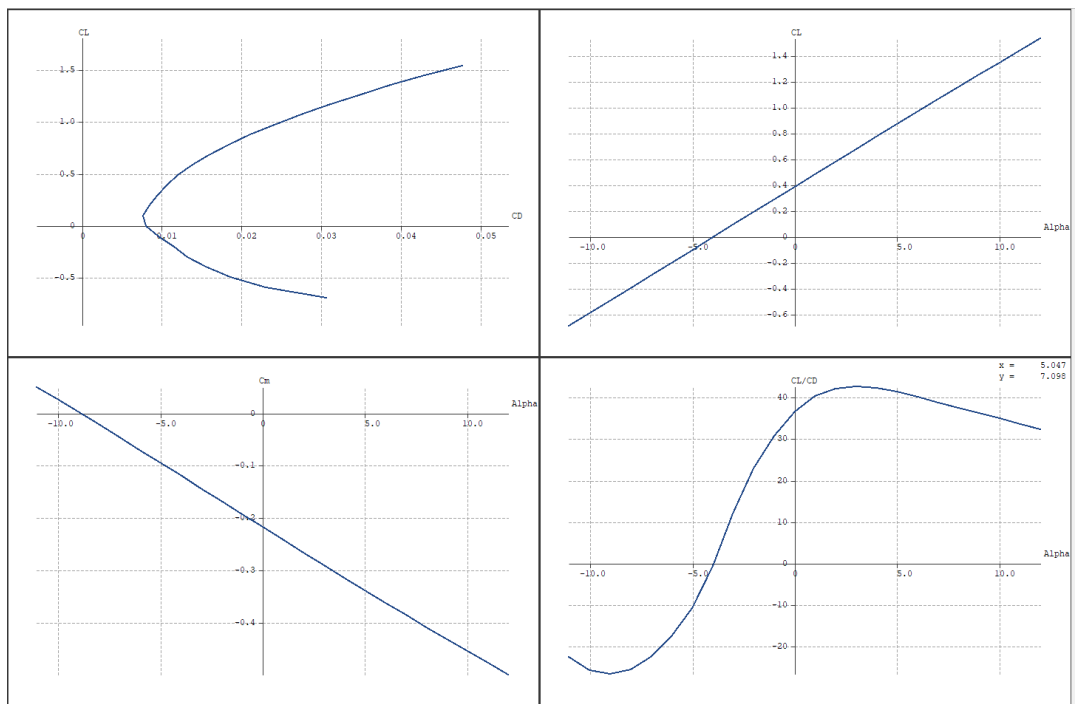


Рис. 6.1. Графіки основних аеродинамічних характеристик планера.

Після цього, отримані характеристики були перераховані з врахуванням аеродинамічного опору фюзеляжу. Перерахована поляра планера зображена на рис.

6.2.

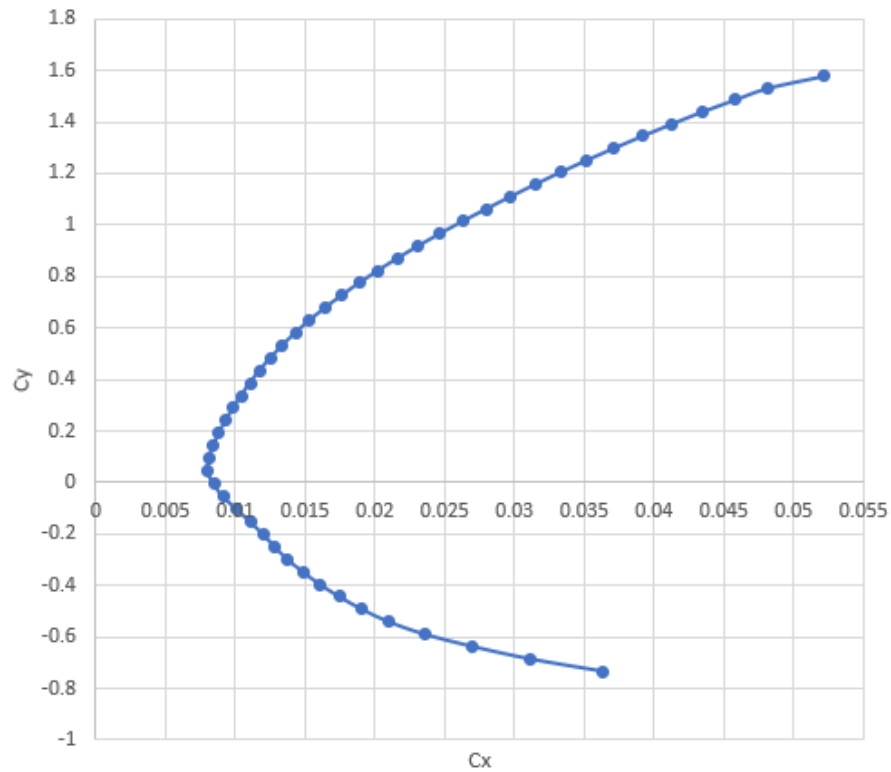


Рис. 6.2 Поляра синтезованого планера

Наступним кроком є оцінка льотних якостей планера по його полярі швидкостей. Маючи залежності $C_y(C_x)$, $K(C_y)$ поляра швидкостей обчислюється за наступними співвідношеннями:

$$\vartheta_{x_i} = \sqrt{\frac{2 \cdot m \cdot g}{C_{y_i} \cdot \rho \cdot S}} \quad (22),$$

де ϑ_{x_i} – горизонтальна швидкість планування, C_{y_i} - відповідний їй коефіцієнт підйомної сили, S – площа крила, m – маса планера, ρ – густина повітря, g – прискорення вільного падіння.

$$\vartheta_{y_i} = \frac{\vartheta_{x_i}}{K} \quad (23),$$

де ϑ_{y_i} – вертикальна швидкість планера, K – аеродинамічна якість.

Для кожної точки поляри планера були обчислені величини ϑ_{y_i} та ϑ_{x_i} , які занесені у таблицю 2. Графік поляри швидкостей наведений на рис. 6.3.

Дані поляри швидкостей

α	C_{y_i}	C_{D_i}	C_D	C_x	ϑ_x	K	ϑ_y
-3	0.092735	0.00016	0.007686	0.008161	69.75383	11.36319	-6.13858
-2	0.190543	0.000607	0.008358	0.008833	48.66238	21.57172	-2.25584
-1	0.288284	0.001366	0.009394	0.009869	39.56212	29.21106	-1.35435
0	0.385909	0.002435	0.010584	0.011059	34.1938	34.89547	-0.97989
1	0.483375	0.003812	0.01205	0.012525	30.55257	38.59281	-0.79166
2	0.580634	0.005495	0.013822	0.014297	27.87653	40.6123	-0.68641
3	0.677642	0.007478	0.015944	0.016419	25.80416	41.27182	-0.62522
4	0.774352	0.009758	0.018386	0.018861	24.13908	41.05572	-0.58796
5	0.870721	0.012328	0.021135	0.02161	22.76409	40.2925	-0.56497
6	0.966703	0.015182	0.02418	0.024655	21.60446	39.20921	-0.551
7	1.062255	0.018312	0.027481	0.027956	20.60988	37.99739	-0.5424
8	1.157333	0.021709	0.030965	0.03144	19.74516	36.81085	-0.5364
9	1.251893	0.025365	0.034686	0.035161	18.98481	35.60459	-0.53321
10	1.345893	0.029269	0.038684	0.039159	18.30984	34.36995	-0.53273
11	1.439292	0.03341	0.042937	0.043412	17.70579	33.15424	-0.53404
12	1.532049	0.037776	0.047692	0.048167	17.16143	31.80703	-0.53955

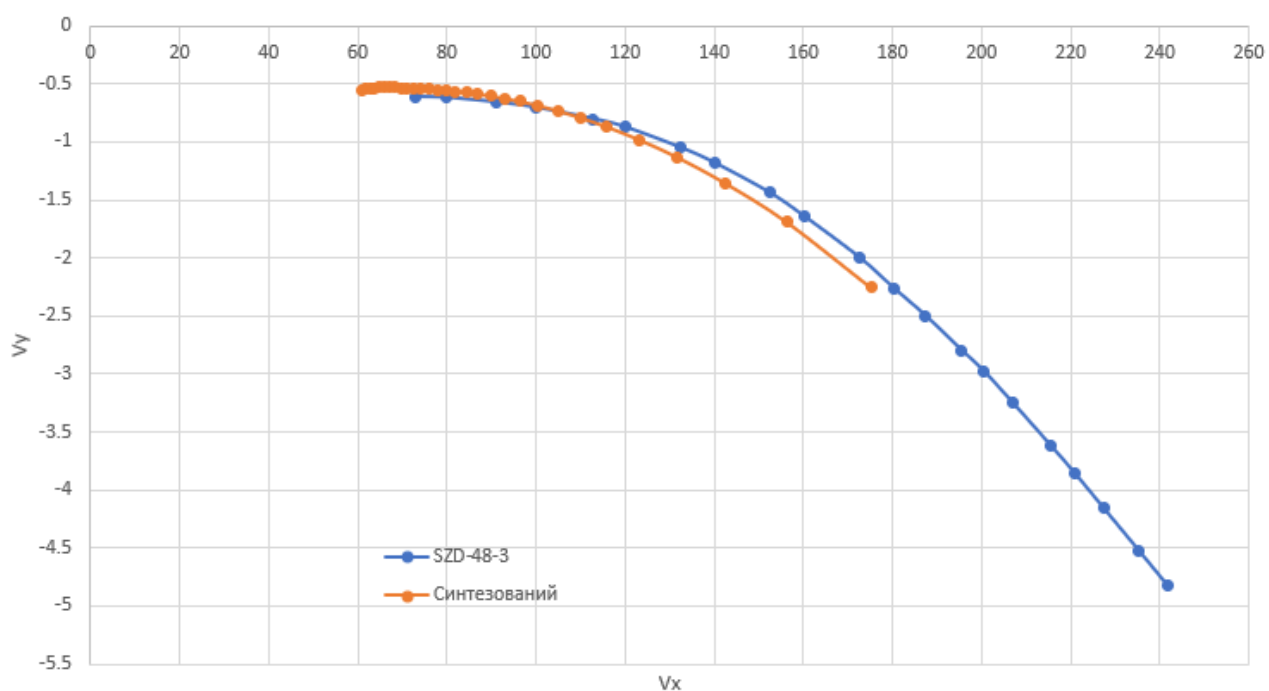


Рис. 6.3. Поляри швидкостей створеного планера та вихідного.

Обчислимо інші параметри планера. Швидкість звалювання:

$$\vartheta_s = \sqrt{\frac{2mg}{\rho S \cdot C_{y_{max}}}} = 61 \text{ км/год.}$$

Швидкість максимальної якості:

$$v_{K_{max}} = \sqrt{\frac{2mg}{\rho S \cdot C_{yK_{max}}}} = 92.87$$

Як видно з рис. 6.3 а також відповідно до даних таблиці 2, можна сказати що результуюча поляра має інший кут дотичної, проведеної з точки (0, 0). Це свідчить про те, що максимальна аеродинамічна якість цих планерів відрізняється. Дійсно, для згенерованого алгоритмом планера, значення $K_{max} = 41,27$, тоді як для планера SZD-48-3 $K_{max} = 40$. Також можна помітити що отриманий планер має меншу швидкість зниження на малих швидкостях (в діапазоні 60-100 км/год) і дещо більшу швидкість зниження на більших швидкостях (відрізок 100-180 км/год).

Висновки по розділу

В даному розділі були досліджені аеродинамічні характеристики планера, спроектованого на основі результатів роботи оптимізаційного алгоритму. Розрахунок проводився в програмі Xflr5 методом LLT. Були отримані залежності $C_{ya}(\alpha)$, $C_{xai}(\alpha)$, $C_{xaf}(\alpha)$, $m_{za}(\alpha)$, а також побудовані їх графіки. Основні величини:

$$v_s = 61 \text{ км/год,}$$

$$v_{K_{max}} = 92.87 \text{ при } m = 326 \text{ кг.}$$

7. ПОРІВНЯННЯ ОТРИМАНИХ АЕРОДИНАМІЧНИХ ХАРАКТЕРИСТИК З ВИХІДНИМИ.

Оскільки основною задачею даного дослідження є розробка методики оптимізації аеродинамічних характеристик планера, то є необхідність в оцінці результатів такої оптимізації. В цьому розділі буде запропонований та досліджений метод такої оцінки.

На першому етапі було обчислено в програмі Xflr5 AX крила планера SZD-48-3 на основі його геометричних характеристик. Дані результати були порівняні з AX синтезованого планера. Для порівняння були взяті дані результатів продувок програмним забезпеченням, а не експериментальні дані. Таким чином було виключені похибки, які дає чисельний метод в порівнянні з експериментом. Це дозволяє оцінити роботу алгоритму оптимізації не беручи до уваги розбіжності в результатах LLT та даних, представлених в КЛЕ планера. Відповідні графіки наведені на рис. 7.1.

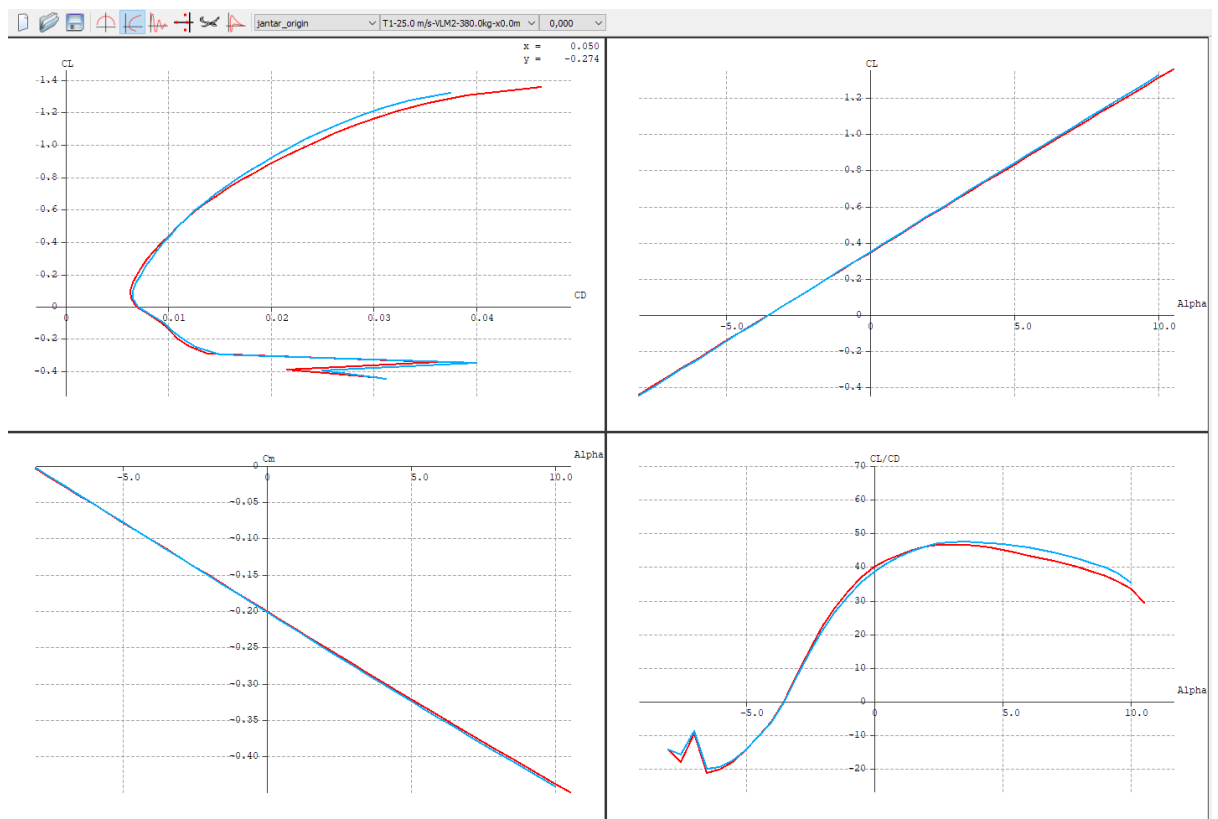


Рис. 7.1. Порівняння синтезованого планера з продувкою SZD-48-3.

Розбіжності в геометричних характеристиках цих планерів наведені в таблиці

3.

Порівняння синтезованого планера з продувкою SZD-48-3

	SZD-48-3	Синтезований планер
Розмах крила, м	15	15
Площа крила, м ²	10,66	11,55
Коренева хорда, м	0,95	0,93
Кінцева хорда, м	0,457	0,55
Форма в плані	Трапеція	Еліпс

При більш детальному розгляді поляр крила досліджуваного планера і еталонного (див. рис. 7.2) можна помітити що на ділянці від $C_y = 0$ до $C_y = 0,5$ графік оригінального планера (зображений на рисунку синім) знаходиться правіше по осі C_x , а на ділянці $C_y \geq 0,5$ навпаки, лівіше. Це свідчить про те, що в діапазоні швидкостей від 109 і вище, які відповідають умові $0 \leq C_y \leq 0,5$, планер буде володіти більшою аеродинамічною якістю, тоді як на нижчих швидкостях – дещо нижчою.

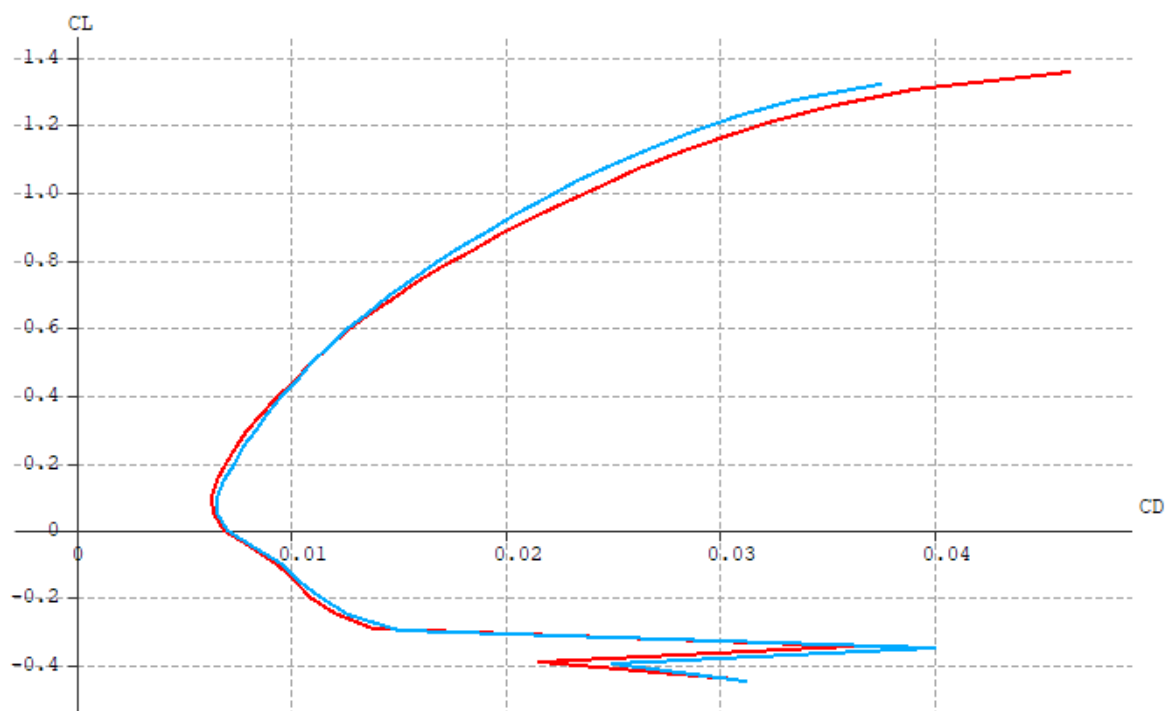


Рис. 7.2. Порівняння поляр

Розглянемо ці два планери з точки зору алгоритму оптимізації. Побудувавши графіки величини, описаної законом (12) очевидно, що площа під графіком «SZD-48-3», який відповідає оригіналу, буде більшою від площі під графіком дослідного крила. А отже і критерій оптимальності буде нижчим.

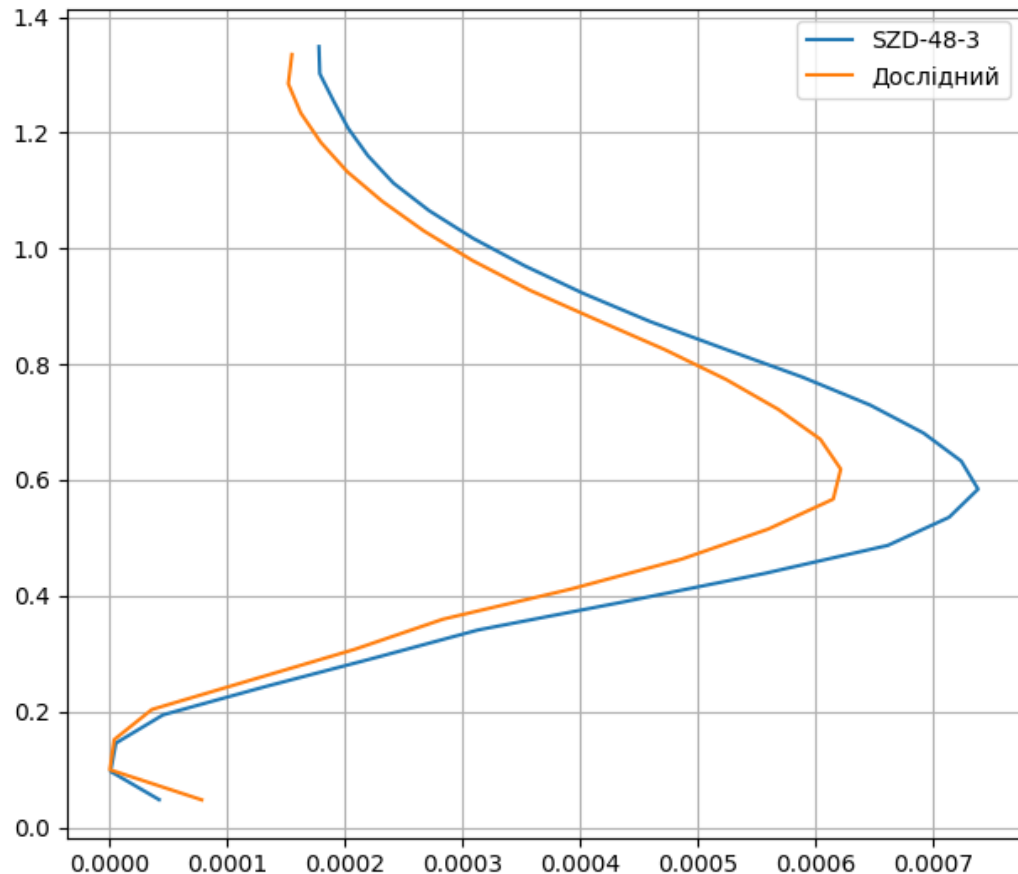


Рис. 7.3. Порівняння критеріїв оптимальності.

Дійсно, для оригінального планера $f(W) = 0.00047$, тоді як для дослідного планера $f(W) = 0.0004$.

За результатами роботи виконаної в розділі 3 цієї магістерської дисертації відношення часу в потоках до часу на переходах для планера даного типу становить 69:31. Тобто більшу частину всього часу, а також і маршруту планер проведе на переході. При переходах, в залежності від маси планера, наявності водобаласту, середньої швидкості набору в потоках пілот встановлює швидкість рівну, а частіше більшу за швидкість максимальної якості. Тому, звідси можна зробити висновок що і середня швидкість проходження маршруту буде вища для спроектованого планера.

Висновки по розділу

У цьому розділі був проведений аналіз роботи алгоритму оптимізації на основі отриманих даних. Так було прийняте рішення використовувати продувку крила планера SZD-48-3 в програмі Xflr5 за точку відліку і порівнювати отримані характеристики з продувкою синтезованого алгоритмом планера. Таким чином з порівняння виключаються похибки, пов'язані з відхиленням продувочних даних від експериментальних. Були побудовані графіки основних аеродинамічних залежностей обох планерів та проаналізовані відмінності між ними. Зроблений висновок про те, що у крила планера, оптимізованого розробленим алгоритмом спостерігається приріст аеродинамічної якості на швидкостях, які відповідають швидкостям на переходах. Це у свою чергу сприяє збільшенню середньої швидкості проходження маршруту.

8. СТАРТАП ПРОЕКТ

8.1. Опис ідеї проекту

8.1.1 Зміст ідеї

«Glider Characteristics Optimizer» або «GCO» – це незалежна команда розробки програмного забезпечення для високоточних інженерних обчислення. Програмне забезпечення сконцентроване в першу чергу на оптимізації аеродинамічних характеристик планерів на основі статистичних даних польотів та моделі атмосфери і потенційно може поширюватися і на спорідненні інженерні питання. Опис ідеї наведений у таблиці 8.1

Таблиця 8.1

Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Оптимізація аеродинамічних характеристик планера на основі статистичних даних польотів та моделі атмосфери	1. На ринку вже існуючих літальних апаратів	Оптимізація при виробництві або оновленні лінійок продукції
	2. На ринку розробників нових літальних апаратів	Можливість скористатися готовим рішенням замість витрат власних коштів, часу, людино-годин. Попередження виникнення додаткових проблем чи витрат, пов'язаних з сферою розробки стартап-проекту
	3. На ринку наукових досліджень та випробувань	Скорочення витрат на розробку ПЗ, швидке та протестоване рішення
	4. Цивільна, військова авіації	Аналогічно до напрямків 1 та 2
	5. Спорт, індустрія відпочинку	

8.1.2 Аналіз техніко-економічних переваг ідеї

Таблиця 8.2

Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	Товари/концепції конкурентів				Слабка сторона	Нейтральн сторона	Сильна сторона
		Мій проект	Flow5	Ansys	Matlab			
1	Економічність (вартість)	низька	низька	дуже висока	висока			X
2	Продуктивність	середня	середня	висока	висока	X		
3	Час розробки	1 міс	2 міс	6 міс	5 міс			X
4	Екологічність	Мінімальний вплив на навколишнє середовище					X	
5	Технологічна досконалість	середня	низька	висока	висока		X	
6	Портативність	висока	середня	низька	низька			X

В порівнянні з конкурентами продукт має перевагу в ціні, часі розробки та портативності. Завдяки ціні та часу розробки можна нівелювати відставання в продуктивності, а краща портативність (відсутність жодних вимог до фізичного транспортування) дає додаткову перевагу над конкурентами.

8.2. Технологічний аудит ідеї проекту

Технологічний аудит - це спосіб перевірки технологічного стану підприємства, рівня організації всіх важливих процесів за допомогою встановлених критеріїв, виявлення сильних та слабких сторін виробництва.

Аудит технологій за допомогою яких можна реалізувати ідею проекту зведено до таблиці 8.3.

Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Використання даних польотів	Статистичний збір та обробка інформації спеціальними установами	Розроблена та імплементована спеціальними профільними установами	Вільний доступ
2	Використання моделі атмосфери	Створення моделі спеціалістами галузі	Розроблена та перевірена спеціалістами галузі	Вільний доступ
3	Обчислення аеродинамічних характеристик планеру	Програмне забезпечення на основі інженерних формул та обрахунків	Наявні слабкі аналоги таких обчислень	Сильно обмежений доступ
4	Порівняння та оптимізація показників різних моделей планерів	Програмне забезпечення	Наявні слабкі аналоги такого функціоналу	Вкрай обмежений
Обрана технологія реалізації ідеї проекту: програмне забезпечення				

Аудит показав, що технології для реалізації ідеї доступні. Отже розробка і виготовлення такого програмного забезпечення можливе.

8.3. Аналіз ринкових можливостей запуску стартап-проекту

8.3.1. Аналіз попиту

Попередня характеристика потенційного ринку стартап-проекту зведена до таблиці 8.4.

Таблиця 8.4

Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальних обсяг продаж, грн/ум.од	інформація недоступна
3	Динаміка ринку (якісна оцінка)	зростає
4	Наявність обмежень для входу (характер обмежень)	сертифікації та перевірка на відповідність стандартам безпеки
5	Специфічні вимоги до стандартизації та сертифікації	Відповідність інженерним стандартам та стандартам безпеки
6	Середня норма рентабельності в галузі (або по ринку), %	40

Виконавши попередню характеристику ринку стартап-проекту можемо зробити висновок, що кількість конкурентів на ринку обмежена, а показник рентабельності ідеї достатньо високий. Отже за відносно короткий часовий проміжок можливо покрити усі затрати на розробку та виготовлення. Оскільки попит в даній галузі високий, присутня конкуренція не є завадою для виходу на ринок.

8.3.2. Потенційні групи клієнтів

У таблиці 8.5 зведено характеристики потенційного ринку стартап-проекту.

Таблиця 8.5

Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Професійне та точне обчислення даних, необхідних та критичних для виробництва планерів	- виробники планерів - експертно-аналітичні компанії - обслуговуючі компанії	Так як результатом роботи є точні та однозначні розрахунки, відмінності у використанні та поведінці різних груп не передбачається	- точність - безпека - оперативність - широка база даних по ринку

Проаналізувавши інформацію, зрозуміло що попит на ринку достатній для того, щоб виправдати витрати на отримати достатній прибуток. Споживач зможе отримати необхідний йому продукт оскільки розробка буде вестися на замовлення для кожного окремого проекту (планера, іншого літаючого засобу), що дозволить адаптувати характеристики продукту під кожного замовника.

8.3.3. Аналіз ринкового середовища

Для подальшого аналізу ринкового середовища, а саме ринкових загроз та можливостей, зведено таблиці 8.6 та 8.7

Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренти	Поява нових експертів в галузі	- розширення функціоналу - розширення та вдосконалення баз даних по ринку - оптимізація ціни та якості - залучення нових кадрів для розробки швидших та «легших» рішень
2	Зістарення технології	Програмне забезпечення постійно еволюціонує і доволі швидко може знецінитися	Постійний моніторинг технологій цієї сфери. Оновлення або повна зміна підходів до розробки ПЗ відповідно до ситуації
3	Кадрове питання	Нестача професіоналів такої вузької сфери, що вимагає високий рівень експертних знань	Розробка програм навчання, інternатури, співпраця з університетами для пошуку нових талантів

Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Потреба	Гостра потреба в швидких і простих рішеннях обчислювальних програм	Після вдалої розробки – задіяність в суміжних сферах
2	Розвиток сфери	Авіаційна сфери починає набирати обертів у державі	Очікується зріст попиту, тому зараз є шанс зайняти вдале місце в сфері
3	Визнання	На міжнародному ринку цінується якісний та	Підняття статусу як компаній, так і держви. Можливість

		оптимізований продукт, яким ми можемо вважати цей стартап-проект	задіяності в визнаних міжнародних проектах зі статусом професіоналів
--	--	--	--

З аналізу ринкового середовища стає зрозуміло, що є ряд факторів загроз та можливостей, на кожен з яких необхідно відповідати певними діями, оскільки ігнорування цих факторів негативно впливатиме на успіх проекту.

8.3.4. Аналіз пропозиції

Наступним кроком проводиться аналіз пропозицій, в якому визначаються загальні риси конкуренції (таблиця 8.8).

Таблиця 8.8

Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії)
1. Тип конкуренції – монополія\ олігополія\ монополістична\ чиста	Конкуренція вкрай обмежена в вузькій сфері авіації, чиста конкуренція	- розширення функціоналу - розширення та вдосконалення баз даних по ринку - оптимізація ціни та якості - залучення нових кадрів для розробки швидших та «легших» рішень
2. За рівнем конкурентної боротьби – локальний\ національний\ ...	Розглядається як локальна, так і міжнародна конкуренція	На міжнародному ринку цінується якісний та оптимізований продукт, яким ми можемо вважати цей стартап-проект. Підняття статусу як компанії, так і держави. Можливість задіяності в визнаних міжнародних проектах зі статусом професіоналів

3. За галузевою ознакою – міжгалузева\ внутрішньогалузева	Внутрішньогалузева, з перспективою росту в інженерній галузі загалом	Розширення на загально-інженерну галузь, так як тут особливо ціняться точні розрахунку зі знанням продукту, що легко приміняться на таких сферах як авто-будівництво, корабле-будівництво та таке інше
4. Конкуренція за видами товарів: -товарно-родова -товарно-видова -між бажаннями	Товарно-видова. Конкуренція одного типу товару	Впровадження нових технологічних рішень
5. За характером конкурентних переваг цінова\ нецінова	Цінова перевага. Значно менше ціна програмного забезпечення	Відслідкування цінової політики конкурентів
6. За інтенсивністю -марочна\не марочна	Марочна продукція. Зареєстрований бренд	Зареєстрована торгова марка

На сьогодні на конкуренція на ринку є доволі обмеженою, отже маємо чисту конкуренцію. Сфера використання програмного забезпечення з точних обчислень відносно мала, і ще більше обмежується статистичними даними, що повинні використовуватися для таких обчислень. Тому конкуренція залишається внутрішньо галузевою. Оскільки конкуренція ведеться між товарами одного виду, то конкуренція товарно-видова. Реєстрація торгової марки дозволить вести агресивну маркетингову компанію та будувати власне ім'я при потенційному розширення технологічної бази.

8.3.5. Детальний аналіз умов конкуренції в галузі

Наступним кроком проводиться більш детальний аналіз умов конкуренції в галузі (табл 4.9).

Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	flow5.tech ansys.com mathworks.com/products/matlab.html	Інформація відсутня	Flow Asys Matlab	Компанії, для яких важливе швидке обчислення	Продукти є взаємно-допоміжними та взаємно-замінючими
Висновки	Конкурентність може спровокувати змагання в якості продукту, що лише допоможе стартап-проекту довести свою цінність	Вихід на ринок не обмежується потенційними конкурентами, так як сфера розробки дуже вузька	Постачальники встановлюють планку якості продукту, яку потрібно побити для гладкого та вдалого виходу на ринок	Клієнти мало впливають на умови ринку, так як продукт я абсолютним і точним. Єдиний метод впливу – вимога кращого юзер експієнса	Залишається питання конкуренції, описане в попередніх пунктах

Опираючись на таблицю 8.9 можна зробити висновок, що як і для будь-якого товару основним фактором впливу є клієнт, так як саме від задоволень його потреб буде залежити успішність проекту та власне прибуток. Разом з тим, постачальники все ж залишаються вагомим фактором, оскільки саме від них залежить стабільність задоволення потреб та вимог клієнтів. Завдяки низькій відносній вартості пропонованого товару можливо створити суттєву конкуренцію іншим гравцям на ринку.

8.3.6. Фактори конкурентоспроможності

Продовжуючи тему конкурентної здатності стартап-проекту: на основі аналізу конкуренції, а також з врахуванням характеристик ідей проекту і т.д. визначаються та обґрунтовуються фактори конкурентоспроможності (таблиця 8.10).

Таблиця 8.10

Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1	Точність та швидкість виконання роботи	Точність є у кожного конкурента, та стартап-проект може запропонувати оптимізоване використання пам'яті та часу машинами для виконання тих самих обчислень
2	Краща рекомендаційна база	Ширші бази даних по існуючих моделях, більш глибокі дослідження відомих проблем з ними результуються в кращій експертизі та точніших рекомендаціях, що базуються на перевірених даних
3	Молода команда із ще живим ентузіазмом	Заржавівші люди з обмеженим та прагматичним поглядом на життя ніколи не зможуть створити дійсно кращий продукт. У цьому контексті стартап-проекти завжди будуть у виграші

8.3.7. Аналіз сильних та слабких сторін стартап-проекту

Далі, за визначеними факторами конкурентоспроможності проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 8.11).

Таблиця 8.11

Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали (1-20)	Рейтинг товарів-конкурентів у порівнянні						
			-3	-2	-1	0	1	2	3
1	Інновація	18			X				
2	Спеціальні пропозиції	8		X					
3	Робота з клієнтом	20		X					
4	Гарантія	15				X			
5	Цінова політика	20	X						

Порівняльний аналіз показав, що головними перевагами стартап-проекту перед конкурентом є високо оцінена робота з клієнтом та ціна на кінцевий продукт – програмне забезпечення.

8.3.8. SWOT-аналіз

Кінцевим етапом ринкового аналізу впровадження проекту є складання SWOT-аналізу (таблиця 8.12).

Таблиця 8.12

SWOT-аналіз стартап-проекту

Сильні сторони: <ul style="list-style-type: none">- надання експертних послуг в вузькій галузі- точність та швидкість виконання роботи- краща рекомендаційна база- молода команда із ще живим ентузіазмом	Слабкі сторони: <ul style="list-style-type: none">- можлива недостача професійних і досвідчених кадрів у випадку росту компанії- додаткові витрати на тестування продукту, щоб бути впевненими в конкурентоспроможності
---	---

Можливості: - потреба - розвиток сфери - визнання	Загрози: - конкуренти - зістарення технологій - нестача кадрів
---	--

8.3.9. Альтернативи ринкової поведінки

Опираючись на SWOT-аналіз розроблено альтернативи ринкової поведінки для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів (таблиця 8.13).

Таблиця 8.13

Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Корпоративне шпигунство – виток інформації від/до конкурентів та їх передчасна реалізація такого або покращеного продукту	Висока	1 міс.
2	Реєстрація франшизи	Висока	3 міс.

Перша з розглянутих альтернатив – це злиття проекту з одним із конкурентів, що є найтипівішим проявом корпоративного шпигунства. Дана маніпуляція дозволить значно прискорити запуск стартап-проекту, однак є небезпека повного поглинання конкурентом і втрата контролю над стартап-проектом. Іншою альтернативою є поширення стартап-проекту шляхом реєстрації франшизи. Таким чином увесь контроль та права залишаються у засновників проекту та проект отримує можливість швидше розповсюджуватися по усьому світу, та збільшувати залучені ресурси, як технологічні так і людські.

8.4. Розроблення ринкової стратегії проекту

8.4.1. Стратегія охоплення ринку

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (таблиця 8.14).

Таблиця 8.14

Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Аграрна сфера	Так	Середній	Відсутній на ринку України	Без перешкод
2	Військова сфера	Так	Високий	Відсутній на ринку України	Без перешкод
3	Авіа-будівнича сфера	Так	Високий	Відсутній на ринку України	Без перешкод
4	Приватні клієнти	Так	Середній	Відсутній на ринку України	Без перешкод
5	Спортивна сфера	Так	Середній	Відсутній на ринку України	Без перешкод

Як згадувалось раніше, основний попит припадатиме на розробників та виробників таких літаючих апаратів як планери. В першу чергу це цікаво власне інженерним компаніям та військовій промисловості. Але, разом з тим, не обмежується ними, адже навіть для спорту це залишається фактором розвитку.

8.4.2. Базова стратегія розвитку

Для роботи в обраних сегментах ринку необхідно сформувавши базову стратегію розвитку (таблиця 8.15).

Таблиця 8.15

Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Модернізація існуючого програмного забезпечення	Стратегія диференційованого впливу	Персоналізований підхід до кожного замовника. Низька ціна та спеціальні пропозиції	Стратегія диференціації

8.4.3. Стратегія конкурентної поведінки

Наступним кроком необхідно зробити вибір стратегії конкурентної поведінки (таблиця 8.16).

Таблиця 8.16

Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
ні	так	лише ті, що є базовою технологією	стратегія виклику лідера

Основною визначено стратегію виклику лідера. Головним чином тому, що даний стартап-проект вже не може стати першим гравцем на ринку, але може забезпечити краще співвідношення «ціна-якість».

8.4.4. Стратегія позиціонування

Для детального розгляду стратегії позиціонування зведено та проаналізовано таблиці 8.15 та 8.16, а також взято інформацію з вище наведеної таблиці 8.5. Стратегію позиціонування відображено в таблиці 8.17.

Таблиця 8.17

Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
Продукт з найкращим співвідношенням «ціна-якість», портативність	Стратегія диференціації	співвідношення «ціна-якість», портативність	Якість, портативність, надійсть

Стартап-проект по розробці програмного забезпечення для надточних обчислень має перевагу у співвідношенні «ціна-якість» та портативність (відсутність прив'язаності до великих виробничих потужностей). Особистий підхід до кожного з клієнтів надає перевагу у бізнесовому питанні при наявній конкуренції.

8.5. Розроблення маркетингової програми стартап-проекту

8.5.1. Формування маркетингової концепції товару

В першу чергу необхідно створити маркетингову програму концепції товару, який отримає споживач (таблиця 8.18).

Таблиця 8.18

Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Швидке, точне та персоналізоване програмне забезпечення	Співвідношення «ціна-товар», максимально персоналізоване рішення, що базується на статистичних даних	Ціна, робота з клієнтом, портативність

Перевагами продукту є низька ціна, портативність (відсутність прив'язаності до великих виробничих потужностей) та надання статистично точного і при цьому персонального рішення.

8.5.2. Маркетингова модель товару

Наступним кроком є розробка трирівневої маркетингової моделі товару (таблиця 8.19).

Таблиця 8.19

Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Програмне забезпечення для оптимізації аеродинамічних характеристик планера на основі статистичних даних польотів та моделі атмосфери		
II. Товар у реальному виконанні	Властивості\характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Швидкість та точність обчислень	М	Вр/Тх/Тл/Ор
	Ціна	М	Вр/Тх/Тл
	Портативність	М	Вр/Тх/Тл
III. Товар із підкріпленням	Персоналізований підхід, що технічно заснованих на великому об'ємі статистичних даних		

За рахунок чого потенційний товар буде захищено від копіювання:
запатентованість технології та потенційне створення франшизи

8.5.3. Визначення цінових меж

Наступним важливим кроком потрібно виконати визначення цінових меж (таблиця 8.20).

Таблиця 8.20

Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижні межі встановлення ціни на товар
1.	60 – 250 тис. грн	8 – 150 тис. грн	Високий	8 – 100 тис. грн

Оскільки рівень доходів у споживачів – високий, а вартість продукту менша від вартості товарів-замінників і товарів-аналогів, можна твердо сказати, що товар зацікавить потенційних клієнтів.

8.5.4. Визначення оптимальної системи збуту

Далі проводиться визначення оптимальної системи збуту таблиця 8.21.

Таблиця 8.21

Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Малий обсяг продажів на одного клієнта	Легкий доступ та доставка. Надання гарантії, сервісного (гарантійного) обслуговування	Глибоко всередині вузької індустрії інженерії	Веб продажі а також персональні маркетингові пропозиції найбільшим споживачам

8.5.5. Розроблення концепції маркетингових комунікацій

Завершаючим кроком маркетингової програми є розроблення концепції маркетингових комунікацій, відповідно до раніше обраної основи для позиціонування, визначену специфіку клієнтів таблиця 8.22.

Таблиця 8.22

Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Звичайна	Наукові виставки та презентації, професійні конференції та нетворкінг, інтернет ресурси	Найкраща ціна, персоналізовані рішення на основі точних статистичних даних, портативність	Заохочення користувача обрати саме цей проект	Концентрація на інженерній стороні серед професійних користувачів, спрощена версія для потенційно зацікавлених бізнесів

Висновки по розділу

За результатом розробки стартап-проекту з'ясовано, що проект має можливість комерціалізації, оскільки попит на ринку залишається не покритим. Перспективи виходу на ринок гарні, оскільки на ринку мало конкурентів, які можуть розробити товар-аналог і ще менше конкурентів, у яких вони пропонують його за ціною, доступною такій вузькій сфері інженерії. Як альтернативний розвиток стартап-проекту доцільно використати можливість модернізації вже існуючих ринку рішень. Підсумовуючи, стартап-проект має усі шанси для виходу на ринок, отриманні високої рентабельності та конкурентоспроможності за рахунок переваг описаних у розділі.

ВИСНОВКИ

У першому розділі даної магістерської дисертації були розглянуті праці за схожими темами, сформовані уявлення про методи оптимізації аеродинамічних характеристик планерів. На основі отриманих знань було обрано метод оптимізації, а саме оптимізація за статистичними даними. Сформовані етапи реалізації цього методу для досягнення поставленої задачі.

Одним із етапів був аналіз особливостей польоту планера за маршрутом. Це є дуже важливий етап, оскільки принципи побудови маршруту планера дуже суттєво відрізняються від польоту звичайного літака. Це обумовлено відсутністю СУ на планері та абсолютною його залежністю від стану погоди, розміщення та характеру різноманітних атмосферних явищ, таких як термальні потоки, потоки обтікання, хвильові потоки та інші. Саме тому були розглянуті математичні моделі та закони, які описують такий політ, а також взяті до уваги моделі атмосфери, які застосовуються для узагальненого опису польоту планера за маршрутом.

Третій розділ даної роботи присвячений знаходженню та збору інформації про польоти планерів. Велику роль у цьому процесі зіграв ресурс weglide.org, який надає доступ до файлів треків польотів, які викладають самі пілоти-планеристи. Цей ресурс є чудовим джерелом великої кількості даних, придатних для обробки. Дані про політ зберігаються у файлі формату `igc`, та є досить зручними для їх програмного аналізу. Для цього була написана програма мовою Python, яка виокремлювала зі всієї вибірки необхідні дані. В результаті було отримано інформацію час, проведений планеристами в наборі висоти та на переходах, а також відсотковий розподіл швидкостей впродовж маршруту. Детально ці питання розглянуто у розділі 3 даної роботи.

Наступним кроком було створення оптимізаційної моделі на основі отриманих даних. В якості основи оптимізаційного алгоритму був вибраний генетичний алгоритм. Базуючись на отриманих у розділі 3 даних було розроблено критерій оптимальності, який враховував узагальнену атмосферу впродовж маршруту, а також особливості аеродинамічних характеристик планера стандартного класу. Цільовою функцією оптимізаційної моделі була функція обчислення

аеродинамічних характеристик крила за Теорією несучої лінії. Також був розроблений генератор геометрії крила, який дозволив параметризувати процес створення вхідних даних для цільової функції. Всі елементи, розроблені в даному розділі були реалізовані у програмному коді.

За результатами роботи оптимізаційного алгоритму була створена просторова модель крила планера що дозволило оцінити його аеродинамічні характеристики за допомогою програмного забезпечення Xflr-5. З огляду на обмеження методу LLT, для побудови поляри швидкостей довелося враховувати опір фюзеляжу аналітично.

Для оцінки результатів роботи алгоритму оптимізації продувка крила з оптимальними геометричними параметрами порівнювалась з продувкою крила планера SZD-48-3, який був обраний за точку відліку. Результати порівняння показали переваги синтезованого крила перед крилом вихідного планера при проходженні маршруту.

Варто зазначити, що розроблена програма виконує оптимізацію досить тривалий час, тому є необхідність в ретельному налаштуванні програми а також мати у розпорядженні доволі великі обчислювальні потужності. Результатом роботи оптимізатора є складне в плані крило, кромка якого задається кривою третього порядку. Таке крило є досить дорогим у виготовленні.

За результатами проведених досліджень, розроблених моделей та програмного забезпечення був розроблений стартап проект, який показує комерційні можливості та актуальність даної роботи

.Список використаних джерел

1. Тисячук Олег. – Двомісний планер, бакалаврський диплом, КПІ імені Ігоря Сікорського, 2020
2. Fred Thomas; Judah Milgram. Fundamentals of sailplane design. College Park, MD : College Park Press, 1999.
3. Johannes J. Refinement of Glider Aerodynamic Design using CFD / J. Bosman Johannes. // TECHNICAL SOARING. VOL. 37, NO. 2 April–June 2013 – 2013. – С. 20–25.
4. J.L. van Ingen // The eN method for transition prediction. Historical review of work at TU Delft. // 38th Fluid Dynamics Conference and Exhibit 23 - 26 June 2008, Seattle, Washington
5. Carmichael, Bruce H., What Price Performance, Soaring Magazine, TECHNICAL SOARING. VOL. 37, NO. 4 October–December 2013. – С. 58–60.
6. Gerhard Waibel // Modeling thermals, Bad Hindelang, Germany
7. Prandtl L. Tragflügeltheorie. I. Mitteilung / L. Prandtl. // Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse. – 1918. – С. 451–477.
8. Anderson J. D. Fundamentals of Aerodynamics / John Anderson. – New York: McGraw-Hill, 2017. – 1130 с. – (6th ed.).
9. Drela M. XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils / Drela // Low Reynolds Number Aerodynamics / Drela. – Berlin: Springer, 1989. – (Lecture Notes in Engineering; т. 54).
10. Дэн С. Алгоритмы эволюционной оптимизации / Саймон Дэн. – Москва: ДМК Пресс, 2020. – 1002 с.
11. Sighard F. Hoerner: “Fluid dynamic drag. Practical information on Aerodynamic drag and hydrodynamic resistance”;
12. Simon D. Evolutionary Optimization Algorithms / Dan Simon. – Hoboken, New Jersey: John Wiley & Sons, Inc, 2013. – 742 с.
13. Snorri Gudmundsson. Design of Sailplanes, General Aviation Aircraft Design: Applied Methods and Procedures APPENDIX C4, 2014

Додаток 1.

Код програми для створення кубічних сплайнів.

```
import numpy as np
#from sympy.solvers import solve
#from sympy import Symbol

class Point:
    def __init__(self, xpos, ypos, zpos=0):
        self.x = xpos
        self.y = ypos
        self.z = zpos

def point_sign (p1, p2, p3):
    p1.x = np.round(p1.x, 2)
    p1.y = np.round(p1.y, 2)
    p2.x = np.round(p2.x, 2)
    p2.y = np.round(p2.y, 2)
    p3.x = np.round(p3.x, 2)
    p3.y = np.round(p3.y, 2)

    return (p1.x - p3.x) * (p2.y - p3.y) - (p2.x - p3.x) * (p1.y - p3.y)

def point_in_triangle_my (pt, v1, v2, v3):

    d1 = point_sign(pt, v1, v2)
    d2 = point_sign(pt, v2, v3)
    d3 = point_sign(pt, v3, v1)

    has_neg = (d1 < 0) or (d2 < 0) or (d3 < 0)
    has_pos = (d1 > 0) or (d2 > 0) or (d3 > 0)

    return not (has_neg and has_pos)

def curve_in_triangle(X1,Y1,X2,Y2,X3,Y3,X,Y):
    for xx, yy in zip(X, Y):
        if
point_in_triangle_my(Point(xx,yy),Point(X1,Y1),Point(X2,Y2),Point(X3,Y3)):
        #if not point_in_triangle(X1,Y1,X2,Y2,X3,Y3,xx,yy):
            return False
        return True

def cubic3points(x1, y1, x2, y2, x3, y3, rho):
    x4 = (x1+x3)/2;
    y4 = (y1+y3)/2;
```

not

```

x5 = x4+rho*(x2-x4);
y5 = y4+rho*(y2-y4);
points = np.array([[x1, y1], [x2, y2], [x3, y3], [x4, y4], [x5, y5]])
matrix = np.array([[x1**2, x1*y1, y1**2, x1, y1, 1],
                  [x3**2, x3*y3, y3**2, x3, y3, 1],
                  [x5**2, x5*y5, y5**2, x5, y5, 1],
                  [2*x1*(x1-x2), 2*y1*x1-y2*x1-x2*y1, 2*y1*(y1-y2), x1-x2, y1-y2, 0],
                  [2*x3*(x3-x2), 2*y3*x3-y2*x3-x2*y3, 2*y3*(y3-y2), x3-x2, y3-y2, 0]])
m1 = np.delete(matrix, 0, 1)
A = float(np.linalg.det(m1))

m2 = np.delete(matrix, 1, 1)
B = float(-np.linalg.det(m2))

m3 = np.delete(matrix, 2, 1)
C = float(np.linalg.det(m3))

m4 = np.delete(matrix, 3, 1)
D = float(-np.linalg.det(m4))

m5 = np.delete(matrix, 4, 1)
E = float(np.linalg.det(m5))

m6 = np.delete(matrix, 5, 1)
F = float(-np.linalg.det(m6))

if F != 0:
    A = A/F;
    B = B/F;
    C = C/F;
    D = D/F;
    E = E/F;
    F = F/F;
x = np.linspace(0, 15, 20)
y = np.linspace(0, 1.2, 20)

xx, yy = np.meshgrid(x, y, sparse=False)
z = A*xx**2+B*xx*yy+C*yy**2+D*xx+E*yy+F
#plt.contour(x, y, z)
return [A,B,C,D,E,F]

def cubeval(x, coefs):
    A = coefs[0]
    B = coefs[1]
    C = coefs[2]
    D = coefs[3]

```

```

E = coefs[4]
F = coefs[5]

#symx = Symbol('x')
#symy = Symbol('y')
#_yy=solve(A*symx**2+B*symx*symy+C*symy**2+D*symx+E*symy+F,
symy)

#results = []
#for j in _yy:
#    results.append(np.array([float(j.func(i)) for i in x]))
#return results

#return [res0, res1]
tmp = -4*A*C*x**2 + B**2*x**2 + 2*B*E*x - 4*C*D*x - 4*C*F + E**2
if len(tmp[tmp < 0]) > 0:
    return [],[]

if C == 0:
    return [-(A*x**2 + D*x + F)/(B*x + E), []]

if C != 0:
    res1 = (-B*x - E + np.sqrt(tmp))/(2*C)
    res2 = -(B*x + E + np.sqrt(tmp))/(2*C)
    return [res1, res2]

def sinspace(start, end, num):
    a = np.linspace(0, np.pi/2, num)
    x = np.sin(a)*end
    return x

```

Додаток 2.

Код програми-реалізації генетичного алгоритму

```
import random
from math import *
from Gen2Airfoil import *
from xfoil_dat import *
import sys
import os
from wing_generator import Wing
# remove all files from last run
filelist = [ f for f in os.listdir(".") if f.endswith(".dat") ]
for f in filelist:
    os.remove(f)
# print f
filelist = [ f for f in os.listdir(".") if f.endswith(".log") ]
for f in filelist:
    os.remove(f)

# Airfoil is defined by
# LEU = Leading edge up      LED = Leading edge down
# C25 = Camber at 25%      T25 = Camber at 25%
# C50 = Camber at 50%     T50 = Camber at 50%
# C75 = Camber at 75%     T75 = Camber at 75%

#      b1  b2  w1  e1  e2
#genmaxs = [ 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.1 ]
#genmaxs = [ 0.1, 0.05, 0.2, 0.2, 0.2, 0.2, 0.2, 0.1 ]
genmaxs = [ 1.5, 1.5, 0.99, 2.0, 2.0]# 0.1, 0.1, 0.1 ]

genmins = [ 0.5, 0.15, 0.01, -5.0, -5.0]# 0.01, 0.01, 0.01 ]

ngen = len (genmaxs)

foilnum = 0

def newborn(gen):
    global foilnum
    foilnum+=1
    return [0,gen,'%06d' %foilnum]
```

```

def check_range(gen):
    for i in range(0,ngen):
        if(gen[i]>genmaxs[i]):
            gen[i]=genmaxs[i]
        if(gen[i]<genmins[i]):
            gen[i]=genmins[i]

def proximity(gen1,gen2):
    proximity = 0
    for i in range(0,ngen):
        d = (gen1[i] - gen2[i])/(0.00001+abs(gen1[i] + gen2[i]))
        proximity+= d*d
    return sqrt(proximity)

def breed_random():
    print("        breed_random ")
    child = [0]*ngen
    for i in range(0,ngen):
        child[i] = random.uniform(genmins[i],genmaxs[i])
        check_range(child)
    return newborn(child)

def breed_interpolate(mother,father,weight):
    print("        breed_interpolate")
    child = [0]*ngen
    for i in range(0,ngen):
        child[i] = (1-weight)*mother[i] + weight*father[i]
    return newborn(child)

def breed_crossover(mother,father):
    print("        breed_crossover")
    child = [0]*ngen
    for i in range(0,ngen):
        if(random.random()>0.5):
            child[i] = mother[i]
        else:
            child[i] = father[i]
    return newborn(child)

def breed_mutate(mother, scale):
    print("        breed_mutate")
    child = [0]*ngen
    for i in range(0,ngen):
        child[i] = mother[i] * (1 + random.uniform(-scale,scale) )
        check_range(child)
    return newborn(child)

```



```

def gen2log(gen,logfile):
    #logfile.write(" %3.2f      " %gen[0])
    logfile.write (gen[2])
    logfile.write (' {0:10.2f}      '.format(gen[0]))
    #for i in range(0,ngen):
    # logfile.write(" %1.4f " %gen[1][i])
    #logfile.write("\n")
    logfile.write("[ %1.4f" %gen[1][0])
    for i in range(1,ngen):
        logfile.write(", %1.4f" %gen[1][i])
    logfile.write(' ]\n')

#          LEU LED  C25 C50 C75  T25 T50 C75
testgen = [ 0.5, 0.1, 0.1, 0.1, 0.3, 0.5, 0.6, 0.7 ]

#print testgen
#check_range(testgen)
#print
#print testgen
#print breed_random()

nsubjects = 15
nbest     = 10
name = "testfoil"

niterations = 30
Re = 40000
Ncrit = 9.0

logfile = open("logfile.log","w")

population = [[0,[],""] for i in range(0,nsubjects)]
print(len(population))
breedprob = [0]*9 # breeding probabilities

current_wing = Wing('zero')

def gen2wing(gen, name):
    global current_wing
    current_wing.generate_spline(*gen)
    return

def populate():
    for i in range (0,nsubjects):

```

```

#foilnum++
population[i] = breed_random()
#name = '%06d' % foilnum
gen2airfoil(population[i][1],population[i][2]) # generate Airfoil shape by
Bezier interpolation
Xfoil(population[i][2],Ncrit,Re) # compute fitness in Xfoil
population[i][0] = getLDmax(population[i][2]) # set fitness = LD
print(population[i][2],population[i][2],population[i][0])
gen2log(population[i],logfile)

```

```

def eval_fitness():
    for i in range (0,nsubjects):
        if (population[i][0] == 0):
            # evaluate fittens just for new ones

            gen2airfoil(population[i][1],population[i][2]) # generate Airfoil shape by
Bezier interpolation
            Xfoil(population[i][2],Ncrit,Re) # compute fitness in Xfoil
            population[i][0] = getLDmax(population[i][2]) # set fitness = LD
            print(population[i][2],population[i][0])
            gen2log(population[i],logfile)

```

```

def set_breedprob():
    global breedprob
    suma=0
    # breedprob[0] = 1.0 # modify 1
    # breedprob[1] = 1.0 # modify any
    # breedprob[2] = 1.0 # generate new
    # breedprob[3] = 1.0 # crossover 1 and 2
    # breedprob[4] = 1.0 # crossover 1 and X
    # breedprob[5] = 1.0 # crossover X and Y
    # breedprob[6] = 1.0 # interpolate 1 and 2
    # breedprob[7] = 1.0 # interpolate 1 and X
    # breedprob[8] = 1.0 # interpolate X and Y
    breedprob[0] = 1.0 # modify 1
    breedprob[1] = 0.1 # modify any
    breedprob[2] = 0.1 # generate new
    breedprob[3] = 0.1 # crossover 1 and 2
    breedprob[4] = 1.0 # crossover 1 and X
    breedprob[5] = 0.1 # crossover X and Y

```

```

breedprob[6] = 0.1 # interpolate 1 and 2
breedprob[7] = 0.1 # interpolate 1 and X
breedprob[8] = 1.0 # interpolate X and Y

for i in range (0,len(breedprob)):
    suma+=breedprob[i]
    breedprob[i] = suma
for i in range (0,len(breedprob)):
    breedprob[i]/=suma
for i in range (0,len(breedprob)):
    print(breedprob[i])

def choose_breeding():
    randnum = random.random()
    child = []
    if (( randnum > 0)&( randnum < breedprob[0])):
        child = breed_mutate(population[0][1],0.2) # modify 1 best
    elif(( randnum > breedprob[0])&( randnum < breedprob[1])):
        child = breed_mutate(population[random.randint(2,nbest-1)][1],0.1) #
modify any
    elif(( randnum > breedprob[1])&( randnum < breedprob[2])):
        child = breed_random() # generate new from
scratch
    elif(( randnum > breedprob[2])&( randnum < breedprob[3])):
        child = breed_crossover ( population[0][1] ,population[1][1]) #
crossover 1 and 2
    elif(( randnum > breedprob[3])&( randnum < breedprob[4])):
        child = breed_crossover ( population[0][1]
,population[random.randint(2,nbest-1)][1]) # crossover 1 and X
    elif(( randnum > breedprob[4])&( randnum < breedprob[5])):
        a = random.randint(0,nbest-1) # crossover
X and Y
        b = a
        while (b == a):
            b = random.randint(0,nbest-1)
            child = breed_crossover ( population[a][1] ,population[b][1])
        elif(( randnum > breedprob[5])&( randnum < breedprob[6])):
            child = breed_interpolate( population[0][1] ,population[1][1],0.5)
# interpolate 1 and 2
    elif(( randnum > breedprob[6])&( randnum < breedprob[7])):
        child = breed_interpolate( population[0][1]
,population[random.randint(2,nbest-1)][1],0.5) # interpolate 1 and 2
    elif(( randnum > breedprob[7])&( randnum < breedprob[8])):
        a = random.randint(0,nbest-1) #
interpolate X and Y
        b = a

```

```

while (b == a):
    b = random.randint(0,nbest-1)
    child = breed_interpolate ( population[a][1] ,population[b][1], 0.5)
return child

def evolve():
    population.sort( reverse=True)
    for i in range (0,nbest): # write out last survived
        print(" survived : ",population[i][2]," ",population[i][0])
        if (population[i][0] == 0):
            print("      replace zero ")
            population[i] = choose_breeding()
    for i in range (1,nbest): # replace duplicate genes
        dist = proximity(population[0][1],population[i][1])
        print("dist ",population[0][2]," ",population[i][2]," ",dist)
        if (dist < 0.5 ):
            print(" => kill ! ")
            population[i] = breed_random()
    for i in range (nbest,nsubjects): # replace less fit genes
        population[i] = choose_breeding()

if __name__ == '__main__':

```

```

    set_breedprob()

```

```

    print(" ===== iteration: 0")
    populate()
    for i in range (1,niterations):
        print(" ===== iteration: ",i)
        evolve()
        eval_fitness()
    logfile.close()

```

Додаток 3. Код програми – генератора крила.

```
import os
import time
import scipy as sp
import numpy as np
import numpy.matlib
import matplotlib.pyplot as plt
import pandas as pd
import cubic_spline as cs
from cubic_spline import Point
from scipy import interpolate as si
from subprocess import Popen, PIPE, STDOUT
```

```
precision = 4
verbose = False
plot_generator = False
save_wing = True
```

```
start_v = 80.0
stop_v = 285
```

```
def log(string):
    if verbose:
        print(string)
```

```
class Wing:
    # стандартний конструктор
    it = 0

    def __init__(self, name, span=15):
        # розмах, стандартний 15
        self.span = span
        # хорди
        self.b = np.ndarray([0,0])
        # положення хорд по осі z
        self.z = np.ndarray([0,0])
        # крутки
        self.e = np.ndarray([0,0])
        # профіль
        self.foil = 'FX_S_02-196.dat'
        # коренева хорда
        self.b0 = 0.0
        # кінцева хорда
        self.bk = 0.0
        # крутка в корені
        self.e0 = 0.0
```

```

# крутка закінцівки
self.ek = 0.0
# швидкість 90 км/год
self.V = 25
# кут атаки
self.AoA = 0
#
self.data = pd.DataFrame()
#
self.sections_data = pd.DataFrame()
#
self.name = name
#
self.Mass = 586.0 # кг
#
self.rho = 1.225 # кг/м3
#

if not os.path.exists(self.name):
    self.dir = self.name
    it = 0
else:
    it+=1
    self.dir = self.name+str(it)

#self.test()
# генерація крила, складеного з трапецій
# array_b - масив довжин хорд
# array_e - масив кутів крутки
# array_z - масив координат перерізів
# sections - кількість перетинів
def generate_trap(self, array_b, array_e, array_z, foils, sections):
    lb = len(array_b)
    le = len(array_e)
    lz = len(array_z)
    self.n = sections * lb
    if lb != le or le != lz or lb != lz:
        print("Array's shapes must be the same!")
        return 0
    endp = False

    for i in range(lb-1):
        if i == lb -2:
            endp=True

```

```

        self.z = np.append(self.z, np.linspace(array_z[i], array_z[i+1], sections,
endpoint=endp))
        self.b = np.append(self.b, np.linspace(array_b[i], array_b[i+1], sections,
endpoint=endp))
        self.e = np.append(self.e, np.linspace(array_e[i], array_e[i+1], sections,
endpoint=endp))
        print()
        # генерація крила сплайном
        # b1 - коренева хорда
        # b2 - кінцева хорда
        # w1 - вага сплайну хорд
        # e1 - кут крутки в корені
        # e2 - кут крутки закінцівки
        # w2 - вага сплайну крутки
        # sections - кількість перетинів

def generate_spline(self, b1, b2, w1, e1, e2, w2, sections):
    self.b0 = b1
    self.bk = b2
    self.e0 = e1
    self.e1 = e2

    # вершини трикутника
    t1 = Point(0.0, self.b0)
    t2 = Point(self.span/2, self.b0)
    t3 = Point(self.span/2, self.bk)

    #self.z = np.linspace(0, self.span/2, sections)
    self.z = cs.sinspace(0, self.span/2, sections)
    reprb = cs.cubic3points(t1.x, t1.y, t2.x, t2.y, t3.x, t3.y, w1)
    self.b_spline = reprb
    buff = cs.cubeval(self.z, reprb)
    for i in buff:
        if cs.curve_in_triangle(t1.x, t1.y, t2.x, t2.y, t3.x, t3.y, self.z, i):
            self.b = i
            print(i)

    # робимо крутку

#####

#####
self.e = np.linspace(e1, e2, sections)
#plt.plot([0, self.span/2, self.span/2], [e1, e1, e2], 'b.')
#plt.plot(self.z, self.e)
#plt.show()

```

```

# дзеркалим перетини
# self.b = self.b[::-1]
self.b = np.append(self.b[::-1], self.b, axis=0)
self.z = np.append(-self.z[::-1], self.z, axis=0)
self.e = np.append(self.e[::-1], self.e, axis=0)

if save_wing:
    pd.DataFrame({'b':self.b, 'z': self.z, 'e': self.e}).to_csv('wing.csv')

if plot_generator:
    fig, ax = plt.subplots()
    twinx = ax.twinx()

    print(f'z: {self.z}')
    print(f'b: {self.b}')
    print(f'e: {self.e}')

    ax.plot(self.z, self.b, 'b.-', label='хорди')
    ax.grid()
    ax.set_ylabel('Довжина хорди')
    ax.set_xlabel('Z')
    twinx.plot(self.z, self.e, 'r-', label='крутка')
    twinx.set_ylabel('Кут крутки')

    ax.legend()
    twinx.legend()
    plt.show()

# запуск xfoil для профіля крила
def getForces_CL_XFOIL_3(self, path, CL, Mach, Re, Ncrit, output_file):

    log('\n[Running xfoil]')
    log(f'XFOIL input: {path}, output: {output_file}')
    log(f'Cl: {CL}, Mach: {Mach}, \nRe: {Re}, Ncrit: {Ncrit}')

    def Cmd(cmd):
        ps.stdin.write(cmd)

    self.foil_AoA = 0.0
    self.foil_CD = 0.0
    self.foil_CDp = 0.0
    self.foil_Cm = 0.0
    self.foil_Top_Xtr = 0.0

```



```

self.foil_Bot_Xtr = 0.0

numNodes = '280'

saveFlnmPolar = f'{output_file}'
if os.path.exists(saveFlnmPolar):
    os.remove(saveFlnmPolar)
ps = Popen('xfoil.exe', stdin=PIPE,stderr=PIPE,stdout=PIPE)
ps.stderr.close()

def my_bytes(str, enc):
    #log(str, end="")
    return bytes(str, enc)

Cmd(my_bytes(f'PLOP\n', 'utf-8'))
Cmd(my_bytes(f'G F\n', 'utf-8'))
Cmd(my_bytes(f'\n', 'utf-8'))
Cmd(my_bytes(f'LOAD {path}\n', 'utf-8'))
Cmd(my_bytes(f'PPAR\n', 'utf-8'))
Cmd(my_bytes(f'N {numNodes}\n', 'utf-8'))
Cmd(my_bytes(f'\n\n', 'utf-8'))

Cmd(my_bytes(f'OPER\n', 'utf-8'))
Cmd(my_bytes(f'ITER 1000\n', 'utf-8'))
Cmd(my_bytes(f'VPAR\n', 'utf-8'))
Cmd(my_bytes(f'N\n', 'utf-8'))
Cmd(my_bytes(f'{Ncrit}\n', 'utf-8'))
Cmd(my_bytes(f'\n', 'utf-8'))
Cmd(my_bytes(f'VISC {np.around(Re,precision)}\n', 'utf-8'))
Cmd(my_bytes(f'Mach {np.around(Mach,precision)}\n', 'utf-8'))
Cmd(my_bytes(f'PACC\n', 'utf-8'))
Cmd(my_bytes(f'{saveFlnmPolar}\n', 'utf-8'))
Cmd(my_bytes(f'\n', 'utf-8'))
Cmd(my_bytes(f'CL {np.around(CL, 6)}\n', 'utf-8'))
Cmd(my_bytes(f'\n', 'utf-8'))
Cmd(my_bytes(f'quit\n', 'utf-8'))
ps.stdout.close()
ps.stdin.close()
ps.wait()

if os.path.exists(saveFlnmPolar):
    with open(saveFlnmPolar) as fidCP:
        lines = fidCP.readlines()
        dataBuffer = lines[-1].split()
        log(dataBuffer)

```

```

    #if type(dataBuffer[0]) == type('str'):
    #    print('[----FAIL!----]\n')
    #    return 0,0,0,0,0,0,0

    foil_AoA = float(dataBuffer[0])
    self.foil_CL = float(dataBuffer[1])
    self.foil_CD = float(dataBuffer[2])
    self.foil_CDP = float(dataBuffer[3])
    self.foil_Cm = float(dataBuffer[4])
    self.foil_Top_Xtr = float(dataBuffer[5])
    self.foil_Bot_Xtr = float(dataBuffer[6])

    log(['---SUCCESS---'])
else:
    print('Cannot find output file. Please check input data')
    print('[----FAIL!----]')
    return 0,0,0,0,0,0,0
return self.foil_AoA, self.foil_CL, self.foil_CD, self.foil_CDP, \
        self.foil_Cm, self.foil_Top_Xtr, self.foil_Bot_Xtr

#
def LLT(self):

    if os.path.exists('log.txt'):
        os.remove('log.txt')

    #Хорды и их положения по размаху для полукрыла
    L = self.span
    N = len(self.b)
    n = np.arange(1, N+1)
    V = self.V
    AoA = self.AoA
    self.S = np.trapz(self.b, self.z)

    self.cl_start = 2 * self.Mass * 9.8 / (self.rho * start_v**2*self.S)
    self.cl_end = 2 * self.Mass * 9.8 / (self.rho * stop_v**2*self.S)

    #Угловая координата сечения по преобразованию Прандтля-Глоерта
    th = np.arange(1, N+1) / N * np.pi

    #Сшиваем лево и право
    #b = np.append(b, b[::-1], axis=0)

    phi = self.e
    phi = phi * (np.pi/180)
    #Сшиваем половинки

```

```

alpha0 = np.zeros(len(phi))
#alpha0 = np.append(alpha0, alpha0[::-1], axis=0)
#Матрица полетного угла атаки, для всех сечений одинакова, в радианах
alpha = np.ones(len(n)) * self.AoA * np.pi / 180
#Снова рассчитываем линейные координаты сечений, для всех
#Формируем левую и правую половину уравнения, затем решаем его
res = np.pi * self.b / 4 / L * (alpha-alpha0-phi) * np.sin(th)
coef =
np.sin(np.outer(th,n))*(np.pi*np.outer(self.b,n)/4/L+np.matlib repmat(np.sin(th),N,1).T)
#a = coef/res;
a = np.linalg.lstsq(coef, res, rcond=None)
#Находим циркуляцию
gamma = 4*np.matmul(np.sin(np.outer(th,n)), a[0])
self.cl = 2 * gamma / self.b
#Находим скосы потока
tmp0 = np.outer(th, n)*(a[0] * n)
tmp_sin = np.sin(tmp0)
th_sin = np.sin(th)
w=-(tmp_sin / th_sin)
#Рассчитываем удлинение крыла
_lambda = (L)**2/self.S
#Рассчитываем Су крыла
self.CL=_lambda*np.pi*a[0][0]
#Считаем эффективность
eff = (1+np.matmul(n[1:],(a[0][1:]**2/a[0][0]**2)))
tmp=np.trapz(self.cl, self.z )
#Рассчитываем индуктивное сопротивление крыла
self.CDi=self.CL**2/np.pi/_lambda*eff
#Числа Маха и Рейнольдса
M = V/340
Re = self.b/1.46e-5*V
self.cl[-1] = 0
#Вектор Су в сечениях для которых заданы профиля
clcs = si.splrep(self.z[10:], self.cl[10:])
#clcl = si.splev(np.linspace(0, self.span, len(self.b)), clcs)
clcl = si.splev(self.z[10:], clcs)
#foils = [1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]
cd = np.zeros(int(N/2))

#print()
#print(f'CL: {self.CL}\nAoA: {self.AoA}\n Cl: {clcl}\n Re: {Re}')
print(f'CL: {self.CL}\nAoA: {self.AoA}')

for i in np.arange(0, int(N/2)):
    log(f'z: {self.z[i+10]}, b : {self.b[i+10]}')

```

```

        AoA, _CL, CD, CDp, Cm, Top_Xtr, Bot_Xtr =
self.getForces_CL_XFOIL_3(f'foils\\{self.foil}', self.cl[i], M,
                        Re[i+10],
9,
f'out\\{self.name}_a{self.AoA}_{i}.txt')
    cd[i] = float(CD)
    #self.cdb = np.append(cd[:-1], cd, axis=0) * self.b
    #self.sumcd = np.trapz(self.z, self.cdb)
    self.cdb = cd * self.b[10:] / self.S
    self.sumcd = 2 * np.trapz(self.cdb, self.z[10:])
    #self.sumcd += CDi
    self.CDf = self.sumcd/np.trapz(self.b, self.z)
    self.cdb = np.append(self.cdb[:-1], self.cdb, axis=0)
    print(f'sumcd: {self.sumcd}')

def LLT_range(self, angles):
    if len(angles) == 0:
        print('Empty range of AoA')
        return
    for idx, angle in enumerate(angles):
        print(f'[WING: {self.name}, ANGLE: {angle}, AoA : {self.AoA}
({idx+1}/{len(angles)}] ', end=")
        self.AoA = angle
        start_llt = time.time()
        stop_llt = time.time()
        try:
            start_llt = time.time()
            self.LLT()
            stop_llt = time.time()
            tmpdict = {'angle':[self.AoA], 'V': [self.V], 'CL':[self.CL],
'CD':[self.sumcd], 'Cdi': [self.CDi], 'CDf':[self.CDf]}
            self.data = self.data.append(pd.DataFrame(tmpdict),
ignore_index=True)

            sectdict = {'a':np.tile(angle, len(self.b)), 'V':np.tile(self.V, len(self.b)),
                        'z':self.z, 'b':self.b, 'e': self.e, 'cl':self.cl, 'cd':self.cdb}
            self.sections_data = self.sections_data.append(pd.DataFrame(sectdict))
            print(f'---SUCCESS---] [{stop_llt-start_llt}s]')
        except Exception as e:
            #plt.plot(self.z, self.cl, label=f'angle {self.AoA}')
            print(['----FAIL!----]' + str(e))

def write_summary(self, file=""):
    csvname1 = 'out\\'
    csvname2 = 'out\\'
    if file == "":

```

```

        csvname1 = self.name + '_wing'
        csvname2 = self.name + '_sections'
        if os.path.exists(csvname1) or os.path.exists(csvname2):
            csvname1 += str(self.it)
            csvname2 += str(self.it)
        else:
            csvname1 = file + '_wing'
            csvname2 = file + '_sections'
        self.data.to_csv(csvname1 + '.csv')
        self.sections_data.to_csv(csvname2 + '.csv')

    def plot_data(self):
        plt.plot(self.data['CD'] + self.data['Cdi'], self.data['CL'], 'b.-', label='myllt')
        xflr = pd.read_csv('xflr.csv')
        plt.plot(xflr['CD'], xflr['CL'], 'r.-', label='xflr')
        plt.legend()
        plt.show()

    def test(self):
        cls = np.arange(-0.2, 1.2, 0.1)
        cl = []
        cd = []
        for c in cls:
            AoA, _CL, CD, CDp, Cm, Top_Xtr, Bot_Xtr =
self.getForces_CL_XFOIL_3(f'foils\Foil_1.dat', c, 25/340,
                            1700000, 9,
f'out\\test{self.name}_a{self.AoA}_{c}.txt')
            print(f'AoA: {AoA}, CL: {_CL}, CD: {CD}')
            cl.append(c)
            cd.append(CD)
        plt.plot(cd, cl)
        plt.show()

# перевод швидкості в Cy
def speed2cy(speeds, mass, s, rho=1.225):
    cy = 2 * mass * 9.8 / (rho * speeds ** 2 * s)
    return cy

# зчитування розподілу швидкостей по відсоткам з файлу і перерахунок в
Cy
def speeds_percent(filename='final_speed_percent.csv'):
    sdf = pd.read_csv(filename)
    sdf['cl'] = 2 * self.Mass * 9.8 / (self.rho * self.V * self.S)
    return {'cl':sdf['cl'].tolist(), 'percent' : sdf['percent'].tolist()}

# функція офігенності

```

```

def eval_fitness(self):
    p = self.speeds_percent()
    p = p[p['speed'] > start_speed]
    p = p[p['speed'] < stop_speed]

    calc_data = self.data[self.data['CL'] > self.cl_start]
    calc_data = self.data[self.data['CL'] < self.cl_stop]

    res = np.trapz(p['percent'] * (calc_data['CD'] + calc_data['Cdi']),
calc_data['CL'])
    return res

if __name__ == '__main__':

    # ініціалізація об'єктів
    w1 = Wing('trap')
    w2 = Wing('taps')
    w3 = Wing('spline')
    weight = 0.5

    # генеруємо перше крило (два перетина - одна трапеція)
    # дві хорди - 1.5 і 1
    # крутка - від 0 до -2
    # координати по z - 0, 15
    # 10 перетинів
    w1.generate_trap([1.5, 1], [0, -2], [0, 7.5],["test", "test"], 10)

    # генеруємо друге крило (чотири перетини)
    w2.generate_trap([1.5, 1.45, 1.35, 1], [0, -0.5, -1, -2], [0, 3, 5, 7.5],
np.repeat("test", 4), 5)
    #          b1  b2 w1   e1 e2 w2 n
    w3.generate_spline(0.90, 0.4, 0.3, 0.1, -1.8, 1, 10)
    w3.foil = 'FX_67-K-17017.dat'

    # малюємо графіки
    plot_wing = False

    if (plot_wing):
        plt.plot(w1.z, w1.b, label='Трапеція')
        plt.plot(w2.z, w2.b, label='Трапеції')
        plt.plot(w3.z, w3.b, '-.', label=f'Сплайн, w = {weight}')
        plt.plot(w3.z, np.ones(40), '.', label=f'Сплайн, w = {weight}')
        plt.xlim(0, 8)
        plt.ylim(0.95, 1.6)
        plt.legend()
        plt.show()

```

```

start = time.time()
w3.LLT_range(np.arange(-14, 25, 1))
stop = time.time()
print(f'LLT_range execution time: {stop-start}')

w3.write_summary()
w3.plot_data()

fig, ax = plt.subplots(constrained_layout=True)
fig.subplots_adjust(right=0.75)
ax.plot(w3.z, w3.b, 'g-', label='Хорди')
#ax.legend(loc='upper right')
#plt.plot(w3.z, w3.foil_CL, label='Cy xfoil.')
cy_ax = ax.twinx()
p_clax, = cy_ax.plot(w3.z, w3.cl, 'b-', label='Cy розрах')
cy_ax.set_ylabel('Cy')
cy_ax.legend(loc='center right')

cx_ax = ax.twinx()
cx_ax.spines["right"].set_position(("axes", 1.2))
cx_ax.set_ylabel('Cx')
p_cxax, = cx_ax.plot(w3.z, w3.cdb, 'r-', label='Cx перерізів')
cx_ax.legend(loc='lower right')

#plt.plot(w3.z, w3.cl, label='Cy розрах')
#plt.plot(w3.z, w3.cdb, label='Cx перерізів')
plt.legend(loc='upper right')
plt.show()

```