

Міністерство освіти і науки України
Криворізький національний університет
Факультет інформаційних технологій
Кафедра автоматизації, комп'ютерних наук і технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеню вищої освіти – магістр
за освітньо-професійною програмою
«Кіберфізичні системи в промисловості, бізнесі та транспорті»

зі спеціальності

174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

тема роботи:

***«Комп'ютерна система керування завантаженням літаків з
урахуванням визначеного центру маси у площині вантажних
відділень»***

Виконав ст. гр. АКІТР-23-2м. _____ Матрос В. В.

Керівник _____ Савицький О. І.

Нормоконтроль _____ Маринич І. А.

Завідувач кафедри _____ Рубан С. А.

КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет: інформаційних технологій

Кафедра: автоматизації, комп'ютерних наук і технологій

Ступінь вищої освіти: Магістр

Спеціальність: 174 – Автоматизація, комп'ютерно-інтегровані технології та
робототехніка

ЗАТВЕРДЖУЮ

Зав. кафедри: к.т.н. Рубан С.А.

« 5 » липня 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра

студентові групи АКІТР-23-2м. Матросу Владиславу Віталійовичу

1. Тема кваліфікаційної роботи: «Комп'ютерна система керування
завантаженням літаків з урахуванням визначеного центру маси у площині
вантажних відділень»

затверджено наказом по університету № 595с від 04.07.2024 р.

2. Термін здачі кваліфікаційної роботи: 01.12.2024 р.

3. Склад кваліфікаційної роботи: Пояснювальна записка обсягом 78с., додатки,
презентація у Microsoft PowerPoint (15 слайдів) в електронному та друкованому
вигляді

4. Консультанти кваліфікаційної роботи:

Розділ 1-3

к.т.н. доц. Савицький О. І.

5. Календарний план:

№	Етапи роботи	Термін виконання
1	<i>Вступ</i>	<i>20.07.24</i>
2	<i>Розділ 1</i>	<i>16.08.24</i>
3	<i>Розділ 2</i>	<i>18.09.24</i>
4	<i>Розділ 3</i>	<i>19.10.24</i>
5	<i>Висновки</i>	<i>15.11.24</i>
6	<i>Оформлення кваліфікаційної роботи</i>	<i>24.11.24</i>
7	<i>Підготовка презентації та графічного матеріалу</i>	<i>28.11.24</i>
8	<i>Підготовка доповіді до захисту</i>	<i>01.12.24</i>

6. Дата видачі завдання: 28.06.2024р.

Керівник _____ /Савицький О. І./

7. Запевнення: Я, Матрос Владислав Віталійович, запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про академічну доброчесність Криворізького національного університету ознайомлений.

Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі умисних порушень робота не допускається до захисту або оцінюється незадовільно.

Здобувач _____ /Матрос В. В./

АНОТАЦІЯ

Матрос В. В. «Комп'ютерна система керування завантаженням літаків з урахуванням визначеного центру маси у площині вантажних відділень».

Кваліфікаційна робота на здобуття ступеню вищої освіти магістр за освітньо-професійною програмою «Кіберфізичні системи в промисловості, бізнесі та транспорті» 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка. – Криворізький національний університет, Кривий Ріг, 2024.

Об'єктом кваліфікаційної роботи є дослідження та аналіз процесу реалізації автоматизованого керування процесом завантаження літаків.

У першому розділі було досліджено та проаналізовано теоретичні основи завантаження літаків, принципи функціонування існуючих аналогів систем керування, були поставлені актуальність задач систем керування завантаженням літаків та комплекс завдань розробки систем керування завантаженням літаків з урахуванням визначеного центру маси у площині вантажних відділень.

У другому розділі було розроблено математичні моделі для системи керування завантаженням літаків та досліджено синтез керування процесом систем керування завантаженням літаків.

У третьому розділі було розроблено структурну, функціональну та на основі їх принципів модель програмного комплексу систем керування завантаженням літаків і також програмні класи та модулі програмного проекту.

Ключові слова:

АВТОМАТИЗАЦІЯ, ЗАВАНТАЖЕННЯ ЛІТАКІВ, ВАНТАЖ, АВІАТРАНСПОРТ, ШТУЧНИЙ ІНТЕЛЕКТ, МАШИННЕ НАВЧАННЯ, СИСТЕМА КЕРУВАННЯ.

ANNOTATION

Matros V. V. «Computer control system for aircraft loading taking into account a specific center of mass in the plane of cargo compartments».

Graduation master`s work for obtaining an educational degree «Master» for the educational and professional program «Cyber-physical systems in industry, business and transport» 174 - Automation, computer-integrated technologies and robotics. - Kryvyi Rih National University, Kryvyi Rih, 2024.

The object of the qualification work is the study and analysis of the process of implementing automated control of the aircraft loading process.

In the first section, the theoretical foundations of aircraft loading, the principles of operation of existing analogues of control systems were studied and analyzed, the relevance of the tasks of aircraft loading control systems and the set of tasks for developing aircraft loading control systems taking into account a specific center of mass in the plane of cargo compartments were set.

In the second section, mathematical models for the aircraft loading control system were developed and the synthesis of the process control of aircraft loading control systems was investigated.

In the third section, a structural, functional and, based on them, a principle model of the software complex of aircraft loading control systems was developed, as well as software classes and modules of the software project.

Keywords:

AUTOMATIZATION, AIRCRAFT LOADING, CARGO, AIR TRANSPORT, ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, CONTROL SYSTEM.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ПІДХОДІВ ДО ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПРОЦЕСУ	10
1.1 Аналіз теоретичних основ та проблема завантаження літаків.....	10
1.2 Аналіз принципів функціонування існуючих аналогів систем керування завантаженням літаків	15
1.3 Актуальність задачі подальшої інтелектуалізації систем керування завантаження літаків	17
1.4 Постановка комплексу задач розробки систем керування завантаження літаків з урахуванням визначеного центру маси у площині вантажних відділень.....	21
РОЗДІЛ 2 ІДЕНТИФІКАЦІЯ МАТЕМАТИЧНОЇ МОДЕЛІ ТА СИНТЕЗ СИСТЕМИ КЕРУВАННЯ ПРОЦЕСОМ	24
2.1 Розробка математичних моделей для систем керування завантаженням літаків	24
2.2 Дослідження синтезу керування процесом систем керування завантаженням літаків	29
РОЗДІЛ 3 ПРОГРАМНО-ТЕХНІЧНА РЕАЛІЗАЦІЯ СИСТЕМИ КЕРУВАННЯ ПРОЦЕСОМ.....	44
3.1 Розробка структурної і функціональної моделей програмного комплексу систем керування завантаженням літаків.....	44
3.2 Розробка принципової схеми програмного комплексу систем керування завантаженням літаків	48
3.3 Розробка програмних класів та модулів програмного проекту.....	54
ВИСНОВКИ.....	69
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	70

Додаток А.....	73
----------------	----

ВСТУП

У сучасних умовах стрімкого розвитку авіаційних перевезень значно підвищуються вимоги до ефективного та безпечного завантаження літаків. Однією з ключових проблем є оптимальний розподіл маси вантажу в літаку, що безпосередньо впливає на стабільність польоту, а також на витрати пального і загальну економічну ефективність рейсу. Неправильне балансування вантажу може призвести до втрати керованості повітряного судна, що створює загрозу для життя пасажирів і команди. Тому питання автоматизації процесу завантаження з урахуванням визначеного центру маси набуває надзвичайної актуальності.

Для вирішення цих задач створено комп'ютерну систему керування завантаженням літаків, яка дозволяє автоматизувати процес розподілу вантажу в вантажних відділеннях, враховуючи задані параметри, такі як вага, об'єм та місце розташування кожної одиниці вантажу. Це сприяє забезпеченню безпечних польотів, оптимізації роботи авіакомпаній і підвищенню економічної ефективності перевезень.

Огляд літератури підтверджує важливість подальших досліджень у цьому напрямку. Існуючі роботи свідчать про те, що проблема ефективного та безпечного завантаження літаків залишається відкритою, а розробка високотехнологічних систем керування може принести значний внесок у вирішення цієї проблеми. Актуальність даної теми визначається не лише потребою в удосконаленні пасажирських та вантажних перевезень, але й стрімким розвитком сучасних технологій, які можуть бути успішно застосовані в авіаційній сфері.

У світлі вищезазначеного, метою даної науково-дослідної практики для магістерської роботи є підготовка до розробки та впровадження комп'ютерної системи керування завантаженням літаків, яка забезпечить максимальну безпеку та ефективність польотів, визначити прогалини в дослідженнях даної теми та визначити напрямки подальших наукових розвідок.

Метою кваліфікаційної роботи є розробка та впровадження комп'ютерної системи для автоматизованого керування процесом завантаження літаків, яка враховує положення центру маси у площині вантажних відділень.

Об'єктом кваліфікаційної роботи є дослідження та аналіз процесу реалізації автоматизованого керування процесом завантаження літаків.

Предметом кваліфікаційної роботи є комп'ютерна система керування завантаженням літаків з урахуванням визначеного центру маси у площині вантажних відділень.

Задачею кваліфікаційної роботи є розробка комп'ютерної системи керування завантаженням літаків з урахуванням визначеного центру маси у площині вантажних відділень.

РОЗДІЛ 1

АНАЛІЗ ПІДХОДІВ ДО ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПРОЦЕСУ

1.1 Аналіз теоретичних основ та проблема завантаження літаків

Теоретична база обраної проблеми кваліфікаційної роботи ґрунтується на комплексному вивченні сучасних підходів до керування завантаженням літаків, зокрема врахування центру маси в площині вантажних відділень. Проблема ефективного розподілу маси є однією з ключових у авіаційних перевезеннях, оскільки від цього залежить безпека польотів та економічна ефективність використання авіапарку. Категорії та поняття, що розглядаються в даній роботі, включають визначення центру маси, оптимізацію завантаження, балансування літака та автоматизацію цих процесів. Центр маси – це точка, в якій зосереджена уся маса системи, і його точне визначення є критично важливим для збереження стабільності та керованості літака під час польоту.

Сучасні дослідження у сфері авіаційної інженерії та логістики спрямовані на розробку методів, що дозволяють знизити час завантаження літаків, підвищити точність розподілу маси вантажу, мінімізувати помилки людського фактору та впровадити автоматизовані системи керування цими процесами. Закономірності розподілу маси тісно пов'язані з основними фізичними принципами, що регламентують стійкість літака. Характеристики таких систем залежать від технічних параметрів літака, специфіки вантажу та типу польоту. Одним із ключових показників ефективності завантаження є витрати пального, які значно знижуються при правильно збалансованому завантаженні літака.

Законодавче регулювання цієї сфери базується на міжнародних авіаційних нормах та стандартах, таких як правила ІАТА (Міжнародна асоціація повітряного транспорту) та ІСАО (Міжнародна організація цивільної авіації). Вони визначають стандарти безпеки завантаження, вимоги до розподілу вантажу, а також встановлюють процедури сертифікації систем, що використовуються для

автоматизації процесу. Важливим фактором, що впливає на ефективність процесу завантаження, є обмеження щодо допустимих відхилень від розрахункового центру маси, які контролюються відповідно до законодавства.

Стратегічні аспекти проблеми полягають у необхідності підвищення ефективності авіап перевезень в умовах глобалізації та збільшення попиту на швидкі й безпечні перевезення вантажів. Фактичні аспекти включають реальні виклики авіакомпаній, які стикаються з проблемами перевантаження, неправильного розподілу маси та недосконалості існуючих систем завантаження. Сучасні тенденції вказують на те, що автоматизація та інтеграція комп'ютерних систем у процеси завантаження дозволяють значно підвищити ефективність авіап перевезень.

Ступінь вивченості цієї проблеми досить високий, проте залишається актуальною необхідність подальшого вдосконалення автоматизованих систем для оптимального розподілу маси вантажу. Сучасні наукові дослідження підтверджують, що впровадження алгоритмів машинного навчання та штучного інтелекту у ці системи може ще більше підвищити точність і швидкість завантаження, зменшити ймовірність помилок та покращити загальні результати авіап перевезень.

Планування навантаження – це стратегія, яка складається з кількох елементів, пов'язаних разом для досягнення ефективності та безпеки транспортування. Ось основні компоненти планування навантаження. Ефективний процес планування завантаження передбачає повне розуміння технічних характеристик транспортного засобу доставки, включаючи вантажопідйомність, об'єм, конфігурацію салону та висоту підлоги вантажу. Ці знання є вирішальними для оптимізації розміщення вантажу [1]. Сумісність вантажів – це практика групування подібних вантажів для запобігання пошкодженню та забезпечення безпечного транспортування. Це передбачає завантаження крихких предметів разом, утримання важких предметів окремо та уникнення розміщення несумісних речовин поруч. Розподіл вантажу означає стратегічне розташування товарів у транспортному засобі доставки для підтримки стабільності. Це включає розміщення важких предметів нижче та ближче до центру літака для більшої

стабільності, особливо під час поворотів і раптових маневрів. Використання обмежувачів, таких як ремені, стяжки та вантажні дошки, щоб запобігти зсуву товарів і підтримувати стабільність вантажу, також є частиною цього. Оптимізація маршруту є критичним аспектом планування навантаження. Це передбачає пошук найефективніших маршрутів доставки транспортних засобів на основі трафіку, наявності водіїв і бажаних часових вікон, що також дозволяє змінювати послідовність завантаження або вивантаження предметів на основі змін у вимогах [2].

Оптимізація операцій із завантаження вантажів справді має вирішальне значення для постійного розвитку авіакомпаній. У цей процес залучені різні зацікавлені сторони з різними цілями. Відділ продажів прагне максимізувати прибуток при мінімізації витрат, відділ наземного обслуговування прагне скоротити зусилля, пов'язані з обробкою, а експлуатація літаків зосереджена на оптимізації споживання палива та використання авіаційного засобу пакування (ULD) [3].

Авіаційний засіб пакування – це спеціальний авіаційний контейнер або піддон, призначений для перевезення багажу, вантажів та пошти в широкофюзеляжних, а також у деяких типах вузькофюзеляжних літаків (див. рис. 2.1). Дозволяє об'єднувати безліч різних видів вантажу та багажу в укрупнену, стандартизовану за габаритами вантажну одиницю (транспортний пакет), що значно спрощує та прискорює процес навантаження та розвантаження літака. Кожен пакет має окрему транспортну накладну, що дозволяє відстежувати його вміст [4].



Рисунок 1.1 – Авіаційний засіб пакування (ULD)

ULD відіграють важливу роль у консолідації вантажів перед завантаженням у вантажний відсік літака. Процес навантаження передбачає використання різних пристроїв, наприклад, гідравлічних підйомників (навантажувачів). Згодом ULD вручну переміщуються в кінцеве положення на літаку за допомогою роликової підлоги та фіксуються на місці сітками, щоб запобігти неправильному завантаженню.

Неправильне завантаження може призвести до серйозних наслідків, включаючи пошкодження відсіку літака під час польоту, виїзд злітно-посадкової смуги під час зльоту чи посадки та втрату контролю над польотом [5-7].

Щоб забезпечити безпечне та ефективне завантаження вантажів, авіаперевезення вантажів повинні відповідати загальноприйнятим правилам, обмеженням і процедурам завантаження. Інструкції із завантаження, надані

персоналом, відповідальним за процес, повинні відповідати вимогам до розподілу вантажу та балансу, як зазначено в документації салону літака.

Враховуючи складний характер повітряних вантажних перевезень, які залежать від швидкого виконання робіт і прагнуть скоротити час завантаження, вкрай важливо збалансувати обмеження безпеки без надмірного збільшення експлуатаційних витрат. У цій статті пропонується алгоритм, який стосується процесу призначення пріоритетів для контейнерів (ULD), враховуючи численні обмеження та забезпечуючи подальше розвантаження вантажу в попередньо визначеному порядку. Цей алгоритм має на меті оптимізувати процес завантаження вантажів, враховуючи різні цілі та обмеження, з якими стикаються різні зацікавлені сторони в авіаційній вантажній галузі.

Поточний план завантаження повітряного вантажу окремо розглядає проблему палетування повітряного вантажу (ACPP) і проблему ваги та балансування літака (WBP), що впливає на оптимізацію корисного навантаження та центру ваги літака (CG). Завдяки вдосконаленню обчислювальної потужності комп'ютера тепер можлива спільна комбінаторна оптимізація ACPP і WBP. Запропоновано три моделі цілочисельного лінійного програмування: двоцільова модель оптимізації (BOM), комбінаторна модель оптимізації (COM) і вдосконалена комбінаторна модель оптимізації (IOM). Цілями моделей є максимальна вантажопідйомність і найменше відхилення CG від визначеної цільової CG [8-11].

Моделі також враховують широкий спектр обмежень у фактичних процедурах пакування та укладання, таких як об'єм, вага, положення завантаження, баланс літака та інші аспекти літака та вантажних пристроїв. Чотири сценарії з різними умовними показниками для трьох моделей розв'язані для літака B777F за допомогою Gurobi. Результати обчислень демонструють, що BOM має найшвидшу швидкість рішення, але відхилення CG є найбільшим, і в деяких випадках результати відхилення CG є неприйнятними. COM має найдовший час вирішення, що важко витримати на практиці. Незважаючи на те, що обчислювальне вирішення займає трохи більше часу, ніж BOM, IOM пропонує найкраще рішення для оптимізації. На рисунку 2.2 детально показана проблема. Відповідно до

різноманітних обмежень набір предметів вантажу має бути призначено кільком ULD, а потім ці завантажені ULD мають бути призначені положенням вантажних відсіків літака, які спеціально призначені для ULD [12-15].

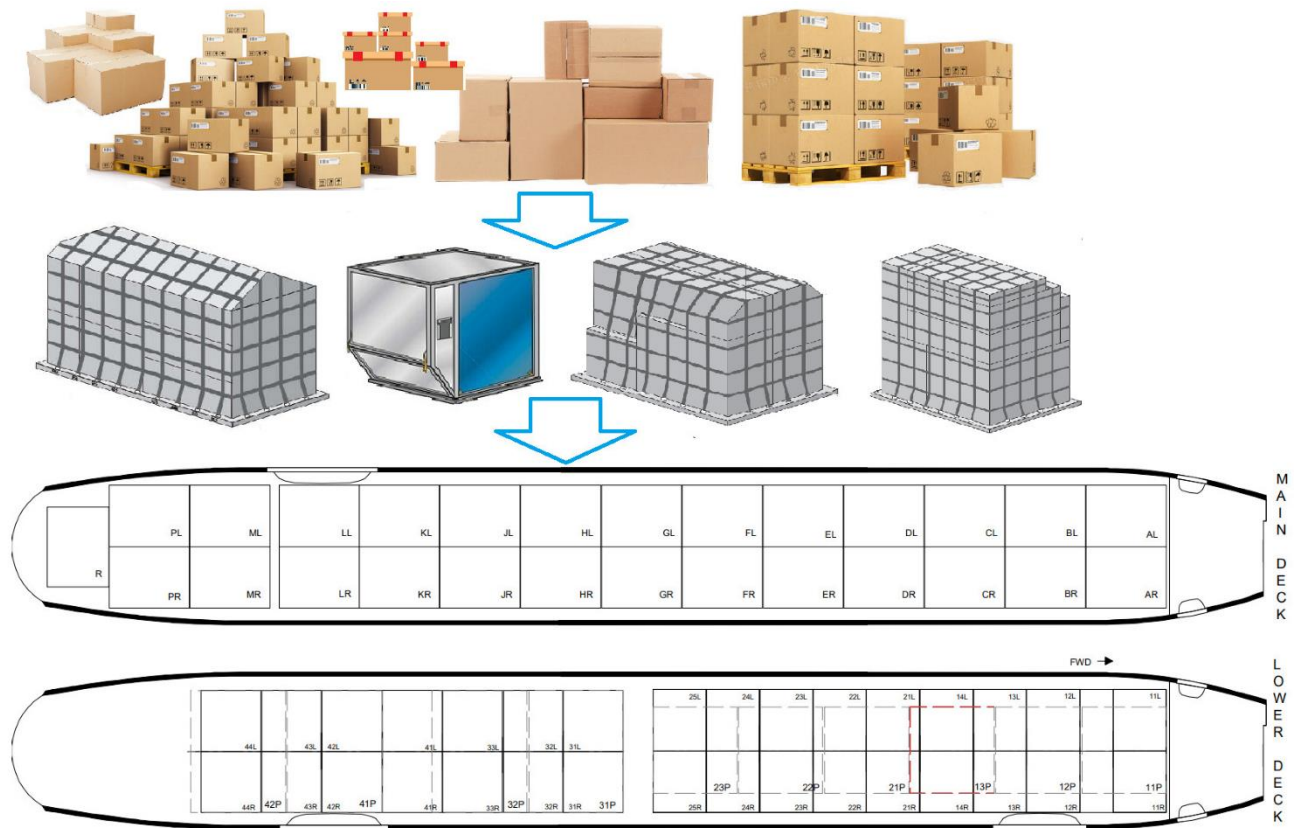


Рисунок 1.2 – Опис проблеми зі схемою розташування

1.2 Аналіз принципів функціонування існуючих аналогів систем керування завантаженням літаків

На основі узагальнення світового й вітчизняного досвіду автоматизованого керування в авіаційній галузі, можна відзначити декілька ключових концептуальних підходів до визначення понять автоматизованих систем керування завантаженням літаків. Світова практика показує, що автоматизація процесів у цій сфері має велике значення для підвищення ефективності та безпеки польотів. Основні підходи базуються на таких принципах, як оптимізація ресурсів, зниження впливу людського фактора, інтеграція інформаційних систем і впровадження штучного інтелекту.

Серед окремих видів автоматизованого керування, що використовуються в процесі завантаження літаків, можна виділити такі, як системи планування завантаження (Load Planning Systems), системи управління балансом літака (Weight and Balance Management Systems) і інтегровані логістичні платформи, що забезпечують взаємодію між різними підсистемами авіапідприємств. Ці системи працюють на основі алгоритмів, що враховують параметри вантажу, положення центру маси, а також дотримуються стандартів безпеки, що є критично важливим у сучасних умовах.

Основні завдання, які стоять перед автоматизованими системами керування завантаженням, включають: точне визначення положення вантажу для забезпечення стабільності літака, мінімізацію витрат пального через оптимальний розподіл маси, скорочення часу на завантаження та вивантаження літака, а також зниження ризиків, пов'язаних із людськими помилками. Джерела інформації для дослідження цієї теми включають міжнародні стандарти авіаційної безпеки (ІАТА, ІСАО), наукові дослідження у сфері авіаційної логістики та інженерії, а також практичні кейси впровадження подібних систем у авіакомпаніях по всьому світу. ІАТА асоціація, що виступає координатором та представником інтересів авіатранспортної галузі у таких галузях як забезпечення безпеки польотів, виробництво польотів, тарифна політика, техобслуговування, авіаційна безпека, розробка міжнародних стандартів спільно з ІСАО, яка досліджує проблеми організації міжнародної цивільної авіації, повітряних трас, створення аеропортів і аеронавігаційних засобів, розробляє міжнародні стандарти для конструювання й експлуатації повітряних суден, правила з використання устаткування, засобів зв'язку і контролю над польотами; сприяє уніфікації митних, імміграційних і санітарних правил і т. д.

Суть досліджуваного процесу полягає в автоматизованому розрахунку і коригуванні параметрів завантаження для кожного рейсу, що включає аналіз характеристик вантажу, визначення оптимального розподілу маси та контроль дотримання вимог до безпеки. У сучасних автоматизованих системах використовується машинне навчання для вдосконалення цих процесів, що дозволяє

враховувати велику кількість факторів у реальному часі, таких як погодні умови, тип повітряного судна та специфіка польоту.

Дискусійні питання, пов'язані з автоматизацією процесів завантаження, стосуються рівня автономності таких систем та можливості повної заміни людського фактору. Частина дослідників стверджує, що автоматизовані системи можуть не враховувати деякі специфічні ситуації, які потребують рішення оператора. Інші вважають, що розвиток штучного інтелекту дозволить створити більш надійні та точні системи, які здатні самостійно приймати рішення у складних ситуаціях.

На мою думку, повне усунення людського фактору в авіаційній галузі на даному етапі є неможливим, проте впровадження автоматизованих систем значно підвищує безпеку та ефективність. Автоматизація дозволяє знизити ризики помилок і оптимізувати процеси, що є особливо важливим в умовах зростаючих обсягів авіаперевезень. Однак, критично важливо забезпечити надійність та безпеку цих систем, особливо у випадках, коли доводиться мати справу з надзвичайними обставинами.

1.3 Актуальність задачі подальшої інтелектуалізації систем керування завантаження літаків

Актуальність задачі подальшої інтелектуалізації систем керування завантаженням літаків безпосередньо пов'язана з розвитком технологій та економічними викликами, які постають перед Україною, зокрема її регіонами, такими як Кривий Ріг. В умовах глобалізації та активної інтеграції у світові авіаційні процеси, підвищення ефективності перевезень вантажів стає важливою стратегічною задачею для держави, яка прагне розширювати свої можливості у міжнародних логістичних ланцюгах.

Останні роки показали, що Україна активно працює над розвитком своєї авіаційної інфраструктури, а також над покращенням умов для міжнародного співробітництва в сфері авіаперевезень. Статистичні дані за останні 3-5 років

демонструють позитивні тенденції у розвитку авіаційного сектора, хоча пандемія COVID-19 та початок повномасштабної війни тимчасово пригальмували цей процес. Наприклад, в Україні поступово збільшувався обсяг вантажних перевезень, зокрема повітряним транспортом, що свідчить про зростаючу роль країни в міжнародній логістиці.

Кривий Ріг, як один із найбільших промислових центрів України, має важливе стратегічне значення для економіки регіону та держави загалом. Промислові підприємства міста активно використовують повітряні перевезення для транспортування вантажів, зокрема металургійної продукції, що підкреслює важливість інтелектуалізації систем керування завантаженням літаків для оптимізації таких перевезень. Крім того, Кривий Ріг активно інтегрується у міжнародні логістичні ланцюги, що вимагає модернізації існуючих логістичних систем та впровадження нових технологій.

Згідно з офіційними статистичними даними, зростання вантажних перевезень в Україні, зокрема в регіонах, таких як Дніпропетровська область, де розташований Кривий Ріг, свідчить про необхідність подальшого розвитку сучасних інформаційних технологій у сфері авіаційної логістики. Однак, важливими проблемами залишаються недостатня автоматизація процесів, залежність від застарілих технологій та необхідність інтеграції нових інтелектуальних систем. Це стосується не лише комерційних авіаперевезень, але й стратегічних логістичних завдань, що мають важливе значення для безпеки держави.

Таким чином, у контексті сучасних проблем України та її регіонів, зокрема Кривого Рогу, інтелектуалізація систем керування завантаженням літаків має суттєве значення для підвищення ефективності та конкурентоспроможності українських підприємств на міжнародній арені. Використання штучного інтелекту та автоматизованих систем дозволить значно покращити процеси завантаження, зменшити витрати та підвищити безпеку авіаперевезень.

Також в межах розширення наукових досліджень та розробки новаторських технологій у галузі комп'ютерного керування завантаженням літаків з урахуванням центру маси, важливим етапом є патентний пошук. Цей етап дозволяє визначити

існуючі патентні рішення та забезпечити унікальність та оригінальність нового дослідження.

Перший описаний патент відомої авіаційної компанії Aurrigo визначає сучасну автоматизовану систему для ефективного завантаження вантажів на літак [21]. Система використовує інтегрованих роботів, оснащених високоточними механізмами, які автоматично переміщують та розміщують вантажі у вантажних відділеннях літака. Додатково, вона включає в себе мережу датчиків, розташованих у стратегічних точках, для збору даних про масу, розміри та розташування вантажів.



Рисунок 1.3 – Автономне завантаження літаків компанії Aurrigo

Алгоритм системи враховує різні параметри, такі як форма вантажів, їх вага, а також габаритні характеристики, для оптимального розподілу у відділеннях. Впровадження штучного інтелекту дозволяє системі самостійно адаптуватися до різноманітних видів вантажів та умов, що покращує її універсальність. Крім того, роботи взаємодіють з пілотом та граунд-контрольними системами для координації та забезпечення безпеки завантаження.

Така інноваційна система не лише прискорює процес завантаження, але й гарантує оптимальний центр ваги, що впливає на стабільність літака під час польоту. Патент свідчить про прагнення авіакомпанії до впровадження передових технологій у сфері авіаційної логістики.

Наступний патент є важливим кроком в напрямку сучасних інновацій у галузі авіаційної логістики. Він використовує передові методи навчання машин для покращення процесу завантаження вантажів на літак. Система вбудовує алгоритми машинного навчання, які навчаються на основі реальних даних та здатні прогнозувати оптимальний розподіл вантажів [22].

Автоматизовані рішення використовуються для аналізу маси, розмірів та характеристик кожного вантажу, що надає системі здатність самостійно адаптуватися до різноманітних сценаріїв завантаження. Застосування методів навчання машин дозволяє системі працювати на основі накопиченого досвіду та швидко вдосконалюватися відповідно до змінних умов.

Оптимізація розподілу вантажів враховує центр маси літака, забезпечуючи не лише ефективне завантаження, але й стабільність польоту. Система інтегрується з іншими авіаційними системами та забезпечує взаємодію для забезпечення безпеки та оптимальності всього процесу. Цей патент свідчить про стратегічний підхід до використання сучасних технологій для розвитку ефективних рішень у галузі авіаперевезень.

Патент від компанії Goodloading [23] визначає систему, яка автоматично адаптує конфігурацію вантажного відділення літака для оптимального розподілу ваги. За допомогою датчиків та моторизованих механізмів, вантажні відсіки можуть змінювати їх форму та розміри в реальному часі. Це забезпечує системі можливість динамічного реагування на зміни у завантаженні та підтримує оптимальний центр маси для безпечного польоту.

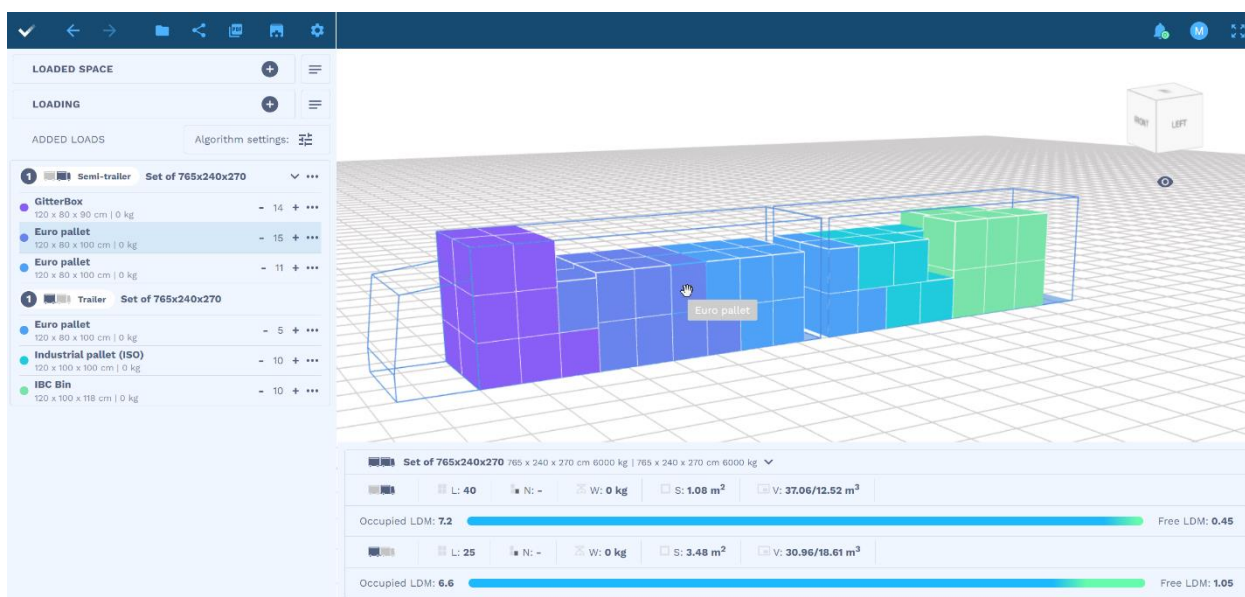


Рисунок 1.4 – Приклад інтерфейсу програмного продукту компанії Goodloading

Патенти підтверджують актуальність дослідження в галузі комп'ютерного керування завантаженням літаків. Алгоритми автоматизації, реального часу та використання штучного інтелекту вже є об'єктом патентних рішень. Однак, розробка алгоритмів, які б забезпечували ефективний розподіл вантажу з урахуванням центру маси, є актуальною та може стати інноваційним внеском в галузь авіаційної технології.

1.4 Постановка комплексу задач розробки систем керування завантаження літаків з урахуванням визначеного центру маси у площині вантажних відділень

Постановка комплексу задач розробки інтелектуальних програмних засобів контекстного розпізнавання військових літаків на супутникових Постановка комплексу задач для розробки системи керування завантаженням літаків з урахуванням визначеного центру маси у площині вантажних відділень охоплює наступні аспекти:

1. Аналіз існуючих підходів та вимог до завантаження літаків. Необхідно провести глибокий аналіз сучасних систем завантаження літаків, включаючи

технічні стандарти, вимоги безпеки, а також економічні аспекти авіаційних перевезень. Слід вивчити існуючі автоматизовані системи та алгоритми, які застосовуються для керування балансуванням і розподілом маси, та визначити їхні переваги і недоліки.

2. Формулювання математичної моделі розрахунку розташування вантажу з урахуванням центру маси. Потрібно розробити математичну модель, яка описує процес завантаження вантажу з урахуванням центру маси вантажного відділення. Модель має враховувати всі параметри вантажу: вагу, об'єм, можливі місця розташування, а також обмеження на допустиме навантаження та його розподіл по секціях вантажного відділення літака.

3. Розробка алгоритмів оптимізації завантаження. Основне завдання – створити ефективні алгоритми, що автоматизують процес розподілу вантажу з урахуванням центру маси. Алгоритми повинні забезпечувати оптимальний розподіл вантажу для підтримання стабільності та керованості літака під час польоту, а також мінімізацію витрат пального і часу на завантаження.

4. Інтеграція системи з іншими бортовими та наземними системами. Розроблена система повинна легко інтегруватися з існуючими бортовими системами контролю маси і балансування, а також з логістичними системами аеропортів. Це забезпечить автоматизовану взаємодію між різними елементами авіаційної інфраструктури і полегшить роботу персоналу.

5. Автоматизація процесу з використанням штучного інтелекту. Одним з ключових завдань є впровадження алгоритмів штучного інтелекту, які можуть аналізувати різноманітні варіанти розподілу вантажу, прогнозувати потенційні ризики, пов'язані з неправильною вагою, і самостійно адаптуватися до зміни вхідних параметрів. Це дозволить знизити вплив людського фактору і забезпечити більш точне балансування літака.

6. Оцінка економічної ефективності та безпеки. Ключове завдання – оцінити, як впровадження системи вплине на економічну ефективність авіаперевезень (зниження витрат на пальне, час завантаження, підвищення пропускної

спроможності) і на рівень авіаційної безпеки. Важливо також оцінити потенційні витрати на інтеграцію та обслуговування системи.

Цей комплекс задач охоплює весь процес створення системи керування завантаженням літаків, від аналізу та моделювання до впровадження й тестування, і є основою для підвищення ефективності авіаперевезень та забезпечення високого рівня безпеки польотів.

У висновку до першого розділу можна підсумувати, що ефективний розподіл маси є ключовим фактором забезпечення стабільності та безпеки польотів, зниження витрат на паливо та підвищення економічної ефективності авіаперевезень. Аналіз сучасних автоматизованих систем показав, що їхнє вдосконалення можливе завдяки використанню алгоритмів машинного навчання та штучного інтелекту, які мінімізують людський фактор та підвищують точність обчислень. Законодавче регулювання, яке базується на міжнародних стандартах IATA та ICAO, відіграє важливу роль у забезпеченні безпеки вантажних перевезень та формуванні вимог до автоматизованих систем. Окремо наголошено на необхідності інтеграції таких систем із іншими компонентами авіаційної інфраструктури, що сприяє їх ефективності та зручності у використанні. Стратегічний розвиток у цьому напрямі є критичним у контексті зростаючих обсягів авіаперевезень, зокрема в Україні та її регіонах, таких як Кривий Ріг, які активно інтегруються у міжнародні логістичні ланцюги. Важливим етапом дослідження є патентний пошук, що дозволяє визначити інноваційність підходів.

У майбутньому перспективи автоматизації включають впровадження нових технологій для підвищення ефективності, зниження витрат та покращення безпеки. Дискусійні аспекти щодо можливості повної заміни людського фактору в автоматизованих системах викликають інтерес серед науковців, але нині повна автономія видається малоімовірною. Водночас, подальша інтелектуалізація таких систем залишається актуальною задачею, особливо у контексті глобалізації та економічних викликів сучасності. Узагальнення світового досвіду свідчить про значний потенціал автоматизованих рішень у цій сфері, що підтверджується їхньою ефективністю та безпекою в реальних умовах експлуатації.

РОЗДІЛ 2

ІДЕНТИФІКАЦІЯ МАТЕМАТИЧНОЇ МОДЕЛІ ТА СИНТЕЗ СИСТЕМИ КЕРУВАННЯ ПРОЦЕСОМ

2.1 Розробка математичних моделей для систем керування завантаженням літаків

У даному розділі розглядаються основні етапи життєвого циклу математичної моделі процесу керування завантаженням літаків з урахуванням визначеного центру маси, що автоматизується. Першим етапом є ідентифікація об'єкта моделювання, що передбачає визначення основних характеристик і параметрів процесу завантаження, які будуть враховані при побудові моделі. На цьому етапі важливо чітко окреслити всі змінні, такі як маса вантажу, об'єм, розташування у вантажних відділеннях та їхній вплив на центрування і стабільність літака під час польоту. Від ідентифікації залежить вибір підходу до моделювання: формальний, алгоритмічний або графічний. Формальні моделі надають можливість описати процес у вигляді математичних рівнянь і співвідношень, алгоритмічні моделі дозволяють відтворити послідовність дій для ефективного розподілу вантажу, а графічні моделі забезпечують візуальне представлення процесу, що може використовуватися в документації або для полегшення розуміння системи.

Наступним важливим етапом є отримання результатів моделювання. Цей процес передбачає використання побудованої моделі для аналізу ефективності рішень, які приймаються в процесі завантаження літака. Отримані результати можуть включати оптимальний розподіл вантажу, який забезпечує мінімальне відхилення центру маси, а також оцінки економічних показників, таких як витрати на паливо або час завантаження. Використання формальних моделей дозволяє отримати аналітичні вирази для розрахунку основних характеристик процесу, алгоритмічні моделі можуть автоматизувати ці розрахунки, а графічні моделі роблять процес більш наочним, що полегшує його розуміння та подальше впровадження.

Результати моделювання стають основою для вирішення практичних завдань у сфері проектування, прогнозування та керування процесом завантаження. На основі отриманих даних можна розробляти нові стратегії завантаження літаків, що дозволяють оптимізувати процес не тільки в рамках одного рейсу, але й у більш глобальних сценаріях. Модель допомагає прогнозувати наслідки змін у характеристиках вантажу чи зовнішніх умовах і пропонувати оптимальні варіанти їх вирішення. Крім того, автоматизовані алгоритми, створені на базі математичної моделі, дозволяють оперативно реагувати на зміни в реальному часі, забезпечуючи тим самим стабільність і безпеку польоту.

Таким чином, життєвий цикл математичної моделі складається з кількох взаємопов'язаних етапів, починаючи від ідентифікації об'єкта моделювання, через отримання результатів, і закінчуючи їх використанням для вирішення завдань керування, прогнозування та оптимізації процесу завантаження літаків. Кожен з цих етапів має ключове значення для підвищення ефективності і безпеки роботи авіаційних систем, що в кінцевому результаті сприяє поліпшенню якості авіаперевезень.

У розділі, присвяченому ідентифікації математичної моделі та синтезу керування процесом завантаження літаків, однією з ключових характеристик моделі є її точність. Точність математичної моделі відображає, наскільки отримані за допомогою моделювання результати відповідають реальним даним. Для нашої теми, що стосується автоматизованого керування завантаженням літаків з урахуванням визначеного центру маси, точність є критичним показником, оскільки невірний розрахунок центру маси може вплинути на стійкість літака під час польоту, що загрожує безпеці.

При розробці комп'ютерної системи керування завантаженням основними факторами, що впливають на точність моделі, є правильність визначення маси та об'ємів вантажу, розташування кожного вантажного модуля у вантажних відсіках, а також відповідність моделі реальним фізичним властивостям літака. Для розрахунку точності можна використовувати показник середньоквадратичної похибки (RMSE), що вимірює відхилення між прогнозованими значеннями

(отриманими за допомогою моделі) та фактичними значеннями характеристик завантаження і центру маси літака.

Наприклад, нехай прогнозоване значення центру маси для певного вантажу складає 45% від довжини літака, а фактичне значення, отримане під час контрольних замірів після завантаження, становить 44,5%. Тоді похибка в цьому випадку складе 0,5%. Якщо така похибка є прийнятною з точки зору авіаційних стандартів і не впливає на стабільність польоту, можна вважати, що модель має достатню точність.

Якщо допустима похибка моделі для розрахунку центру маси становить, наприклад, 1-2% від загальної маси літака або його габаритів, то модель, яка демонструє похибку в межах цих значень, вважається достатньо точною. При цьому чим менша похибка, тим більш точною вважається модель. У разі, коли похибка перевищує допустимі межі, необхідно провести подальше коригування моделі, підвищивши її точність шляхом уточнення вихідних даних або поліпшення алгоритмів розрахунку.

Вірогідність математичної моделі, особливо у випадку автоматизованих систем керування завантаженням літаків, є важливим показником надійності побудованої моделі. Вірогідність характеризує ймовірність того, що модель буде функціонувати без помилок під час використання у реальних умовах. У цьому контексті необхідно враховувати всі можливі джерела помилок, що можуть виникнути на етапі побудови та застосування моделі, зокрема похибки при вимірюванні параметрів, неточності у розрахунках або помилки в алгоритмах.

Для математичної моделі керування завантаженням літака важливою є ймовірність того, що при обчисленні розподілу маси і центрування не буде допущено помилок, які могли б вплинути на стабільність літака під час польоту. Однак не всі помилки на локальному рівні призводять до загальної помилки моделювання. Наприклад, якщо завантаження вантажних відділень літака моделюється за допомогою структурних графів, і один із шляхів завантаження містить локальну помилку (наприклад, неточно визначене місце для окремого

контейнера), то загальна модель може залишатися коректною, якщо основні параметри, такі як центр маси, залишаються в межах допустимих значень.

Вірогідність математичної моделі може бути розрахована на основі аналізу ймовірностей виникнення помилок на кожному з етапів моделювання. Для цього необхідно врахувати такі фактори, як точність вимірювань вантажу, можливі збої в роботі датчиків або алгоритмів, а також особливості структури даних, що використовуються в моделі. В результаті, ймовірність помилки в моделі можна виразити через суму ймовірностей локальних помилок на різних етапах моделювання.

Однак у деяких випадках локальні помилки можуть не призвести до серйозних наслідків. Наприклад, якщо модель передбачає розподіл вантажу по декількох вантажних відділеннях і незначна похибка виникла лише в одному з них, загальний вплив на центр маси літака може бути мінімальним і не критичним для безпеки польоту. Таким чином, важливо проводити аналіз чутливості моделі до помилок і оцінювати, які локальні неточності можуть бути допустимими без шкоди для загальної вірогідності та надійності системи.

У підсумку, вірогідність моделі для комп'ютерної системи керування завантаженням літаків полягає в тому, наскільки ймовірно, що модель працюватиме без серйозних помилок при обчисленні оптимального розподілу маси. Це враховує можливі джерела похибок і мінімізує ризик їх впливу на загальну точність і адекватність моделювання.

Адекватність математичної моделі – це показник, що відображає, наскільки точно модель відтворює з необхідною повнотою всі важливі для дослідження властивості об'єкта. У випадку автоматизованої системи керування завантаженням літаків, адекватність полягає в тому, наскільки правильно модель описує процес завантаження з урахуванням центру маси та інших критичних параметрів, таких як вага вантажу, його розподіл по відсіках і вплив на стійкість літака під час польоту.

Оскільки будь-яка модель завжди спрощує оригінал, говорити про абсолютну адекватність неможливо. Модель завжди опускає деякі деталі або наближає їх до спрощених умов, які не завжди повністю відповідають реальним фізичним

процесам. Наприклад, у системі керування завантаженням можуть бути спрощені параметри аеродинамічних характеристик літака або взаємодія між окремими елементами вантажу, що розташовуються у різних відділеннях.

Оцінка адекватності моделі базується на порівнянні її результатів з фактичними даними, отриманими під час експериментів або в реальних умовах експлуатації. Це може бути досягнуто шляхом аналізу похибок між розрахунковими і фактичними значеннями, таких як центр маси чи загальна вага завантаження. Наприклад, якщо модель прогнозує центр маси з похибкою в 1% від реальних даних, її можна вважати адекватною для використання у відповідних умовах.

Ступінь адекватності також залежить від вимог до точності моделювання. Якщо модель демонструє достатній рівень точності для розв'язання конкретного завдання, вона може вважатися адекватною, навіть якщо в інших умовах її результати можуть відрізнятись від реальних. Таким чином, оцінювання адекватності полягає у визначенні допустимих меж відхилення результатів моделі від реальних характеристик об'єкта. Це дозволяє створити ефективну систему керування, яка буде функціонувати в рамках певних допущень та спрощень, необхідних для оптимізації процесу завантаження літаків.

Отже, адекватність моделі не є абсолютною характеристикою, а відображає ступінь її подібності до реального об'єкта, що оцінюється через показники точності або відмінності від оригіналу.

Таким чином, у цьому розділі були не лише розраховані основні характеристики математичної моделі (точність, вірогідність та адекватність), але й проаналізовані результати моделювання з точки зору їх відповідності реальним умовам. Для теми автоматизованого керування завантаженням літаків з урахуванням центру маси критично важливо досягти високої точності моделі, оскільки навіть незначні похибки можуть негативно вплинути на безпеку польоту.

2.2 Дослідження синтезу керування процесом систем керування завантаженням літаків

Дослідження синтезу керування процесом у системі керування завантаженням літаків з урахуванням визначеного центру маси у площині вантажних відділень є важливим етапом розробки автоматизованої системи, яка забезпечує безпечний та ефективний розподіл вантажу в літаку. Синтез керування процесом передбачає створення математичних та алгоритмічних моделей, які дозволяють оптимізувати процес завантаження, забезпечуючи баланс центру маси літака. Це необхідно для підтримки стійкості та керованості літака під час польоту, що має безпосередній вплив на його безпеку.

Основною метою синтезу керування є розробка таких алгоритмів, які, на основі вхідних даних про вагу, об'єм та розташування вантажу, можуть забезпечити оптимальне рішення для його розподілу по вантажних відділеннях. Ці алгоритми повинні враховувати не лише вагу та положення вантажу, але й інші важливі параметри, такі як тип літака, його аеродинамічні характеристики, експлуатаційні обмеження, та вимоги до безпеки польотів.

Синтез керування також передбачає розробку стратегій для вирішення задач компенсації зміщення центру маси у випадку нерівномірного розподілу вантажу або виникнення нештатних ситуацій, коли початковий план завантаження не може бути реалізований у повній мірі. Наприклад, якщо певний вантаж не може бути розташований у передбаченому місці через технічні або логістичні обмеження, система керування повинна автоматично перебудувати план завантаження таким чином, щоб мінімізувати негативний вплив на центрування літака.

Важливим етапом у синтезі керування є моделювання і тестування запропонованих рішень. За допомогою математичних моделей процесу завантаження можна проаналізувати, як різні сценарії розміщення вантажу впливають на загальну стійкість літака і його льотні характеристики. Алгоритми керування повинні бути протестовані в різних умовах, включаючи як ідеальні, так

і критичні сценарії, щоб переконатися в їх надійності і стійкості до зовнішніх впливів.

Загалом, синтез керування процесом завантаження літаків з урахуванням центру маси є комплексною задачею, яка вимагає точного моделювання, високої обчислювальної ефективності алгоритмів і гнучкості в прийнятті рішень. В результаті успішного дослідження та впровадження такої системи можна досягти значного підвищення безпеки польотів, ефективності використання вантажного простору літака та скорочення часу на виконання завантажувально-розвантажувальних операцій.

У розділі синтезу керування процесом системи керування завантаженням літаків важливим етапом є визначення бажаних динамічних характеристик системи та їх реалізація відповідно до обраного критерію оптимальності. Система керування повинна забезпечувати не лише точність і надійність керування, але й відповідати визначеним вимогам щодо швидкодії, стабільності та робастності. Для цього потрібно обрати критерій оптимальності, який буде використовуватися при синтезі керування.

Критерієм оптимальності у системі керування завантаженням літаків може бути мінімізація часу завантаження при одночасному забезпеченні оптимального розподілу маси з урахуванням центру ваги. Крім того, важливим показником може бути точність розподілу вантажу по відділеннях, яка забезпечить необхідні аеродинамічні характеристики літака під час польоту. Залежно від особливостей задачі, можливими критеріями також можуть бути мінімізація перевантаження на окремі відділення або економія енерговитрат на завантажувальні операції.

При синтезі керування необхідно враховувати передаточну та імпульсну перехідну функції системи, що дозволяють оцінити реакцію системи на вхідний сигнал і визначити її динамічні характеристики, такі як тривалість перехідного процесу, перерегулювання та точність у номінальному режимі. Для досягнення необхідних динамічних показників використовуються різні критерії оптимізації, серед яких особливо поширеним є інтегральний квадратичний критерій. Він

дозволяє мінімізувати відхилення системи від бажаного стану, забезпечуючи одночасно її стійкість та точність.

Однак при синтезі керування важливо враховувати не лише критерії оптимізації, але й обмеження, що накладаються фізичними характеристиками системи та її елементів. Наприклад, потужність завантажувальних механізмів може бути обмеженою, що ставить обмеження на швидкість виконання операцій або допустимі перевантаження. Крім того, необхідно враховувати можливі впливи, такі як коливання ваги вантажу, зміни в умовах навколишнього середовища або інші непередбачувані фактори, які можуть вплинути на роботу системи.

З огляду на це, система керування повинна бути робастною, тобто зберігати свою ефективність і стабільність за наявності таких збурювань або невизначеностей. Це вимагає додаткових механізмів компенсації та адаптації до змінних умов. Наприклад, якщо фактична вага вантажу відрізняється від запланованої, система повинна автоматично перебудувати план завантаження таким чином, щоб мінімізувати негативний вплив на центр маси літака і його стабільність.

Отже, результати синтезу керування повинні включати не лише визначення бажаних динамічних характеристик системи, але й реалізацію алгоритмів, які забезпечать виконання цих вимог з урахуванням критеріїв оптимізації та існуючих обмежень. Це дозволить забезпечити ефективну і безпечну роботу системи керування завантаженням літаків у реальних умовах експлуатації.

На першому етапі синтезу систем керування визначення оптимальних характеристик системи є ключовим завданням. Ці характеристики повинні враховувати як бажані динамічні показники (точність, швидкодія, стабільність), так і обмеження, що накладаються фізичними та технічними параметрами системи. Оптимальні характеристики визначаються на основі критерію оптимальності, такого як мінімізація часу перехідного процесу, максимальна точність або стійкість системи під дією зовнішніх збурень.

Проте ці оптимальні характеристики часто не можуть бути досягнуті в повному обсязі через практичні обмеження, такі як недостатня потужність

технічного обладнання, обмеженість ресурсів або недосконалість математичних моделей. Тому їх слід розглядати як теоретичну межу, до якої система повинна прагнути. Реальні системи керування повинні бути розроблені таким чином, щоб максимально наближатися до цих оптимальних характеристик, водночас враховуючи практичні обмеження.

Таким чином, процес синтезу систем керування передбачає не лише розробку моделей і алгоритмів для досягнення оптимальних параметрів, але й пошук компромісів між теоретичними вимогами і реальними можливостями системи.

Другий етап синтезу систем керування полягає у раціональній апроксимації оптимальних характеристик бажаними, які можна ефективно реалізувати на практиці. Це означає, що при розробці системи керування необхідно знайти баланс між точністю, складністю і надійністю. Хоча теоретично можна досягти оптимальних характеристик, на практиці вони часто потребують спрощення для забезпечення стабільної роботи системи, зменшення витрат на реалізацію і підтримку, а також підвищення її надійності.

Раціональна апроксимація дозволяє зберегти необхідну близькість до оптимальних умов, проте враховує реальні обмеження системи. В цьому контексті іноді завдання синтезу звужується до визначення корегувальних пристроїв, які можуть покращити функціональні властивості системи без суттєвих змін у її структурі. Це особливо актуально в випадках, коли система вже містить функціонально необхідні елементи, що реалізують конкретний спосіб керування, і завданням є лише оптимізація їх роботи.

Частковим завданням цього етапу синтезу є визначення параметрів системи при заданій її структурній схемі. Це може включати вибір коефіцієнтів підсилення, налаштування фільтрів, а також параметрів алгоритмів керування для досягнення бажаних динамічних характеристик. Завдяки такому підходу можна забезпечити простоту і надійність реалізації системи, не відхиляючись суттєво від оптимальних показників.

Завершальним етапом синтезу системи керування є детальний аналіз отриманої системи для перевірки її відповідності встановленим вимогам. Цей етап

може бути реалізований як розрахунковим, так і експериментальним шляхом. Одним з найбільш ефективних методів для проведення аналізу є використання комп'ютерного моделювання, яке було розроблено в попередньому розділі.

Під час аналізу системи перевіряються ключові динамічні характеристики, такі як точність, стабільність, швидкість реагування на зміни вхідних сигналів, а також її робастність до зовнішніх збурень. Моделювання дозволяє змоделювати різні сценарії роботи системи, включаючи граничні умови, і оцінити, як система справляється з потенційними проблемами, такими як зміщення центру маси або нерівномірний розподіл вантажу.

У разі, якщо результати аналізу виявляють невідповідність системи встановленим вимогам, можуть бути розроблені рекомендації щодо корекції параметрів або структури системи. Це може включати додаткові корегувальні пристрої, зміни в алгоритмах керування або адаптацію елементів системи для покращення її функціональних характеристик.

Таким чином, цей етап є критично важливим для забезпечення того, щоб система керування не лише відповідала теоретичним вимогам, але й була готова до практичного застосування, забезпечуючи ефективне та безпечне керування процесом завантаження літаків.

Математичне дослідження балансу вантажу для системи керування завантаженням літаків із врахуванням визначеного центру маси у площині вантажних відділень базується на основних фізичних принципах динаміки твердого тіла та розподілу мас. Літак розглядається як тверде тіло, яке має певний центр маси, важливий для забезпечення стійкості під час польоту. Завантаження вантажних відділень повинно бути таким, щоб загальний центр маси залишався в межах допустимого діапазону, визначеного технічними характеристиками літака. Центр маси повинен знаходитись в межах стабілізуючої площини літака для забезпечення належного управління під час польоту.

Центр маси завантаженого літака визначається через координати кожного вантажу та його масу. Відомі маси вантажів та їх координати дозволяють обчислити зважені середні значення координат, що і дають координати центра маси

літака. Основне завдання полягає в тому, щоб правильно розподілити вантажі таким чином, щоб центр маси залишався в допустимих межах, визначених для конкретного типу літака. Це завдання може бути сформульоване як задача оптимізації, де метою є мінімізація відхилення центра маси від його номінального значення, забезпечуючи відповідність технічним вимогам.

Для вирішення цієї задачі можуть бути використані чисельні методи, такі як лінійна або нелінійна оптимізація, що дозволяє знаходити оптимальний розподіл вантажу за заданими обмеженнями. Моделювання допоможе визначити оптимальні позиції для кожного вантажу, забезпечуючи баланс літака і дотримання всіх вимог до безпеки польоту. Окрім цього, точність моделі можна оцінити через порівняння результатів моделювання з фактичними даними про балансування вантажу, що дозволяє визначити похибки та адекватність моделі.

Математична модель балансування вантажу дозволяє ефективно керувати процесом завантаження літака і гарантує належний рівень безпеки під час польоту. Застосування таких моделей в автоматизованих системах керування значно підвищує ефективність процесу, знижуючи ризики, пов'язані з нерівномірним завантаженням літака.

Формули є ключовою частиною математичного дослідження балансу вантажу для системи керування завантаженням літаків. Основною задачею є визначення центру маси літака з урахуванням завантаження. Центр маси визначається як зважене середнє координат мас вантажу, розташованого у різних частинах вантажних відділень.

Рішення проблеми розподілу вантажу в літаку ґрунтується на забезпеченні того, щоб центр маси літака знаходився в межах допустимих діапазонів, визначених для конкретної моделі літака. Це є важливим для стабільності і керованості літака під час польоту. Для цього використовуються математичні моделі та методи оптимізації. Основою процесу є збір точних даних про кожен вантаж, включаючи його масу та координати місця розташування в літаку. Ці координати повинні лежати в межах допустимого діапазону, встановленого авіаційними стандартами для безпеки. Для вирішення цієї задачі використовуються

чисельні методи, такі як лінійна або нелінійна оптимізація, генетичні алгоритми або методи пошуку з табу. Залежно від методології, рішення може бути знайдене з використанням спеціалізованих програм, таких як MATLAB, Excel, Python або спеціалізовані авіаційні програми для завантаження вантажу. Після того, як знайдено оптимальний розподіл вантажу, проводиться моделювання роботи системи керування завантаженням літака. Для цього можна використовувати симуляційне моделювання, яке дозволить перевірити відповідність рішення авіаційним стандартам і вимогам безпеки.

Також можна врахувати різні сценарії розподілу вантажу, наприклад, нерівномірне завантаження або зміни в положенні вантажу через турбулентність. Якщо центр маси знаходиться поза допустимими межами, потрібно здійснити перерозподіл вантажу. Для цього можна перемістити окремі елементи вантажу в інші відділення літака з метою приведення центра маси у відповідність до вимог. Перерозподіл може бути проведений вручну або автоматизованою системою, яка пропонує варіанти завантаження. Оптимальний розподіл вантажу на борту літака — це критично важливе завдання, яке забезпечує безпечний і стабільний політ. Використання математичних моделей і методів оптимізації дозволяє вирішити цю проблему автоматизовано, мінімізуючи людські помилки та забезпечуючи точне керування вантажем.

Існує кілька поширених типів верхньопалубних ULD, форми контуру яких максимально підганяються під внутрішні обводи кабіни з метою збільшити використання корисного об'єму. Оскільки прямі зовнішні обводи (як у LD3 або LD6) призводять до деякої втрати об'єму, при формуванні контуру верхньопалубних вантажних пакетів створюються закруглені обводи, що оптимально підходять під контур завантаження кабіни. Верхньопалубні засоби пакування, аналогічно таким для нижнього відсіку, можуть бути повнорозмірними (займати всю ширину фюзеляжу і мати два обводи), так і половинними (мають один обвід і розміщуються попарно зліва або праворуч від осі поперечного перерізу літака, як, наприклад, контейнер AAX). Але на відміну від нижньопалубних, вони вищі і часто в 2-4 рази довші (див рис 2.3) [16].

777-200F Specifications

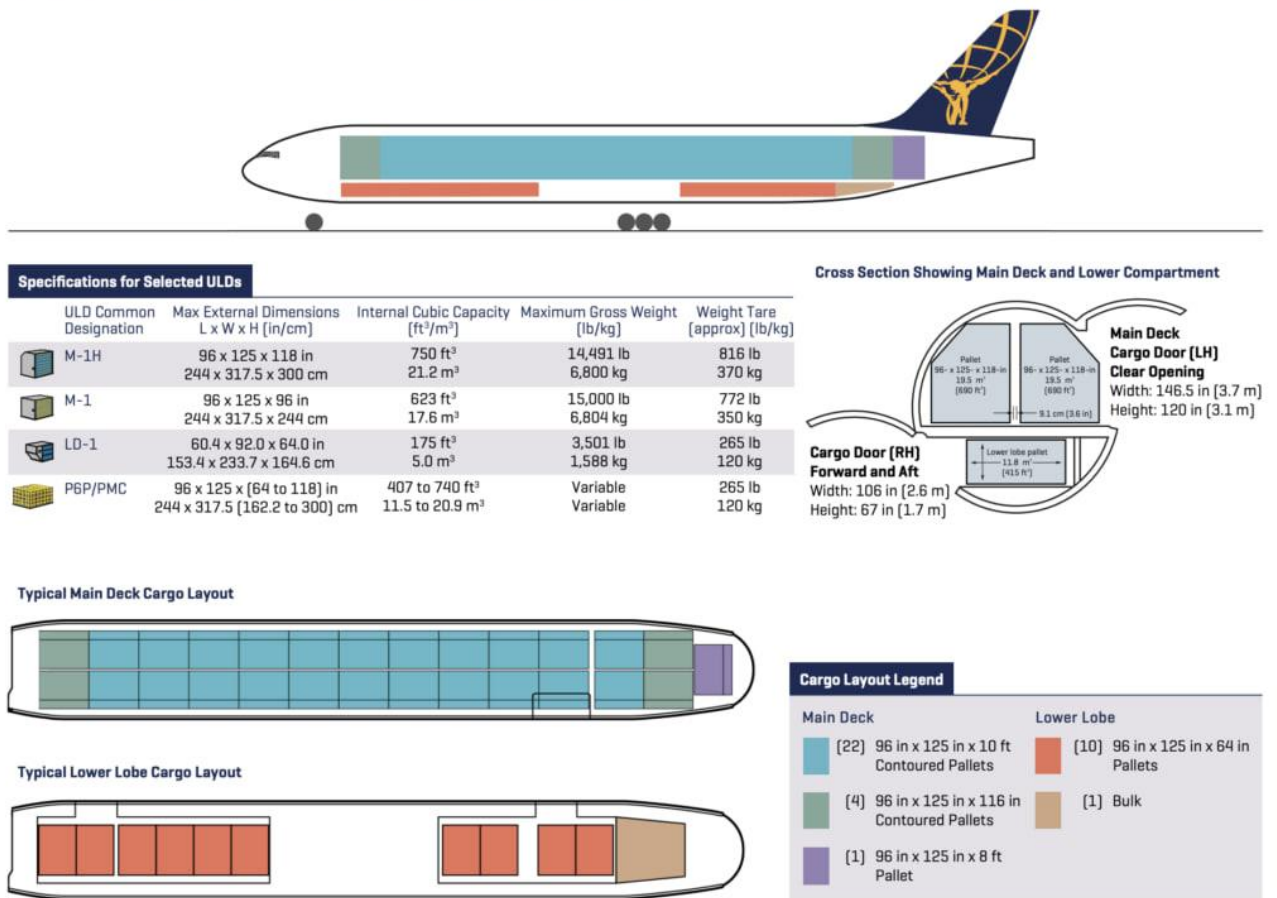


Рисунок 2.1 – Схема перерізу літака з верхньою та нижньою палубою

Багато транспортних компаній використовують так звані двопрофільні верхньопалубні ULD, які в невеликих літаках на кшталт Boeing 727 або 737 займають повну ширину фюзеляжу і мають два обводи, а у разі навантаження у більші літаки (наприклад, Boeing 767 або A300) повертаються на 90 градусів попарно. Це дозволяє суттєво спростити логістику за невеликих втрат корисного об'єму.

Обмеження ваги є одним з найважливіших обмежень для безпеки експлуатації літака. У пропонованій моделі вказані обмеження ваги позиції та загального корисного навантаження. Крім того, багато обмежень стосуються загальної ваги літака, позначеної (Weight). Це складається з трьох частин і є сумою експлуатаційної маси порожнього літака (OEW), його корисного навантаження (PL) і ваги палива (Fuel) [17] і обчислюється за формулою:

$$Weight = \sum_i W_i + OEW + Fuel \quad (2.1)$$

де W_i – вага завантаженого ULD i . виходячи з кількості завантаженого палива, загальну вагу можна розділити на три величини, включаючи злітну вагу (TOW), посадкову вагу (LW) і нульову вагу палива (ZFW), тоді як (TOF) є злітною вагою палива.

Центр ваги літака можна виразити як балансири (BA) або відсоткову середню аеродинамічну хорду (MAC). BA – це справжня міра в дюймах або метрах від контрольної точки, яку можна легко обчислити за допомогою рівняння:

$$BA_{CG} = \frac{\sum_i W_i \cdot BA_i}{\sum_i W_i}, \quad (2.2)$$

де BA_i – плече балансу елемента i , яке є відстанню від опорної точки до елемента і W_i – вага завантаженого контейнера.

У представлених формулах розглядається обмеження ваги літака та його вплив на безпеку і ефективність експлуатації. Загальна вага літака, позначена як (Weight), складається з трьох основних компонентів: ваги порожнього літака (OEW), ваги корисного навантаження (PL) і ваги палива (Fuel). Вага кожного завантаженого контейнера чи одиниці корисного навантаження позначається як W_i , і сумарна вага обчислюється через додавання ваг усіх завантажених елементів до (OEW) і ваги палива. Для забезпечення балансу літака важливо враховувати центрування, що описується через центр ваги (CG), який визначає рівновагу літака навколо його осі.

Центр ваги визначається за допомогою величини BA_i , яка є середньою величиною плечей балансу для всіх завантажених елементів, розрахованою через їх відстань від контрольної точки. Плече балансу BA_i кожного завантаженого елемента враховує його місце розташування щодо контрольної точки, а також вплив ваги цього елемента. Формула для визначення BA_i включає відношення добутку ваги та плечей балансу всіх елементів до їх загальної ваги.

Ці формули дозволяють враховувати вагу та центрування під час завантаження літака, забезпечуючи рівномірний розподіл ваги та правильне положення центру мас. Це є ключовим фактором для підтримання стабільності літака в польоті. Водночас враховуються такі важливі параметри, як злітна вага (TOW), посадкова вага (LW) і нульова вага палива (ZFW), які впливають на операційну ефективність та безпеку польоту. Таким чином, правильне розрахування ваги і центру мас гарантує відповідність нормам авіаційної безпеки та оптимізує експлуатацію літака.

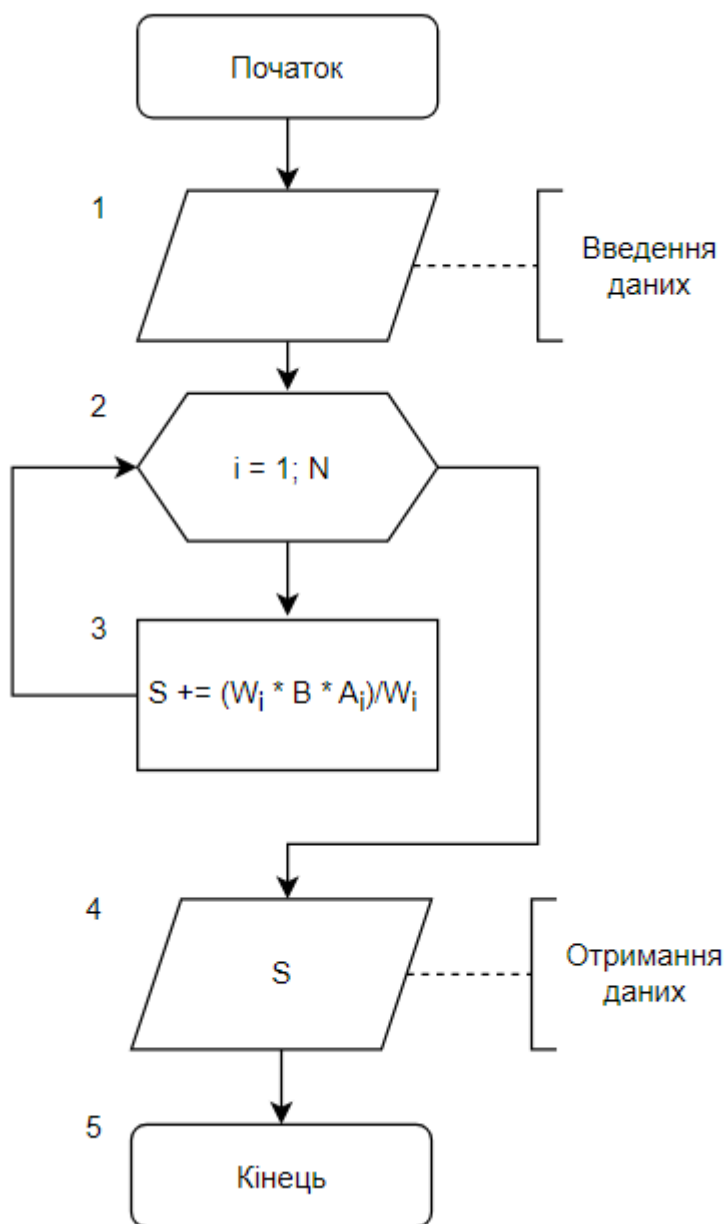


Рисунок 2.2 – Блок-схема алгоритму розрахунку центру ваги літака

Однак на практиці середня аеродинамічна хорда є більш зручною і широко використовуваною, ніж ВА. Вона визначається як відношення контрольної відстані крила, виміряної від передньої кромки МАС до розташування СГ, до довжини хорди (див. рис. 2.4). Співвідношення між ВА та %МАС наведено таким чином:

$$\%MAC = \frac{(BA_{CG} - l_{emac})}{l_{mac}} \times 100, \quad (2.3)$$

де l_{mac} – довжина середньої аеродинамічної хорди, а l_{emac} – довжина від точки відліку до переднього краю середньої аеродинамічної хорди.

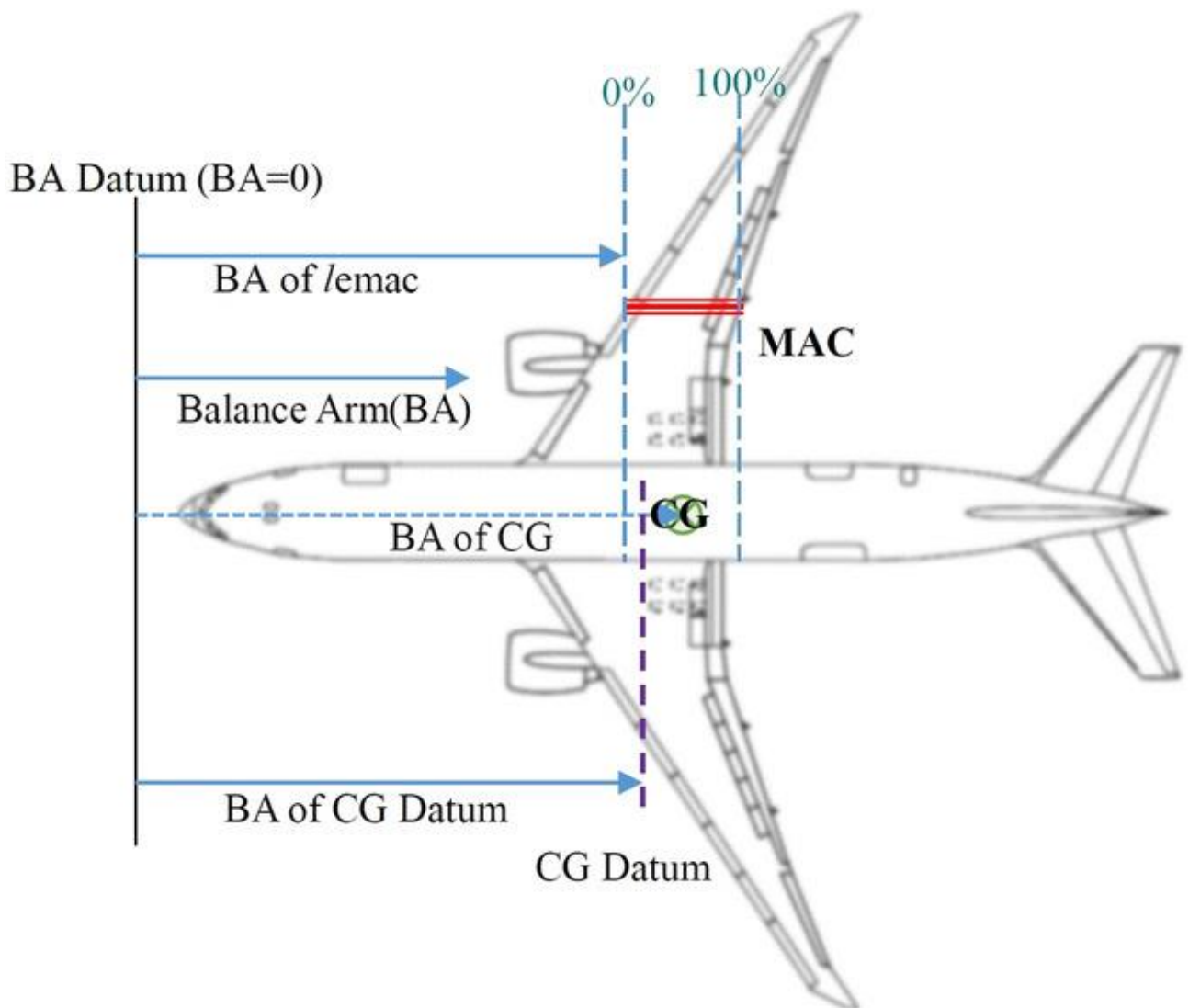


Рисунок 2.3 – Центр ваги літака

Методологія, що базується на інтернеті речей, дозволяє отримувати реальний час інформації про розташування вантажів, їхні параметри та стан, використовуючи датчики та засоби зв'язку. Система надає змогу оперативно реагувати на зміни в завантаженні та впливі на центр маси.

Однією з переваг є можливість автоматичного моніторингу вантажів, що дозволяє збільшити точність та швидкість процесу завантаження. Дані, зібрані в режимі реального часу, можуть використовуватися для аналізу та оптимізації процесів, що впливає на загальну ефективність авіап перевезень [18].

Недоліком може бути залежність від стабільного підключення до мережі Інтернету та висока вартість впровадження технологій IoT. Також, деякі аспекти безпеки і конфіденційності даних вимагають ретельного розгляду та заходів забезпечення.

Загалом, система на основі Інтернету речей є перспективною для вдосконалення процесів завантаження літаків, забезпечуючи більшу автоматизацію та оптимізацію роботи літака під час перевезення вантажів.

Наступним розглянемо алгоритм ефективного розподілу вантажу у вантажних відділеннях літака з метою забезпечення оптимального центру ваги для підтримки стабільності польоту. Автори визначають ключові фактори, які впливають на центр ваги, такі як маса вантажів, їхні розміри та розташування. Цей алгоритм використовує математичні моделі та оптимізаційні методи для врахування різноманітних параметрів, таких як маса, розмір та тип вантажів [19].

Процес врахування великої кількості факторів дозволяє враховувати специфічні вимоги різних видів літаків та різних місць вантаження. Застосування алгоритму дозволяє досягти оптимальної стабільності літака та покращити безпеку польотів.

Недоліком може бути висока обчислювальна складність алгоритму, особливо при великій кількості вантажів. Крім того, важливо враховувати динамічні зміни у розподілі вантажів під час польоту, що вимагає постійного оновлення алгоритму.

Загалом, алгоритм відображає інноваційний підхід до проблеми оптимального розподілу вантажів для забезпечення стабільності літака, що може

стати важливим етапом у розвитку систем управління завантаженням та підвищенні ефективності авіаперевезень [20].

Ці джерела слугують важливою основою для подальшого вдосконалення та розвитку комп'ютерної системи керування завантаженням літаків з урахуванням центру маси в площині вантажних відділень. На їх основі можна враховувати переваги та недоліки різних методів, а також впроваджувати інновації для досягнення оптимальної ефективності та безпеки у авіаційному транспорті.

Якість керування системою може бути оцінена за кількома ключовими показниками, які відображають її ефективність та стабільність. До таких показників належать:

1. Максимальне динамічне відхилення – це найбільше значення, на яке вихідна величина системи відхиляється від встановленого значення протягом перехідного процесу. Цей показник дозволяє оцінити, наскільки система здатна швидко реагувати на зміни в умовах управління.

2. Залишкове відхилення – це залишкове значення, яке система утримує після завершення перехідного процесу. Він показує, наскільки близько вихідна величина наближається до бажаного значення у стаціонарному режимі.

3. Час регулювання – це час, за який система досягає та залишається в межах певного відхилення від встановленого значення після початку управлінського впливу. Цей показник критично важливий для оцінки швидкості реакції системи.

4. Перерегулювання – це величина, на яку вихідна величина перевищує задане значення під час перехідного процесу. Високий рівень перерегулювання може свідчити про нестабільність системи або недостатнє налаштування її елементів.

5. Квадратичний інтегральний критерій якості – цей показник характеризує сумарну площу, обмежену кривою перехідного процесу, і дозволяє оцінити якість керування з урахуванням відхилень на всьому протязі часу. Він часто використовується для оптимізації систем керування, оскільки забезпечує комплексний підрахунок всієї інформації про динаміку системи.

Оцінка якості керування на основі цих показників дозволяє не лише зрозуміти, наскільки ефективно система виконує свої функції, але й виявити можливі недоліки та визначити шляхи для її вдосконалення. Це важливо для забезпечення надійності та безпеки, особливо у таких критичних галузях, як авіаційний транспорт.

У підсумку до другого розділу можна проаналізувати ключові аспекти життєвого циклу математичної моделі системи автоматизованого керування завантаженням літаків, зокрема її точність, вірогідність та адекватність. Було визначено, що процес моделювання включає ідентифікацію об'єкта, отримання результатів моделювання та їх застосування для вирішення практичних завдань. Чітка ідентифікація параметрів, таких як маса, об'єм вантажу, а також його розташування у вантажних відділеннях, є критичною для забезпечення точності моделі.

Точність моделі оцінюється через похибки розрахунків, наприклад, у визначенні центру маси літака. При цьому важливо досягти мінімальних похибок, щоб забезпечити безпеку польоту. Вірогідність моделі визначається ймовірністю коректного функціонування системи без серйозних помилок, навіть за наявності незначних локальних похибок. Адекватність моделі оцінюється через її здатність відображати ключові характеристики процесу завантаження, враховуючи необхідні спрощення для практичного застосування.

Було показано, що використання формальних, алгоритмічних та графічних моделей сприяє комплексному аналізу процесу. Формальні моделі забезпечують математичну точність, алгоритмічні дозволяють автоматизувати обчислення, а графічні роблять процес більш наочним. Це сприяє ефективному використанню моделі для розробки стратегій завантаження та прогнозування наслідків змін у зовнішніх умовах чи параметрах вантажу.

На основі отриманих результатів моделювання розробляються оптимальні стратегії завантаження, що забезпечують мінімальні відхилення центру маси, стабільність літака та економічну ефективність. Система автоматизованого

керування дозволяє оперативно реагувати на зміни в реальному часі, знижуючи ризики порушення безпеки польотів.

Процес синтезу керування передбачає створення моделей і алгоритмів, що оптимізують завантаження літаків. Розроблені системи забезпечують баланс між точністю, швидкістю та стабільністю, враховуючи обмеження технічних і ресурсних можливостей. Завдяки комп'ютерному моделюванню проводиться аналіз ефективності системи в різних умовах експлуатації, включаючи екстремальні сценарії.

У підсумку, створена математична модель є потужним інструментом для забезпечення ефективного і безпечного завантаження літаків. Вона дозволяє автоматизувати процес, підвищити його точність і знизити ризики, що має вирішальне значення для сучасних авіаційних систем.

РОЗДІЛ 3

ПРОГРАМНО-ТЕХНІЧНА РЕАЛІЗАЦІЯ СИСТЕМИ КЕРУВАННЯ ПРОЦЕСОМ

3.1 Розробка структурної і функціональної моделей програмного комплексу систем керування завантаженням літаків

Опис програмно-технічної реалізації системи керування завантаженням літаків передбачає створення структурної схеми, яка відображає основні компоненти системи, їх функції та взаємодію. Така схема допомагає зрозуміти призначення системи, її роботу в основних режимах і зв'язки між її елементами (див. рис. 3.1).

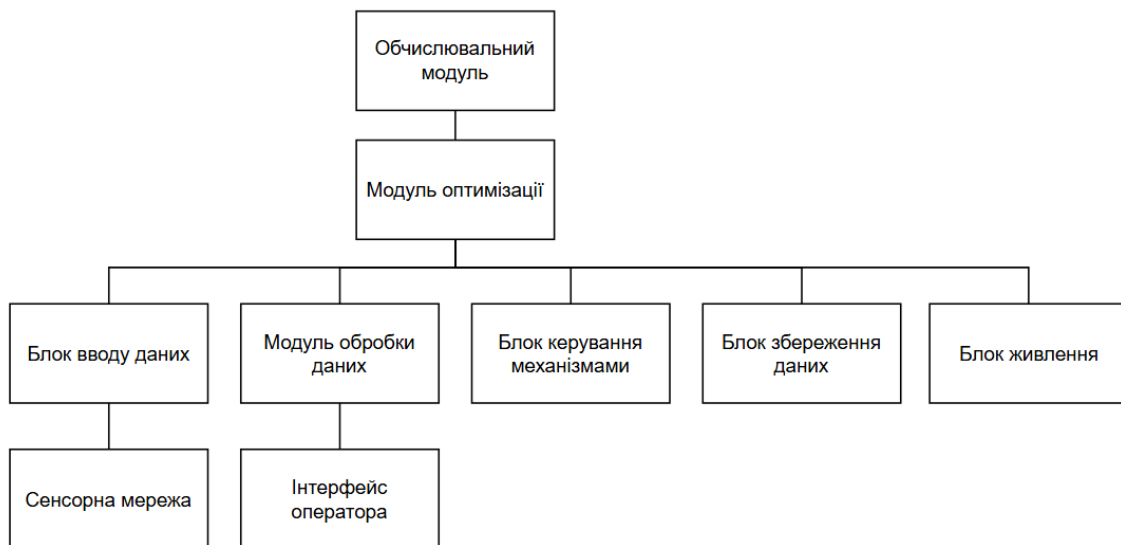


Рисунок 3.1 – Структурна схема системи

Система складається з кількох основних функціональних блоків. Центральним елементом є обчислювальний модуль, який виконує основні обчислення та координує роботу інших частин системи.

Ключовим структурним блоком є модуль оптимізації, який на основі алгоритмів розрахунку центру маси та оптимального розподілу вантажу визначає,

чи відповідає поточне завантаження безпечним умовам польоту. Цей модуль також генерує пропозиції щодо зміни розташування вантажу в разі, якщо центр маси виходить за допустимі межі. До нього підключено блок вводу даних, що приймає інформацію про характеристики вантажу (маса, габарити, тип) та його координати у вантажних відділеннях.

Для забезпечення безперервного збору даних від датчиків на борту літака використовується сенсорна мережа. Ця мережа складається з датчиків маси, положення, температури та інших характеристик вантажу. Датчики передають інформацію до обчислювального модуля через комунікаційний блок, який може бути реалізований як дротова або бездротова мережа (наприклад, через Wi-Fi або CAN-шину) [24].

Модуль обробки даних відповідає за перевірку коректності вхідної інформації, її фільтрацію та попередню обробку. На цьому етапі відбувається перевірка наявності помилок або некоректних значень у даних, що надходять із сенсорної мережі або блоку вводу даних.

Для візуалізації роботи системи використовується інтерфейс оператора, який включає дисплей або монітор для відображення статусу завантаження, попереджень і рекомендацій для перерозподілу вантажу. Інтерфейс також дозволяє оператору вручну вносити корективи у випадку несправності автоматичної системи.

Блок керування механізмами відповідає за передачу команд на актуатори або інші механічні пристрої, які можуть фізично переміщати вантаж у потрібне положення. Цей блок працює на основі сигналів, отриманих від модуля оптимізації.

Для забезпечення надійної роботи всієї системи використовується блок живлення, який забезпечує електроенергією всі компоненти системи, зокрема обчислювальний модуль, датчики та виконавчі пристрої. У разі перебоїв з основним живленням система переходить на резервний акумулятор.

Блок збереження даних зберігає інформацію про попередні розподіли вантажу, що дозволяє аналізувати ефективність роботи системи, прогнозувати можливі збої та генерувати звіти для авіакомпаній.

Взаємодія між цими блоками відбувається через центральну шину даних або локальну мережу, що забезпечує швидкий обмін інформацією між усіма компонентами.

Основний режим роботи системи передбачає збір даних про завантаження, їх обробку та оптимізацію розподілу вантажу. У разі виявлення проблем (наприклад, вихід центру маси за допустимі межі) система автоматично генерує попередження оператору та пропонує способи вирішення.

Структурна схема системи керування завантаженням літака відображає всі ключові компоненти, їх функціональні можливості та взаємодію, що дозволяє системі ефективно забезпечувати безпеку польотів та автоматизувати процеси завантаження. Вона включає блоки, відповідальні за вимірювання та контроль центру маси, моніторинг розподілу вантажу, а також управління процесом розподілу вантажу з урахуванням обмежень щодо маси та балансу. Взаємодія між компонентами здійснюється через централізовану обчислювальну платформу, що отримує дані з сенсорів та інших джерел інформації, що дозволяє здійснювати аналіз і приймати рішення в реальному часі. Завдяки цій схемі забезпечується оптимальний розподіл вантажу, що сприяє підвищенню ефективності експлуатації літака, зменшенню ризику пошкоджень і підвищенню безпеки польотів.

Отже, структурна схема системи керування завантаженням літака відображає всі ключові компоненти, їх функціональні можливості та взаємодію, що дозволяє системі ефективно забезпечувати безпеку польотів та автоматизувати процеси завантаження.

На рисунку 3.2 зображена функціональна схема комп'ютерної системи керування завантаженням літаків з урахуванням визначеного центру маси у площині вантажних відділень, що відображає процеси, які відбуваються в різних частинах системи, а також способи реалізації основних функцій і наочно описує поведінку системи та її станів під час роботи. Вона демонструє, як здійснюються інформаційні, математичні та фізичні перетворення, і які функціональні елементи забезпечують ці процеси.

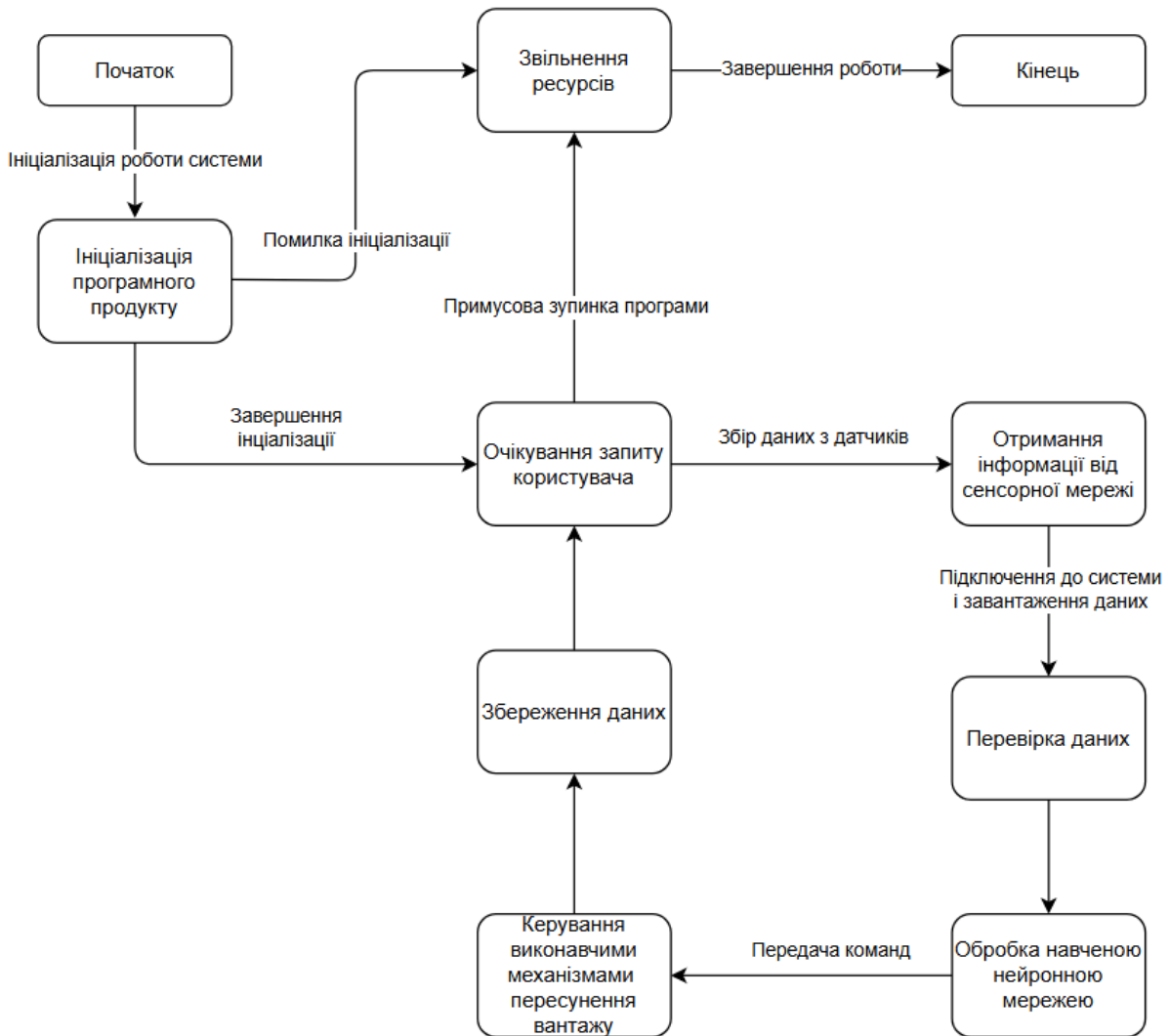


Рисунок 3.2 – Функціональна схема системи

Під час запуску програми відбувається ініціалізація роботи системи та проводиться ініціалізація стартового вікна програмного продукту. Після завершення ініціалізації комп'ютерна система чекає на запит користувача. При невдалому налаштуванні комп'ютерної системи керування завантаженням літаків з урахуванням визначеного центру маси у площині вантажних відділень клієнт отримує повідомлення про помилку і надається можливість повторити спробу. Якщо збір даних з сенсорних датчиків пройшов успішно, вони передаються на валідацію.

На наступному етапі модуль перевірки даних відповідає за фільтрацію, виявлення та корекцію помилок у вхідній інформації. Цей модуль розташований між модулем збору даних і обчислювальним модулем, щоб забезпечити достовірність даних для подальшої обробки. Він має вхід від модуля збору даних і вихід до обчислювального модуля.

Для розв'язання задач оптимізації розташування вантажу використовується модуль оптимізації, який розробляє стратегію розподілу вантажу з урахуванням центру маси, допустимих обмежень і алгоритму мінімізації дисбалансу розроблений на моделі за допомогою навченої нейронної мережі.

Після передачі команд керування виконавчими механізмами безпосередньо реалізуються зміни у розташуванні вантажу. Отримуються команди від обчислювального модуля через модуль оптимізації та передають їх на механізми, які можуть змінювати положення вантажу і потім відбувається перевірка виконання команд і контроль навантажень на механізми.

Для збереження даних та аналізу історії завантаження використовується модуль збереження даних, який забезпечує запис і доступ до інформації про попередні розподіли вантажу, а також результати роботи системи. Він підключений до обчислювального модуля через інтерфейс передачі даних.

На кінцевому етапі користування програмним комплексом перед завершенням роботи відбувається примусова зупинка серверу та звільнення ресурсів програми.

3.2 Розробка принципової схеми програмного комплексу систем керування завантаженням літаків

Принципова схема комп'ютерної системи керування завантаженням літаків з урахуванням визначеного центру маси у площині вантажних відділень детально описує елементи, зв'язки між ними та принципи функціонування системи. Вона розробляється на основі структурної та функціональної схем, враховуючи всі основні процеси та їх взаємодію.

На рисунку 3.3 в самому початку принципової схеми розташовується центральний процесорний модуль (CPU), який є мікроконтролером системи. Він реалізує алгоритми обробки даних, оптимізації та управління. Модуль має вхідні порти для отримання даних від сенсорів через аналого-цифрові перетворювачі (ADC), які відповідають за конвертацію аналогових сигналів у цифрову форму для обробки. Виходи процесорного модуля з'єднуються з інтерфейсом оператора, блоками візуалізації та управління виконавчими механізмами.

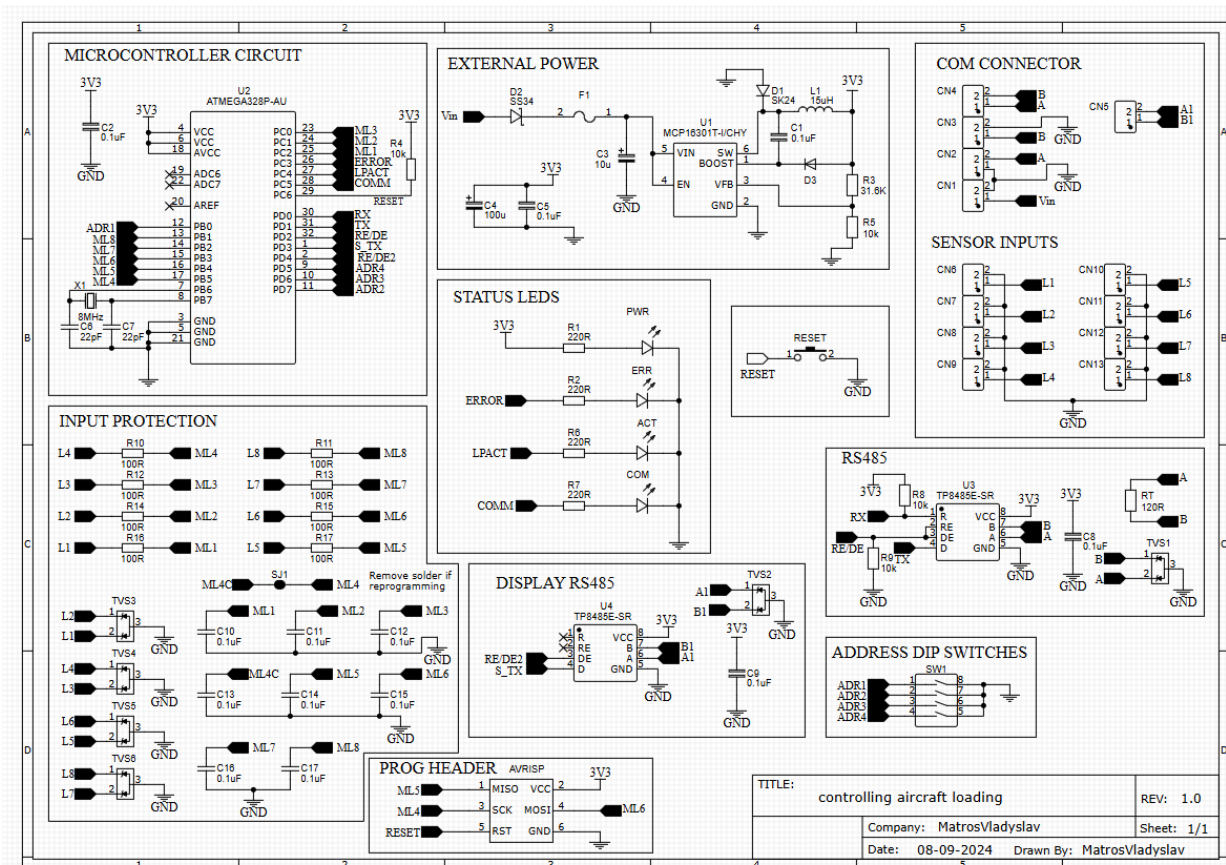


Рисунок 3.3 – Принципова схема системи

Для збору даних про параметри вантажу використовується модуль сенсорної мережі, який включає датчики ваги, положення і температури. Кожен датчик підключений до своєї шини даних через мультиплексор (MUX), який забезпечує ефективний збір даних. Датчики з'єднані з модулем збору даних, що оснащений фільтрами для обробки сигналів і зменшення шумів.

Обробка даних починається у модулі перевірки даних, який інтегрує перевірку точності, фільтрацію та нормалізацію. Він підключений до сенсорної

мережі та центрального процесорного модуля через інтерфейс зв'язку I2C або SPI, що забезпечує швидкий і надійний обмін даними [25].

Інтерфейс оператора представлений блоками введення (клавіатура, сенсорний екран) і виведення (LCD-дисплей). Він має прямий зв'язок із центральним модулем і забезпечує інтерактивний доступ до системи, дозволяючи оператору контролювати процес завантаження, задавати вхідні параметри та отримувати рекомендації.

Для оптимізації розташування вантажу використовується модуль оптимізації, який взаємодіє з центральним процесорним модулем. У цьому блоці реалізовано програмно-апаратні алгоритми, які розраховують центр маси, мінімізують відхилення та визначають оптимальну конфігурацію завантаження.

Для управління фізичним переміщенням вантажу застосовується блок виконавчих механізмів. Він включає серводвигуни, які отримують сигнали від процесора через контролери PWM (широкоімпульсна модуляція). Цей блок також оснащений датчиками зворотного зв'язку, що дозволяють контролювати виконання команд.

Додатково система містить блок живлення, який забезпечує стабільну подачу енергії до всіх компонентів. Він оснащений захистом від перевантаження та короткого замикання. Для збереження даних і запису історії використовується пам'ять типу EEPROM або флеш-пам'ять, інтегрована з процесором.

З'єднання між модулями реалізовано за допомогою стандартних роз'ємів, позначених на схемі відповідними позиційними позначеннями (наприклад, J1 для входів живлення, J2 для підключення датчиків). Усі елементи мають унікальні ідентифікатори, які відображають їх тип і функціональність, наприклад, R1 для резисторів, C1 для конденсаторів, U1 для мікросхем.

Принципова схема демонструє, як кожен компонент взаємодіє з іншими, і пояснює принцип роботи системи у нормальних і аварійних режимах. Вона забезпечує чітке уявлення про механізми управління, оптимізації та контролю завантаження, що робить її важливою частиною проектування системи.

Також можна інтегрувати технологію LoRa в комп'ютерну систему керування завантаженням літаків. Технологія LoRa (Long Range) є ефективним рішенням для створення малопотужних мереж із великим радіусом дії, що ідеально підходить для завдань автоматизації в авіаційній галузі. Інтеграція LoRa в комп'ютерну систему керування завантаженням літаків може значно підвищити її функціональність, зручність і точність [26].

Ключовою перевагою LoRa є її здатність забезпечувати стійкий зв'язок на великих відстанях із мінімальним енергоспоживанням. У контексті управління завантаженням літаків ця технологія дозволяє з'єднати різноманітні сенсори, вагові пристрої та контролери в єдину інтелектуальну мережу. Наприклад, усі пристрої, які вимірюють вагу вантажу, його розташування або стан, можуть безперервно передавати дані в реальному часі на центральну систему.

Для взаємодії оператора з цією системою можна використовувати планшет, оснащений програмним забезпеченням, яке враховує специфіку завантаження. Користувач вводить дані про тип і вагу вантажу, після чого програма розраховує оптимальне місце його розташування в літаку, враховуючи центр маси та інші параметри безпеки.

Завдяки LoRa планшет отримує дані від сенсорів у вантажному відсіку в реальному часі, що дозволяє оцінювати правильність розміщення вантажу та відстежувати зміни. Після підтвердження розташування вантажу користувач може одним натисканням надсилати сигнал на інші пристрої системи, які автоматично перевіряють стан і фіксацію вантажу.

Ще однією перевагою інтеграції LoRa є можливість роботи системи в умовах обмеженої мережевої інфраструктури. Завдяки високій енергоефективності датчики та планшет можуть функціонувати протягом тривалого часу без необхідності частого заряджання.

Використання LoRa також підвищує мобільність оператора. Планшет дозволяє виконувати завдання навіть на значній відстані від літака, що знижує необхідність прямої фізичної присутності в зоні завантаження. Крім того, за допомогою хмарного з'єднання система може автоматично синхронізуватися з

іншими елементами аеропорту, наприклад із системами обслуговування літаків або відстеження польотів. Ця інтеграція також сприяє підвищенню рівня безпеки. Наприклад, у випадку перевантаження система може автоматично надсилати попередження оператору, пропонуючи альтернативні варіанти перерозподілу вантажу.

Таким чином, технологія LoRa не лише забезпечує ефективний зв'язок між пристроями, але й створює основу для розвитку інтелектуальних рішень у галузі авіації. Її застосування в комп'ютерних системах керування завантаженням літаків відкриває нові можливості для автоматизації процесів, підвищення точності й зменшення витрат. Інтеграція таких інновацій може стати важливим кроком у вдосконаленні сучасних авіаційних систем, що відповідатимуть вимогам безпеки та ефективності.

На основі даної технології можливо створити планшет для оператора комп'ютерною системою керування завантаженням літаків з урахуванням визначеного центру маси у площині вантажних відділень, який інтегрований у систему керування завантаженням літаків із використанням технології LoRa, і є ключовим елементом інтерфейсу між оператором і автоматизованою інфраструктурою. Цей пристрій виступає як мобільний центр управління, оснащений спеціалізованим програмним забезпеченням, яке дозволяє швидко й ефективно приймати рішення щодо розміщення вантажу. Планшет з'єднується з мережею сенсорів і контролерів через LoRa, що забезпечує стабільний зв'язок навіть на великих відстанях і за умов значної радіочастотної завантаженості. На екрані планшета оператор вводить початкові дані про вантаж: його вагу, розміри, тип і специфічні вимоги, такі як потреба у спеціальних умовах транспортування. Програма автоматично аналізує ці параметри, враховуючи модель літака, його конфігурацію та поточне завантаження.

Ключовою функцією планшета є візуалізація вантажного відсіку літака в реальному часі. На дисплеї відображається інтерактивна схема, де позначено вже розміщений вантаж і доступні зони для нового. Програмне забезпечення пропонує оптимальні місця для розташування вантажу, керуючись алгоритмами, що

враховують розподіл ваги, положення центру маси, а також забезпечують баланс і стабільність літака. Оператор може вручну скоригувати ці рекомендації, якщо це необхідно, після чого затверджені дані надсилаються через LoRa на сенсори й контролери, розташовані у вантажному відсіку.

Планшет також виконує функцію зворотного зв'язку. Він отримує дані від сенсорів про фактичну вагу і положення розміщеного вантажу, що дозволяє перевіряти відповідність запланованому завантаженню. У разі виявлення невідповідностей система генерує сповіщення і пропонує варіанти коригування. Завдяки використанню LoRa зв'язок між планшетом і сенсорами є стабільним навіть у складних умовах, таких як металеві конструкції вантажного відсіку або перешкоди на локації.

Програмне забезпечення планшета може бути інтегроване з іншими системами аеропорту, такими як системи управління логістикою, бази даних польотів і служби технічного обслуговування. Це дозволяє автоматично синхронізувати інформацію про завантаження з іншими відділами. Наприклад, у разі затримки завантаження через технічні проблеми система може оперативно повідомляти диспетчерів або персонал, відповідальний за технічне обслуговування. Ергономіка планшета адаптована для використання в польових умовах. Він має ударостійкий корпус, захист від пилу і вологи, що важливо в умовах активної діяльності в зоні аеропорту. Сенсорний екран оптимізований для роботи в рукавичках, а інтерфейс програми розроблений так, щоб мінімізувати кількість дій, необхідних для виконання завдань. Крім того, планшет може бути оснащений модулем для роботи в автономному режимі, що дозволяє продовжувати управління завантаженням навіть у разі тимчасових перебоїв зв'язку.

Отже, планшет для оператора системою керування із технологією LoRa забезпечує інтеграцію розумної мережі пристроїв у процес управління завантаженням, підвищує точність і безпеку, а також зменшує час на виконання операцій. Це робить його незамінним інструментом для сучасної авіаційної логістики.

3.3 Розробка програмних класів та модулів програмного проекту

Для розробки комп'ютерної системи керування завантаженням літаків з урахуванням визначеного центру маси у площині вантажних відділень та машинного навчання моделі для нього було обрано мову програмування Python. Python є однією з найпопулярніших мов програмування, оскільки вона має відкритий вихідний код, проста у використанні та вивченні та дуже потужна для додатків машинного навчання. Усі останні досягнення в галузі машинного та глибокого навчання досягаються за допомогою бібліотек, написаних на Python. Тому вона використовується як основна мова програмування для написання всіх видів алгоритмів, коду та сценаріїв [27].

Існують інші мови програмування, які також можуть бути використані для розробки комп'ютерної системи керування завантаженням літаків з урахуванням визначеного центру маси у площині вантажних відділень, одна з яких, наприклад, мова програмування R. Однак вона має більш складний синтаксис, ніж Python, що може ускладнити розробку та підтримку програмних систем. R – це мова програмування, яка спеціально розроблена для статистичного аналізу та машинного навчання. Вона має велику бібліотеку статистичних функцій та інструментів машинного навчання, але не є такою гнучкою, як Python, і може бути менш придатною для розробки комплексних програмних систем.

У цілому, Python є хорошим вибором для розробки програмного комплексу керування завантаженням літаків з урахуванням визначеного центру маси у площині вантажних відділень, завдяки його ряду переваг, які роблять його корисним для цієї задачі, включаючи простоту та читабельність коду, гнучкість та масштабованість, доступність бібліотек та інструментів машинного навчання, а також безкоштовну вартість.

В даній розробленій програмі клас `AircraftCargoManagerAI` призначений для управління завантаженням літаків, забезпечуючи баланс і оптимальний розподіл вантажу. У процесі ініціалізації об'єкт отримує список секцій вантажних відділень, де кожна секція має назву, координати у площині вантажного відділення та

максимальну місткість. Ці дані зберігаються в полі `self.sections`, а початкове завантаження секцій ініціалізується як словник з нульовими значеннями для кожної секції. Також задається бажане положення центру маси, яке необхідно підтримувати для безпечного польоту.

Метод `add_cargo` відповідає за додавання вантажу в обрану секцію, перевіряючи, чи не перевищує новий вантаж її місткість. Якщо перевищення є, метод викидає помилку, попереджаючи про недопустиме завантаження. У випадку, якщо задана секція не знайдена, також викидається помилка. Це дозволяє уникнути некоректних дій у програмі (див. рис. 3.4). Для розрахунку центру маси використовується метод `calculate_center_of_mass`, що визначає координати центру маси, використовуючи зважену суму координат усіх завантажених секцій, поділену на загальну масу. Якщо у секціях немає вантажу, метод повертає значення `None`, сигналізуючи про відсутність центру маси. Для визначення найкращої секції для нового вантажу використовується метод `recommend_section_ai`, який застосовує модель машинного навчання для аналізу даних. Спочатку обчислюється поточний центр маси. Якщо вантажу немає, центр маси вважається у бажаній точці. Далі обчислюється доступний об'єм кожної секції як різниця між максимальною місткістю секції і поточним завантаженням.

Модель машинного навчання, навчена на історичних даних, передбачає оптимальну секцію для розміщення вантажу, враховуючи вагу, поточний центр маси та доступний об'єм. Результатом є назва рекомендованої секції. Теоретично, розподіл вантажу в літаку має критичне значення для збереження його стійкості та маневреності. Зміщення центру маси поза допустимі межі може спричинити аварійні ситуації, тому важливо підтримувати баланс у кожному етапі завантаження чи розвантаження.

Використання штучного інтелекту дозволяє враховувати складні фактори, зокрема нерівномірність розподілу чи зміни об'єму під час польоту. У цьому випадку застосовується модель `RandomForestRegressor`, що ефективно передбачає оптимальне розташування вантажу на основі історичних даних. Це сприяє

зменшенню ризиків, підвищенню точності та швидкості завантаження, а також забезпечує підтримання балансу літака протягом усього польоту.

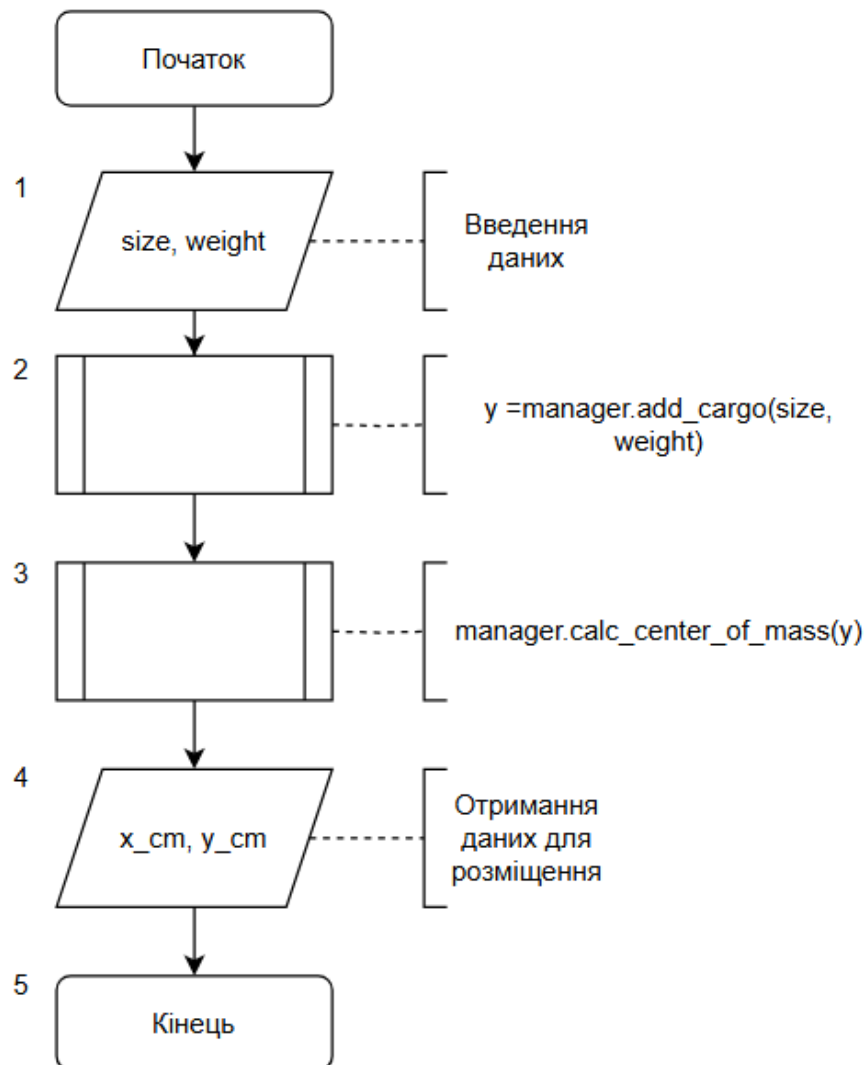


Рисунок 3.4 – Блок-схема алгоритму розрахунку рекомендованих координат для вантажу

Клас інтерфейсу у цій програмі відповідає за взаємодію користувача із системою управління завантаженням. Він забезпечує можливість вводити дані про вантаж, аналізувати поточний стан завантаження, отримувати рекомендації щодо розподілу та контролювати результати.

```

class AircraftCargoManager:
    def __init__(self, cargo_sections):
        self.sections = cargo_sections
        self.cargo_distribution = {section['name']: 0 for section
in cargo_sections}
  
```



```
self.desired_cm = (5, 1) # Бажане положення центру маси
```

Головна функція такого класу полягає в тому, щоб зробити систему інтуїтивно зрозумілою для користувача і водночас забезпечити зручний доступ до ключових функцій управління.

```
def add_cargo(self, section_name, weight):
    for section in self.sections:
        if section['name'] == section_name:
            if self.cargo_distribution[section_name] +
weight <= section['capacity']:
                self.cargo_distribution[section_name] +=
weight
            else:
                raise ValueError(f"Перевищено місткість для
відділення {section_name}!")
        return
    raise ValueError(f"Відділення {section_name} не
знайдено!")
```

Далі виконується обчислення центру маси для розподіленого навантаження в секціях та рекомендацію щодо найкращої секції для розміщення додаткового вантажу. Функція `calculate_center_of_mass` обчислює координати центра маси навантаження на основі поточного розподілу ваги. Спочатку змінні для сумарної ваги, зваженої суми координат по осі x і осі y ініціалізуються значенням нуля. Потім, у циклі, для кожної секції отримується її вага з поточного розподілу вантажу, і ця вага додається до загальної ваги. Одночасно виконується обчислення зважених сум координат x і y шляхом множення ваги на відповідні координати секції. Після завершення циклу перевіряється, чи загальна вага дорівнює нулю; у такому разі функція повертає значення `None`, щоб уникнути ділення на нуль. Інакше координати центра маси обчислюються шляхом поділу зважених сум на загальну вагу, і повертаються обчислені значення `x_cm` та `y_cm`.

```
def calculate_center_of_mass(self):
    total_weight = 0
    x_weighted_sum = 0
    y_weighted_sum = 0

    for section in self.sections:
        weight = self.cargo_distribution[section['name']]
        total_weight += weight
        x_weighted_sum += weight * section['x']
```

```

        y_weighted_sum += weight * section['y']

    if total_weight == 0:
        return None, None

    x_cm = x_weighted_sum / total_weight
    y_cm = y_weighted_sum / total_weight
    return x_cm, y_cm

```

Друга функція, `recommend_section`, призначена для пошуку найкращої секції, де можна розмістити додатковий вантаж із заданою вагою. Ініціалізуються змінні для відстеження найкращої секції і мінімальної відстані до центра маси. У циклі проходження по секціях перевіряється, чи не перевищить додавання нового вантажу максимальну допустиму місткість секції; якщо це станеться, поточна секція пропускається. Інші частини коду, ймовірно, відповідають за визначення, яка секція мінімізує зміщення центра маси, але їх немає в цьому фрагменті.

```

def recommend_section(self, weight):
    best_section = None
    min_distance = float('inf')

    for section in self.sections:
        if self.cargo_distribution[section['name']] + weight
> section['capacity']:
            continue

        # Симуляція додавання вантажу
        simulated_distribution =
self.cargo_distribution.copy()
        simulated_distribution[section['name']] += weight

        # Розрахунок нового центру маси
        total_weight = 0
        x_weighted_sum = 0
        y_weighted_sum = 0

        for s in self.sections:
            w = simulated_distribution[s['name']]
            total_weight += w
            x_weighted_sum += w * s['x']
            y_weighted_sum += w * s['y']

        if total_weight == 0:
            continue

        x_cm = x_weighted_sum / total_weight
        y_cm = y_weighted_sum / total_weight

```

Наступним відбувається визначення найкращої секції для розміщення додаткового вантажу на основі мінімізації відстані між фактичним центром маси та бажаним центром маси. Спочатку обчислюється відстань між поточним центром маси (координати `x_cm` та `y_cm`) і бажаним центром маси (заданий у властивості `self.desired_cm`). Для цього використовується формула евклідової відстані: з різниць відповідних координат обчислюються квадрати, які потім додаються, і з отриманої суми витягується квадратний корінь. Після цього виконується перевірка: якщо обчислена відстань є меншою за мінімальну відстань, яка була знайдена раніше, значення мінімальної відстані оновлюється до поточного, а змінна `best_section` оновлюється назвою секції, яка забезпечує це мінімальне значення. Наприкінці функція повертає назву секції, що найкраще підходить для розміщення нового вантажу, мінімізуючи відхилення центру маси від бажаного положення.

```
# Розрахунок відстані до бажаного центру маси
distance = ((x_cm - self.desired_cm[0])**2 + (y_cm -
self.desired_cm[1])**2)**0.5

        if distance < min_distance:
            min_distance = distance
            best_section = section['name']

    return best_section

# Інтерфейс
class CargoManagementApp:
    def __init__(self, root, manager, sensor_ports):
        self.manager = manager
        self.sensor_ports = sensor_ports
        self.root = root
        self.root.title("Cargo Management System")
        self.create_widgets()

        self.sensor_thread = threading.Thread(target=self.read_sensors)
        self.sensor_thread.daemon = True
        self.sensor_thread.start()
```

Далі створюється графічний інтерфейс користувача для відображення розподілу вантажу, центру маси та рекомендацій щодо завантаження. Спочатку створюється заголовок із текстом "Розподіл вантажу", який розміщується у верхній частині інтерфейсу в першому рядку та розтягується на дві колонки. Потім у циклі

для кожної секції, що зберігається у властивості `self.manager.sections`, створюються мітки, які відображають назву секції разом із її максимальною вантажопідйомністю. Мітки розташовуються у відповідному рядку першої колонки. Також для кожної секції створюється змінна `StringVar`, яка використовується для відображення поточного завантаження секції. Ця змінна зв'язується з міткою, розташованою в тій самій рядку, але в другій колонці. Далі створюється мітка для відображення поточного значення центру маси з текстом "Центр маси: ", яка розташовується під основним блоком міток у колонках, об'єднаних на весь рядок. Нарешті, створюється ще одна мітка для відображення рекомендації щодо завантаження з текстом "Рекомендація: ", яка розташовується під міткою центру маси, також об'єднуючи дві колонки. У результаті інтерфейс надає візуалізацію поточного стану завантаження та супутню інформацію.

```

def create_widgets(self):
    tk.Label(self.root,
              text="Розподіл
вантажу:").grid(row=0, column=0, columnspan=2)

    self.section_vars = {}
    row = 1
    for section in self.manager.sections:
        tk.Label(self.root, text=f"{section['name']} (макс
{section['capacity']} кг):").grid(row=row, column=0)
        var = tk.StringVar(value="0")
        self.section_vars[section['name']] = var
        tk.Label(self.root, textvariable=var).grid(row=row,
column=1)
        row += 1

    self.center_of_mass_label = tk.Label(self.root,
text="Центр маси: -")
    self.center_of_mass_label.grid(row=row + 1, column=0,
columnspan=2)

    self.recommendation_label = tk.Label(self.root,
text="Рекомендація: -")
    self.recommendation_label.grid(row=row + 2, column=0,
columnspan=2)

def update_display(self):
    for section in self.manager.sections:
        name = section['name']

```

```

self.section_vars[name].set(str(self.manager.cargo_distribution[name
]))

        x_cm, y_cm = self.manager.calculate_center_of_mass()
        if x_cm is None:
            self.center_of_mass_label.config(text="Центр маси:
Немає вантажу")
        else:
            self.center_of_mass_label.config(text=f"Центр маси:
x={x_cm:.2f}, y={y_cm:.2f}")

        recommended_section = self.manager.recommend_section(100) #
Наприклад, 100 кг вантажу
        if recommended_section:

self.recommendation_label.config(text=f"Рекомендація: завантажити у
{recommended_section}")
        else:
            self.recommendation_label.config(text="Рекомендація:
Немає місця для цього вантажу")

```

Цей код відповідає за оновлення відображення в графічному інтерфейсі користувача, відображаючи поточний стан розподілу вантажу, центр маси та рекомендації щодо завантаження. Спочатку в циклі обробляються всі секції, що зберігаються у властивості `self.manager.sections`. Для кожної секції зчитується її назва, і значення поточного завантаження, взяте з `self.manager.cargo_distribution`, оновлюється у відповідній змінній `StringVar`, яка пов'язана з графічною міткою. Далі викликається метод `calculate_center_of_mass`, який обчислює координати центру маси. Якщо значення центру маси відсутнє (наприклад, немає вантажу), мітка центру маси оновлюється текстом "Центр маси: Немає вантажу". У разі наявності значень координат центр маси відображається у форматі з двома знаками після коми.

Потім викликається метод `recommend_section` із фіксованим значенням ваги вантажу (наприклад, 100 кг), щоб визначити оптимальну секцію для його завантаження. Якщо рекомендована секція знайдена, відповідна мітка оновлюється текстом із вказівкою на цю секцію. Якщо місця для такого вантажу немає, мітка оновлюється текстом "Рекомендація: Немає місця для цього вантажу".

У результаті метод забезпечує динамічне оновлення інтерфейсу, відображаючи актуальну інформацію для користувача.

```
import tkinter as tk
from sklearn.ensemble import RandomForestRegressor
import numpy as np
import threading
import random

# Дані для навчання ШІ
# X: [вага, поточна x координата ЦМ, поточна y координата ЦМ,
# доступний об'єм у секціях]
# y: [оптимальна секція]
training_data_X = [
    [100, 5, 1, 500],
    [200, 4.8, 1.2, 450],
    [150, 5.1, 1.1, 400],
    [120, 5.2, 1.0, 480],
    [180, 4.9, 1.3, 460],
]
training_data_y = [0, 1, 1, 0, 2] # Секції: 0, 1, 2

# Навчання моделі
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(training_data_X, training_data_y)
```

Цей програмний код виконує підготовку до використання машинного навчання для прогнозування оптимальної секції для завантаження вантажу на основі даних про поточний стан системи. Спочатку імпортуються необхідні бібліотеки: `tkinter` для створення графічного інтерфейсу, `RandomForestRegressor` із бібліотеки `sklearn.ensemble` для побудови моделі машинного навчання, `numpy` для роботи з масивами, `threading` для багатопотоковості та `random` для роботи зі випадковими числами.

Дані для навчання моделі зберігаються у змінних `training_data_X` та `training_data_y`. У `training_data_X` кожен рядок представляє набір вхідних характеристик: вага вантажу, поточні координати центра маси по осях x і y , а також доступний об'єм у секціях літака. Змінна `training_data_y` містить мітки, які відповідають оптимальній секції для кожного набору вхідних характеристик. Значення секцій позначені номерами: 0, 1, 2 тощо.

Далі створюється модель `RandomForestRegressor`, яка є алгоритмом машинного навчання для задач регресії. Вона використовує ансамбль дерев рішень

(Decision Trees) для покращення точності прогнозування. Модель ініціалізується з параметром `n_estimators=100`, що означає використання 100 дерев рішень у регресійному лісі, та фіксованим `random_state=42`, що забезпечує відтворюваність результатів.

Викликається метод `fit`, який навчає модель на даних `training_data_X` та їхніх відповідях `training_data_y`. У результаті модель вивчає залежність між вхідними характеристиками та оптимальною секцією і буде використовувати цю інформацію для прогнозування, коли надаються нові дані.

Це базова частина системи штучного інтелекту, яка інтегрується для автоматичного прийняття рішень щодо оптимального розподілу вантажу в секціях літака.

```
# Клас для керування завантаженням
class AircraftCargoManagerAI:
    def __init__(self, cargo_sections):
        self.sections = cargo_sections
        self.cargo_distribution = {section['name']: 0 for section
in cargo_sections}
        self.desired_cm = (5, 1) # Бажане положення центру маси

    def add_cargo(self, section_name, weight):
        for section in self.sections:
            if section['name'] == section_name:
                if self.cargo_distribution[section_name] +
weight <= section['capacity']:
                    self.cargo_distribution[section_name] +=
weight
                else:
                    raise ValueError(f"Перевищено місткість для
відділення {section_name}!")
            return
        raise ValueError(f"Відділення {section_name} не
знайдено!")
```

Як зазначено вище, представляється клас `AircraftCargoManagerAI`, який використовується для управління розподілом вантажу в різних секціях літака. У методі `__init__` визначаються основні параметри об'єкта класу. У список `cargo_sections` передаються секції літака, кожна з яких має назву (ключ `'name'`) та максимальну вантажопідйомність (ключ `'capacity'`). Для кожної секції ініціалізується словник `cargo_distribution`, який відображає поточний стан

завантаження (початково 0 кг). Також задається бажане положення центру маси літака у вигляді координат (5, 1).

Метод `add_cargo` відповідає за додавання вантажу до певної секції літака. На вхід він приймає назву секції (`section_name`) та вагу вантажу (`weight`). У циклі перебираються всі секції літака, і якщо знайдено секцію з відповідною назвою, перевіряється, чи додавання ваги не перевищує її максимальної місткості (`capacity`). Якщо вагу можна додати, її додають до поточного завантаження секції у словнику `cargo_distribution`. У разі перевищення місткості секції викликається помилка `ValueError` із повідомленням про проблему. Якщо секція з указаною назвою не знайдена, також викликається `ValueError`, але вже з іншим повідомленням.

Цей клас дозволяє управляти завантаженням секцій літака, контролюючи як відповідність місткості, так і правильність введення назв секцій.

```
def calculate_center_of_mass(self):
    total_weight = 0
    x_weighted_sum = 0
    y_weighted_sum = 0

    for section in self.sections:
        weight = self.cargo_distribution[section['name']]
        total_weight += weight
        x_weighted_sum += weight * section['x']
        y_weighted_sum += weight * section['y']

    if total_weight == 0:
        return None, None

    x_cm = x_weighted_sum / total_weight
    y_cm = y_weighted_sum / total_weight
    return x_cm, y_cm

def recommend_section_ai(self, weight):
    x_cm, y_cm = self.calculate_center_of_mass()
    if x_cm is None or y_cm is None:
        x_cm, y_cm = self.desired_cm
```

Цей метод `recommend_section_ai` використовується для визначення оптимальної секції літака, куди можна розмістити вантаж за допомогою алгоритму, що враховує штучний інтелект. На вхід приймається параметр `weight`, який вказує вагу вантажу, що потрібно розподілити.

Спершу викликається метод `calculate_center_of_mass`, щоб отримати поточні координати центру маси вантажу літака (`x_cm` і `y_cm`). Якщо центру маси наразі немає (тобто вантаж у літаку відсутній), то ці значення замінюються на заздалегідь визначені бажані координати центру маси, що зберігаються в атрибуті `desired_cm`. Це гарантує, що навіть за відсутності початкових даних метод зможе працювати, використовуючи цільове значення центру маси як орієнтир для подальших обчислень.

```

        # Доступний об'єм
        available_volumes = [section['capacity'] -
self.cargo_distribution[section['name']] for section in
self.sections]

        # Передбачення найкращої секції
        prediction = model.predict([[weight, x_cm, y_cm,
min(available_volumes)]])
        best_section_index = int(round(prediction[0]))
        return self.sections[best_section_index]['name']

```

Клас, зазвичай, містить методи для створення графічного інтерфейсу, таких як кнопки, текстові поля, таблиці або графіки, які відображають статус завантаження та положення центру маси. Інтерфейс може включати такі функції: введення даних про вагу і вибір секції, що інтегрується з функціями класу `AircraftCargoManagerAI`, оновлення даних про стан завантаження у реальному часі, а також відображення поточного положення центру маси у вигляді графіка чи координат.

Користувач має можливість додавати новий вантаж, після чого система автоматично оновлює відображення розподілу. Якщо вантаж додається автоматично через сенсорний датчик, клас інтерфейсу також обробляє ці дані та передає їх у відповідні методи системи управління. У разі виникнення помилки, наприклад перевищення місткості секції, інтерфейс відображає зрозуміле повідомлення, що пояснює проблему.

Інтерфейс також відображає рекомендації, отримані від AI-моделі, що дозволяє користувачеві дізнатися, у яку секцію краще розмістити вантаж для підтримання центру маси в оптимальній зоні. Всі обчислення, включаючи центр

маси, виконуються у фоновому режимі і відображаються у зрозумілому вигляді для користувача.

Окрім того, клас інтерфейсу може надавати візуальні інструменти для аналізу даних: гістограми завантаження секцій, координати центру маси у вигляді точки на площині чи показники балансу у вигляді кольорових індикаторів. Користувач отримує можливість взаємодіяти із системою як через кнопки на екрані, так і через сенсори або інші пристрої вводу, якщо такі підключені.

```
# Оновлений інтерфейс
class CargoManagementAppAI:
    def __init__(self, root, manager):
        self.manager = manager
        self.root = root
        self.root.title("AI Cargo Management System")
        self.create_widgets()

    def create_widgets(self):
        tk.Label(self.root, text="Розподіл
вантаж:").grid(row=0, column=0, columnspan=2)

        self.section_vars = {}
        row = 1
        for section in self.manager.sections:
            tk.Label(self.root, text=f"{section['name']} (макс
{section['capacity']} кг):").grid(row=row, column=0)
            var = tk.StringVar(value="0")
            self.section_vars[section['name']] = var
            tk.Label(self.root, textvariable=var).grid(row=row,
column=1)
            row += 1

            self.center_of_mass_label = tk.Label(self.root,
text="Центр маси: -")
            self.center_of_mass_label.grid(row=row + 1, column=0,
columnspan=2)

            self.recommendation_label = tk.Label(self.root,
text="Рекомендація AI: -")
            self.recommendation_label.grid(row=row + 2, column=0,
columnspan=2)

            self.weight_entry = tk.Entry(self.root)
            self.weight_entry.grid(row=row + 3, column=0)
            tk.Button(self.root, text="Додати вантаж",
command=self.add_cargo).grid(row=row + 3, column=1)

    def add_cargo(self):
```

```

try:
    weight = int(self.weight_entry.get())
    recommended_section =
self.manager.recommend_section_ai(weight)
    self.manager.add_cargo(recommended_section, weight)
    self.update_display()
except ValueError as e:
    tk.messagebox.showerror("Помилка", str(e))

```

Метод `add_cargo` забезпечує додавання вантажу до літака через графічний інтерфейс. Спершу програма намагається отримати вагу вантажу з введення користувача через текстове поле `weight_entry`. Значення перетворюється на ціле число, оскільки очікується введення числових даних. Далі викликається метод `recommend_section_ai`, який використовує штучний інтелект для визначення оптимальної секції літака, куди слід завантажити вказану вагу. Якщо рекомендована секція знайдена, метод `add_cargo` додає вантаж у цю секцію, використовуючи внутрішній метод керування завантаженням літака, і після цього оновлює відображення даних у графічному інтерфейсі за допомогою функції `update_display`. У випадку, якщо виникає помилка (наприклад, перевищення місткості секції або неправильне введення), викликається повідомлення про помилку через вікно `tk.messagebox`, яке показує користувачеві відповідний текст помилки.

```

def update_display(self):
    for section in self.manager.sections:
        name = section['name']

self.section_vars[name].set(str(self.manager.cargo_distribution[name
]))

    x_cm, y_cm = self.manager.calculate_center_of_mass()
    if x_cm is None:
        self.center_of_mass_label.config(text="Центр маси:
Немає вантажу")
    else:
        self.center_of_mass_label.config(text=f"Центр маси:
x={x_cm:.2f}, y={y_cm:.2f}")

    # Рекомендація
    recommended_section =
self.manager.recommend_section_ai(100)
    if recommended_section:

```

```
self.recommendation_label.config(text=f"Рекомендація AI: завантажити  
у {recommended_section}")  
    else:  
        self.recommendation_label.config(text="Рекомендація AI:  
Немає місця для цього вантажу")
```

Клас інтерфейсу забезпечує зручний доступ до ключових функцій системи, надаючи інтерактивний і зрозумілий спосіб управління розподілом вантажу та контролю центру маси літака.

Таким чином, розроблена програма комп'ютерної системи керування завантаженням літаків, яка використовує штучний інтелект, інтеграцію з сенсорами та графічний інтерфейс, є ефективним і комплексним рішенням для оптимального управління вантажем. В результаті система дозволяє автоматизувати процес розподілу вантажу, забезпечуючи підтримання центру маси у заданій зоні, що є важливим для безпеки польотів і ефективної роботи авіаційної техніки. Завдяки автоматичним розрахункам положення центру маси у реальному часі виключається ризик помилок, спричинених людським фактором. Інтеграція з сенсорами дає змогу автоматично отримувати дані про масу та розташування вантажу, що значно прискорює процес і підвищує його точність.

Використання штучного інтелекту забезпечує прогнозування найкращих секцій для розміщення вантажу, оптимізуючи його розподіл і підтримуючи центр маси у бажаних межах. Код є масштабованим і адаптивним, що дозволяє його застосовувати для різних типів літаків і додаткових функцій, наприклад, врахування нестандартних вантажів з різною вагою. Саме тому система пропонує надійне, гнучке та зручне рішення, яке підвищує безпеку, ефективність і швидкість виконання операцій із завантаженням літаків.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було створено комп'ютерну систему керування завантаженням літаків з урахуванням визначеного центру маси у площині вантажних відділень.

У ході роботи було розроблено математичні моделі для визначення центру маси вантажу, а також алгоритми для оптимального розподілу вантажів з урахуванням обмежень і вимог до безпеки польотів. Проведено аналіз існуючих рішень у сфері управління завантаженням літаків, що дозволило обґрунтувати актуальність запропонованого підходу та визначити напрями для його вдосконалення.

Розроблена система використовує інтеграцію з сенсорами для отримання даних у реальному часі, а також алгоритми штучного інтелекту для рекомендацій щодо розташування вантажу, забезпечуючи високу точність і ефективність керування. Проведене тестування підтвердило відповідність розробленої системи вимогам до точності, адекватності та вірогідності, що дозволяє гарантувати її надійність у реальних умовах експлуатації.

Розроблена система може бути адаптована для використання на різних типах літаків, включаючи можливість подальшої інтеграції з іншими бортовими системами. У перспективі її можна вдосконалити за рахунок впровадження додаткових функцій, таких як врахування розподілу палива або прогнозування змін центру маси під час польоту. Система також має потенціал для розширення функціоналу шляхом використання більш складних алгоритмів штучного інтелекту і машинного навчання, що зробить її ще більш універсальною і надійною для авіаційної галузі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Rakesh Patel. Load Planning: Definition, Importance, Tips, and Beyond. [Електронний ресурс]. URL: <https://www.upperinc.com/guides/load-planning/>.
2. Low, R.; Tekler, Z.D.; Cheah, L. Predicting commercial vehicle parking duration using generative adversarial multiple imputation networks. *Transp. Res. Rec.* 2020, p. 820–831.
3. Brandt F. The Air Cargo Load Planning Problem. Fakultät für Wirtschaftswissenschaften des Karlsruher Instituts für Technologie (KIT). 2017, P. 39-57.
4. Авіаційний засіб пакування. [Електронний ресурс]. URL: https://ua.wikipedia.org/wiki/Авіаційний_засіб_пакування/.
5. Hong Na, H. T. and N. Nananukul (2016). Air Cargo Loading Management System for Logistics Forwarders. In: *Proceedings of 2016 International Conference on Urban Planning, Transport and Construction Engineering, Pattaya*, p. 51–58.
6. Dahmani, N. and S. Krichen (2016). Solving a load balancing problem with a multiobjective particle swarm optimisation approach: application to aircraft cargo transportation. *International Journal of Operational Research* 27 (1-2), p. 62–84.
7. Ceschia, S. and A. Schaerf (2013). Local search for a multi-drop multi-container loading problem. *Journal of Heuristics* 19 (2), p. 275–294.
8. Zhao X, Dong Y, Zuo L. A Combinatorial Optimization Approach for Air Cargo Palletization and Aircraft Loading. *Mathematics*. 2023. [Електронний ресурс]. URL: <https://doi.org/10.3390/math11132798>
9. Feng, B.; Li, Y.; Shen, Z.J.M. Air cargo operations: Literature review and comparison with practices. *Transp. Res. Part C Emerg. Technol.* 2015, p. 263–280.
10. Lurkin, V.; Schyns, M. The airline container loading problem with pickup and delivery. *Eur. J. Oper. Res.* 2015, p. 955–965.
11. Zhao X, Dong Y, Zuo L. A Combinatorial Optimization Approach for Air Cargo Palletization and Aircraft Loading. *Mathematics*. 2023. p. 25–31.

12. Brandt, F.; Nickel, S. The air cargo load planning problem—A consolidated problem definition and literature review on related problems. *Eur. J. Oper. Res.* 2019, p. 399–410.

13. CG deviation. [Електронний ресурс]. URL: <https://discovery.researcher.life/topic/integer-linear-programming-model/14449743?page=3>.

14. Your Guide to Unit Load Devices (ULD). [Електронний ресурс]. URL: <https://www.cargopoint.one/post/your-guide-to-unit-load-devices-uld/>.

15. Brandt, F.; Nickel, S. The air cargo load planning problem – A consolidated problem definition and literature review on related problems. *Eur. J. Oper. Res.* 2019, p. 101–109.

16. Unit load device. [Електронний ресурс]. URL: https://en.wikipedia.org/wiki/Unit_load_device/.

17. Оптимізаційний підхід до проблеми маси літака та балансування з центром ваги. [Електронний ресурс]. URL: https://web.archive.org/web/20220528041902id_/https://ietresearch.onlinelibrary.wiley.com/doi/pdfdirect/10.1049/itr2.12096.

18. Колісніченко А.В. Управління логістичними бізнес-процесами в умовах діджиталізації. Національний авіаційний університет. (2021) [Електронний ресурс]. URL: https://er.nau.edu.ua/bitstream/NAU/53577/1/ФТМЛ_2021_КолісніченкоАВ_208М.pdf.

19. Сагун Є. Розробка та візуалізація комп'ютерної моделі планування завантаження вантажного літака. (2021) [Електронний ресурс]. URL: https://www.academia.edu/69401362/Development_and_Visualization_of_the_Computer>Loading_Planning_Model_for_the_Cargo_Aircraft.

20. Вплив повздовжнього та поперечного зміщення центру ваги у вантажних літаках. [Електронний ресурс]. URL: https://ela.kpi.ua/bitstream/123456789/30940/1/Haluzinskyi_magistr.pdf/.

21. Weight distribution systems and control logic for center of gravity management of aircrafts. [Електронний ресурс]. URL: <https://patents.google.com/patent/US20210371087A1/en>.

22. Improving aircraft performance using machine learning. [Електронний ресурс]. URL: <https://patents.google.com/patent/US202215876581/en>.

23. Goodloading. [Електронний ресурс]. URL: <https://www.goodloading.com/en/about-the-program/>.

24. Controller Area Network [Електронний ресурс]. URL: https://uk.wikipedia.org/wiki/Controller_Area_Network/.

25. Інтерфейс I2C. [Електронний ресурс]. URL: <https://itmaster.biz.ua/directory/standarts/i2c.html>.

26. LoRa [Електронний ресурс]. URL: <https://uk.wikipedia.org/wiki/LoRa>.

27. Python 3.10.3 documentation. [Електронний ресурс]. URL: <https://docs.python.org/3/>.

27. Маринич І. А., Тронь В. В. Методичні рекомендації до виконання кваліфікаційної роботи магістра для студентів спеціальності 151 “Автоматизація та комп’ютерно-інтегровані технології”. Кривий Ріг : Видавничий центр КНУ, 2022. 50с.

28. ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура і правила оформлення. Київ, ДП «УкрННЦ», 2015. 26с. (Інформація та документація).

29. ДСТУ 8302:2015. Бібліографічне посилання. Загальні вимоги та правила складання Київ, ДП «УкрННЦ», 2016. 16 с. (Інформація та документація).

30. ДСТУ 3582:2013. Бібліографічний опис. Скорочення слів і словосполучень в українській мові. Загальні вимоги та правила. Київ, ДП «УкрННЦ», 2013. 23 с. (Інформація та документація).

31. ДСТУ 3651.0-97 Метрологія. Одиниці фізичних величин. Основні одиниці фізичних величин Міжнародної системи одиниць. Основні положення, назви та позначення. Київ, Держстандарт України, 1998. 27с. (Інформація та документація).

ДОДАТОК А

Модуль розробки комп'ютерної системи керування завантаженням літаків

```

class AircraftCargoManager:
    def __init__(self, cargo_sections):
        self.sections = cargo_sections
        self.cargo_distribution = {section['name']: 0 for section
in cargo_sections}
        self.desired_cm = (5, 1) # Бажане положення центру маси

    def add_cargo(self, section_name, weight):
        for section in self.sections:
            if section['name'] == section_name:
                if self.cargo_distribution[section_name] +
weight <= section['capacity']:
                    self.cargo_distribution[section_name] +=
weight
                else:
                    raise ValueError(f"Перевищено місткість для
відділення {section_name}!")
            return
        raise ValueError(f"Відділення {section_name} не
знайдено!")

    def calculate_center_of_mass(self):
        total_weight = 0
        x_weighted_sum = 0
        y_weighted_sum = 0

        for section in self.sections:
            weight = self.cargo_distribution[section['name']]
            total_weight += weight
            x_weighted_sum += weight * section['x']
            y_weighted_sum += weight * section['y']

        if total_weight == 0:
            return None, None

        x_cm = x_weighted_sum / total_weight
        y_cm = y_weighted_sum / total_weight
        return x_cm, y_cm

    def recommend_section(self, weight):
        best_section = None
        min_distance = float('inf')

        for section in self.sections:
            if self.cargo_distribution[section['name']] + weight
> section['capacity']:
                continue

```

```

        # Симуляція додавання вантажу
        simulated_distribution =
self.cargo_distribution.copy()
        simulated_distribution[section['name']] += weight

        # Розрахунок нового центру маси
        total_weight = 0
        x_weighted_sum = 0
        y_weighted_sum = 0

        for s in self.sections:
            w = simulated_distribution[s['name']]
            total_weight += w
            x_weighted_sum += w * s['x']
            y_weighted_sum += w * s['y']

        if total_weight == 0:
            continue

        x_cm = x_weighted_sum / total_weight
        y_cm = y_weighted_sum / total_weight

        # Розрахунок відстані до бажаного центру маси
        distance = ((x_cm - self.desired_cm[0])**2 + (y_cm -
self.desired_cm[1])**2)**0.5

        if distance < min_distance:
            min_distance = distance
            best_section = section['name']

        return best_section

# Інтерфейс
class CargoManagementApp:
    def __init__(self, root, manager, sensor_ports):
        self.manager = manager
        self.sensor_ports = sensor_ports
        self.root = root
        self.root.title("Cargo Management System")
        self.create_widgets()

        self.sensor_thread =
threading.Thread(target=self.read_sensors)
        self.sensor_thread.daemon = True
        self.sensor_thread.start()

    def create_widgets(self):
        tk.Label(self.root,
вантаж:").grid(row=0, column=0, columnspan=2)
        text="Розподіл

        self.section_vars = {}
        row = 1

```

```

        for section in self.manager.sections:
            tk.Label(self.root, text=f"{section['name']} (макс
{section['capacity']} кг):").grid(row=row, column=0)
            var = tk.StringVar(value="0")
            self.section_vars[section['name']] = var
            tk.Label(self.root, textvariable=var).grid(row=row,
column=1)

            row += 1

        self.center_of_mass_label = tk.Label(self.root,
text="Центр маси: -")
        self.center_of_mass_label.grid(row=row + 1, column=0,
columnspan=2)

        self.recommendation_label = tk.Label(self.root,
text="Рекомендація: -")
        self.recommendation_label.grid(row=row + 2, column=0,
columnspan=2)

    def update_display(self):
        for section in self.manager.sections:
            name = section['name']

self.section_vars[name].set(str(self.manager.cargo_distribution[name
]))

            x_cm, y_cm = self.manager.calculate_center_of_mass()
            if x_cm is None:
                self.center_of_mass_label.config(text="Центр маси:
Немає вантажу")
            else:
                self.center_of_mass_label.config(text=f"Центр маси:
x={x_cm:.2f}, y={y_cm:.2f}")

            recommended_section = self.manager.recommend_section(100) #
Наприклад, 100 кг вантажу
            if recommended_section:

self.recommendation_label.config(text=f"Рекомендація: завантажити у
{recommended_section}")
            else:
                self.recommendation_label.config(text="Рекомендація:
Немає місця для цього вантажу")

import tkinter as tk
from sklearn.ensemble import RandomForestRegressor
import numpy as np
import threading
import random

# Дані для навчання ШІ

```

```

# X: [вага, поточна x координата ЦМ, поточна y координата ЦМ,
доступний об'єм у секціях]
# y: [оптимальна секція]
training_data_X = [
    [100, 5, 1, 500],
    [200, 4.8, 1.2, 450],
    [150, 5.1, 1.1, 400],
    [120, 5.2, 1.0, 480],
    [180, 4.9, 1.3, 460],
]
training_data_y = [0, 1, 1, 0, 2] # Секції: 0, 1, 2

# Навчання моделі
model = RandomForestRegressor(n_estimators=10, random_state=42)
model.fit(training_data_X, training_data_y)

# Клас для керування завантаженням
class AircraftCargoManagerAI:
    def __init__(self, cargo_sections):
        self.sections = cargo_sections
        self.cargo_distribution = {section['name']: 0 for section
in cargo_sections}
        self.desired_cm = (5, 1) # Бажане положення центру маси

    def add_cargo(self, section_name, weight):
        for section in self.sections:
            if section['name'] == section_name:
                if self.cargo_distribution[section_name] +
weight <= section['capacity']:
                    self.cargo_distribution[section_name] +=
weight
                else:
                    raise ValueError(f"Перевищено місткість для
відділення {section_name}!")
                return
            raise ValueError(f"Відділення {section_name} не
знайдено!")

    def calculate_center_of_mass(self):
        total_weight = 0
        x_weighted_sum = 0
        y_weighted_sum = 0

        for section in self.sections:
            weight = self.cargo_distribution[section['name']]
            total_weight += weight
            x_weighted_sum += weight * section['x']
            y_weighted_sum += weight * section['y']

        if total_weight == 0:
            return None, None

        x_cm = x_weighted_sum / total_weight

```

```

y_cm = y_weighted_sum / total_weight
return x_cm, y_cm

def recommend_section_ai(self, weight):
    x_cm, y_cm = self.calculate_center_of_mass()
    if x_cm is None or y_cm is None:
        x_cm, y_cm = self.desired_cm

    # Доступний об'єм
    available_volumes = [section['capacity'] -
self.cargo_distribution[section['name']] for section in
self.sections]

    # Передбачення найкращої секції
    prediction = model.predict([[weight, x_cm, y_cm,
min(available_volumes)]])
    best_section_index = int(round(prediction[0]))
    return self.sections[best_section_index]['name']

# Оновлений інтерфейс
class CargoManagementAppAI:
    def __init__(self, root, manager):
        self.manager = manager
        self.root = root
        self.root.title("AI Cargo Management System")
        self.create_widgets()

    def create_widgets(self):
        tk.Label(self.root, text="Розподіл
вантажу:").grid(row=0, column=0, columnspan=2)

        self.section_vars = {}
        row = 1
        for section in self.manager.sections:
            tk.Label(self.root, text=f"{section['name']} (макс
{section['capacity']} кг):").grid(row=row, column=0)
            var = tk.StringVar(value="0")
            self.section_vars[section['name']] = var
            tk.Label(self.root, textvariable=var).grid(row=row,
column=1)
            row += 1

        self.center_of_mass_label = tk.Label(self.root,
text="Центр маси: -")
        self.center_of_mass_label.grid(row=row + 1, column=0,
columnspan=2)

        self.recommendation_label = tk.Label(self.root,
text="Рекомендація AI: -")
        self.recommendation_label.grid(row=row + 2, column=0,
columnspan=2)

        self.weight_entry = tk.Entry(self.root)
        self.weight_entry.grid(row=row + 3, column=0)

```

```

tk.Button(self.root, text="Додати вантаж",
command=self.add_cargo).grid(row=row + 3, column=1)

def add_cargo(self):
    try:
        weight = int(self.weight_entry.get())
        recommended_section =
self.manager.recommend_section_ai(weight)
        self.manager.add_cargo(recommended_section, weight)
        self.update_display()
    except ValueError as e:
        tk.messagebox.showerror("Помилка", str(e))

def update_display(self):
    for section in self.manager.sections:
        name = section['name']

self.section_vars[name].set(str(self.manager.cargo_distribution[name
]))

        x_cm, y_cm = self.manager.calculate_center_of_mass()
        if x_cm is None:
            self.center_of_mass_label.config(text="Центр маси:
Немає вантажу")
        else:
            self.center_of_mass_label.config(text=f"Центр маси:
x={x_cm:.2f}, y={y_cm:.2f}")

        # Рекомендація
        recommended_section =
self.manager.recommend_section_ai(100)
        if recommended_section:

self.recommendation_label.config(text=f"Рекомендація AI: завантажити
у {recommended_section}")
        else:
            self.recommendation_label.config(text="Рекомендація AI:
Немає місця для цього вантажу")

```