

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**ПрАТ «ПВНЗ «ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»****Кафедра Економічної кібернетики та інженерії програмного
забезпечення**

ДО ЗАХИСТУ ДОПУЩЕНА
Зав.кафедри _____
д.е.н., доцент Левицький С.І.

БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА**РОЗРОБКА СИСТЕМИ ОПТИМІЗАЦІЇ СПОЖИВАННЯ ПАЛИВА НА
ОСНОВІ ГЕНЕТИЧНИХ АЛГОРИТМІВ**

Виконав
ст. гр. ІПЗ-217

(підпис)

В.Д. Гриб

Керівник
к.т.н.

(підпис)

О.А. Хараджян

Запоріжжя
2022

ПРАТ «ПВНЗ «ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»

Кафедра Економічної кібернетики та інженерії програмного забезпечення

ДО ЗАХИСТУ ДОПУЩЕНА
Зав.кафедри _____
д.е.н., доцент Левицький С.І.

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ

Студенту гр. _____ ІПЗ–217, спеціальності «Інженерія програмного
забезпечення»

_____ Грибу Владиславу Дементійовичу

1.Тема: _____ Розробка системи імітаційного моделювання
турбовального двигуна

затверджена наказом по інституту № 06.2-14-2 від 18.02.2022 р.

2. Термін здачі студентом закінченої роботи: 18.06.2022 р.

3. Перелік питань, що підлягають розробці

1. Аналіз принципу роботи турбовального агрегату.
2. Аналіз методів імітаційного моделювання.
3. Постановка задачі розробки системи імітаційного моделювання.
4. Розробка математичної моделі турбовального агрегату.
5. Розробка структури імітаційної моделі
6. Розробка алгоритму роботи імітаційної моделі.
7. Опис даних програми
8. Опис програмних модулів.
9. Тестування системи імітаційного моделювання турбовального двигуна.

Дата видачі завдання: 17.06.2022 р.

Студент

(підпис)

В.Д. Гриб

(прізвище та ініціали)

Керівник роботи

(підпис)

О.А. Хараджян

(прізвище та ініціали)

РЕФЕРАТ

Дипломний робота містить 71 сторінок пояснюючої записки, 7 рисунків, 2 додатки.

Об'єкт дослідження: імітаційне моделювання динамічних систем.

Предмет дослідження: програмна система імітаційного моделювання турбовального двигуна.

Мета роботи: розробити алгоритми та програмну реалізацію системи імітаційного моделювання турбовального двигуна.

Завдання роботи: аналіз принципу роботи турбовального агрегату, аналіз методів імітаційного моделювання, постановка задачі розробки системи імітаційного моделювання, розробка математичної моделі турбовального агрегату, розробка структури імітаційної моделі, розробка алгоритму роботи імітаційної моделі, тестування системи імітаційного моделювання турбовального двигуна.

В даній роботі розглянуто будову та принцип роботи турбовального двигуна. На його основі розроблено математичну модель турбовального двигуна. Розглянуто принципи та методи імітаційного моделювання. Обрано метод системної динаміки, який використано в алгоритмі імітаційного моделювання двигуна та програмній системі.

ІМІТАЦІЙНА МОДЕЛЬ, ТЕСТУВАННЯ, ТУРБІНА, ТУРБОВАЛЬНИЙ ДВИГУН, PYTHON

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	6
ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ МАТЕМАТИЧНИХ МОДЕЛЕЙ ТУРБОВАЛЬНИХ ДВИГУНІВ.....	9
1.1. Аналіз принципу роботи турбовального агрегату.....	9
1.2. Аналіз методів імітаційного моделювання.....	17
РОЗДІЛ 2 РОЗРОБКА АЛГОРИТМУ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ.....	26
2.1. Розробка математичної моделі турбовального агрегату.....	26
2.2. Розробка алгоритму роботи імітаційної моделі.....	30
2.3. Розробка структури імітаційної моделі.....	32
РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ.....	34
3.1. Опис програмних модулів.....	34
3.3. Тестування системи імітаційного моделювання турбовального двигуна.....	43
ВИСНОВКИ.....	50
РЕКОМЕНДАЦІЇ.....	51
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	52
ДОДАТКИ.....	54

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ

Слово / словосполучення	Скорочення	Умови використання
В		
Вільна турбіна	ВТ	
Коефіцієнт корисної дії		
	ККД	
П		
Питома витрата палива	ПВП	
Т		
Турбіна високого тиску	ТВТ	
Турбіна низького тиску	ТНТ	

ВСТУП

В 20 столітті газотурбінні двигуни пройшли величезний еволюційний шлях. Починаючи від найперших зразків, котрі встановлювались на перші моделі реактивних літаків, закінчуючи надсучасними зразками, котрі використовуються на сучасних авіалайнерах.

В сучасному світі існує великий спектр різноманітних газотурбінних двигунів. Вони відрізняються за потужністю. Найменш потужні використовуються в відносно легких авіамоделях, найпотужніші в величезних за вантажопідйомністю літаках. Вони відрізняються за призначенням. Є двигуни, котрі використовуються в авіації – літаках та гвинтокрилах. Є двигуни, які використовуються в наземному транспорті, як в машинах спеціального призначення, на кшталт танків, так і в існуючих проектах автомобілів з газотурбінними двигунами. Врешті, газотурбінні двигуни використовуються в стаціонарних енергетичних установках.

Як наслідок великої різноманітності задач для двигунів, виникло багато різновидів газотурбінних двигунів, як то турбореактивні, турбогвинтові, турбовентиляторні, турбовальні.

Але, незважаючи на велику різноманітність таких двигунів, виникає потреба в розробці нових їх моделей. Це пов'язано по перше з розширенням спектру задач, які людство має намір вирішувати за допомогою такого типу двигунів, а по друге – зі збільшенням кількості доступних конструкторам новітніх матеріалів, котрі дозволяють підвищити експлуатаційні характеристики двигунів та зменшити споживання двигунами енергоресурсів.

Розробка проекту газотурбінного двигуна – це відносно невеликий за тривалістю, але надзвичайно період життєвого циклу двигуна. Виходячи зі сформульованих задач проектування формується необхідна інформація як про двигуні загалом, і про всі його елементи. Це, зрештою, визначає зовнішність

двигуна, особливості його конструктивного втілення, характер технологічних процесів при виробництві та умови його експлуатації.

У сучасному двигунобудуванні на етапах науково дослідної конструкторської роботи широко впроваджуються розрахункові методів дослідження, що базуються на використанні зростаючої продуктивності обчислювальних систем та сучасних прикладних програм, які дозволяють моделювати складні фізичні процеси, що відбуваються в двигуні та його основних вузлах. Застосування імітаційного моделювання особливо актуальне на ранніх стадіях проектування для скорочення кількості натурних та напівнатурних експериментів, мінімізації матеріальних та тимчасових витрат. Крім того, поєднання сучасних розрахункових методів та чисельних методів оптимізації дають можливість проведення цілеспрямованого пошуку технічних проектних рішень, що забезпечують граничні значення показників ефективності газотурбінних двигунів, що розробляються, та його окремих вузлів.

Об'єкт дослідження: імітаційне моделювання динамічних систем.

Предмет дослідження: програмна система імітаційного моделювання турбовального двигуна.

Мета роботи: розробити алгоритми та програмну реалізацію системи імітаційного моделювання турбовального двигуна.

Завдання роботи:

- аналіз принципу роботи турбовального агрегату.
- аналіз методів імітаційного моделювання.
- постановка задачі розробки системи імітаційного моделювання.
- розробка математичної моделі турбовального агрегату.
- розробка структури імітаційної моделі
- розробка алгоритму роботи імітаційної моделі.
- тестування системи імітаційного моделювання турбовального двигуна.

РОЗДІЛ 1

АНАЛІЗ МАТЕМАТИЧНИХ МОДЕЛЕЙ ТУРБОВАЛЬНИХ ДВИГУНІВ

1.1. Аналіз принципу роботи турбовального агрегату

Турбовальні двигуни є адаптацією технології газової турбіни, в якій основним виходом є потужність на валу від розширення гарячого газу через турбіну, а не тяга від вихлопних газів. Вони знайшли широкий спектр застосувань, починаючи від стиснення повітря і виробництва електроенергії до приводу гоночних човнів тощо.

Як різновид газової турбіни, турбовальні двигуни дуже схожі на турбореактивні. Принцип роботи ідентичний: атмосферні гази поглинаються на вході, стискаються, змішуються з паливом і спалюються, а потім розширюються через турбіну, яка живить компресор. Однак є дві ключові відмінності, які відрізняють турбовальні двигуни від турбореактивних.

На відміну від турбореактивних двигунів, основним призначенням яких є вироблення тяги з розширених газів, турбовальні двигуни призначені для вироблення потужності на валу. Цей фактично призводить до намагань конструкторів максимально ефективного турбовального двигуна мінімізувати те, що стає відомим як «залишкова тяга». У турбовальному двигуні турбіна зазвичай розділена на дві секції. Перша, турбіна високого тиску, використовується для приводу компресора, замикаючи термодинамічну петлю. Друга секція турбіни, турбіна низького тиску зазвичай механічно від'єднується від першої, і тому називається «вільною турбіною». Так само, як ТВТ забезпечує потужність на валу компресора, секція ТНТ забезпечує потужність на вихідному валу.

Щоб оцінити газотурбінний двигун, спочатку потрібно знати принцип роботи. Робочим тілом газотурбінного двигуна є газ, і його властивості змінюються протягом робочого циклу. Цикл можна визначити як процес, у якому газ в системі зазнає змін, щоб досягти кінцевого стану. Кілька параметрів газової турбіни, інакше відомих як змінні стану, такі як температура, тиск, масова витрата, зазнають змін і задовольняють термодинамічним законам для завершення циклу. Співвідношення, яке існує між різними змінними стану в циклі, можна знайти за допомогою 1-го і 2-го законів термодинаміки. Властивості газу доступні через таблицю властивостей або співвідношення кривої. Виконана робота і теплота, передана в цьому циклі, є функціями шляху (не залежать від стану газу). Інші параметри, такі як ентальпія, яка є мірою доступної енергії системи, та ентропія, яка є мірою недоступної роботи системи, допомагають визначити стан газу в системі.

Термодинамічний цикл газової турбіни може бути замкнутим або відкритим циклом. У замкнутому циклі робоче тіло залишається в системі, а високотемпературний відпрацьований газ охолоджується до початкового стану перед тим, як повернутися в цикл. У відкритому циклі робочий газ безпосередньо виходить із системи після проходження через турбіну. Існує багато параметрів, які характеризують умови роботи газотурбінного двигуна. Одними з найважливіших з них є питома витрата палива та тепловий коефіцієнт корисної дії.

Питома витрата палива визначається як відношення спаленого палива до виробленої потужності. Важливо зменшити ПВП в газотурбінному двигуні, щоб забезпечити кращу ефективність. Теплова ефективність - це відношення виробленої потужності до тепла, що додається в систему. Тепловий ККД в цілому можна підвищити за рахунок збільшення робочого тиску газу в системі або підвищення робочої температури турбіни. Теплова ефективність є важливим фактором загальної ефективності розглянутої системи (наприклад, для систем виробництва електроенергії).

Теплову ефективність простого циклу Брайтона можна підвищити, додавши до початкового простого циклу додаткові цикли нижньої частини або цикли рекуперації тепла. Цикл дна використовує тепло відпрацьованого вихідного циклу і діє як додаткове джерело до початкового циклу. У циклах рекуперації тепла використовується концепція рециркуляції відпрацьованого тепла в межах одного циклу, і цей процес також відомий як цикл регенерації/рекуперації. На додаток до вищезгаданих методів, існують інші модифікації, які можна зробити для підвищення ефективності циклу Брайтона, наприклад, турбовентилятор тощо.

Існує кілька типів газотурбінних двигунів, робота яких основана на використанні термодинамічного циклу, описаного вище. Широку класифікацію можна зробити на основі кількості валів двигуна, наприклад, одновальний або двовальний двигун. Одновальний двигун має компресор, турбіну і навантаження на одному валу. Для двовального двигуна компресор і турбіна газогенератора знаходяться на одному валу, тоді як силова турбіна і навантаження знаходяться на окремому валу. Принцип роботи подібний для обох випадків, коли навколишнє повітря (робоче тіло) проходить з вільного потоку у вхідний отвір компресора, який потім підвищує тиск повітря. Це повітря під тиском змішується з паливом та спалюється всередині камери згоряння. Отримана газова суміш високого тиску та високої температури розширюється в турбіні, щоб отримати роботу. Перевагами одновального двигуна є його компактні розміри і високий ККД. Для прискорення або уповільнення одновального двигуна в залежності від коливань навантаження потрібно менше часу. З іншого боку, головною перевагою двигунів з подвійним валом є автономна робота силової турбіни, що забезпечує більшу гнучкість для можливого застосування газової турбіни. Експлуатаційне порівняння одновального двигуна з двовальним з точки зору часткового навантаження: типовий газотурбінний двигун розроблений для роботи з більш високою ефективністю (коефіцієнт проектного навантаження), який зазвичай знаходиться близько проектної точки. Таким чином, експлуатація газової турбіни при

частковому навантаженні швидко знижує загальну паливну ефективність (враховуючи, що ефективність двигуна є функцією швидкості та навантаження), оскільки компресор і турбіна досягають нижчої ефективності незалежно під час цієї операції. Отже, якщо газогенеруюча система відокремлена від системи навантаження, тобто у випадку двовального двигуна, де газогенераторна система відокремлена від силової турбіни, то загальний ККД двигуна в результаті буде вищим при частковому навантаженні. З іншого боку, хоча одиновальний двигун не має особливої переваги при частковому навантаженні, якщо його правильно сконструювати з регулятором швидкості, загальна ефективність часткового навантаження одиновального двигуна також може бути покращена.

Для утилізації потужності на вихідному валу, яку продукують турбовальні двигуни, їм дуже часто потрібно використовувати коробку передач для зменшення швидкості обертання валу та, відповідно, збільшення крутного моменту. Наприклад, у той час як вільна турбіна може обертатися зі швидкістю понад 3000 об/хв, лопаті вертольота зазвичай обертаються зі швидкістю від 300 до 400 об/хв. Це зменшення, можливо, до рівня або вище, ніж співвідношення 10:1, неминуче вимагає досить складної трансмісії. На відміну від турбовальних двигунів, навантаження їх турбогвинтових родичів часто підключається безпосередньо до вихідного валу двигуна, без будь якої складної трансмісії.

Форми, які набуває енергії в турбовальному двигуні, в основному:

- тепло;
- робота;
- кінетична енергія;
- тиск.

Джерелом енергії для двигуна є паливо. Щоб горіння відбувалося в оптимальних умовах або взагалі на великій висоті, необхідно підвищити тиск повітря, що всмоктується, через компресор. Компресор споживає роботу і збільшує потенційну енергію повітря, підвищуючи його тиск. Далі, теплота, що

виділяється при згорянні палива, підвищує температуру газів, що згоріли, і збільшує інший тип потенційної енергії. Робота, яку споживає компресор, надходить від турбіни, яка поглинає частину енергії згорілих газів. Потім енергія, що залишилася, може бути використана або для викиду газів через сопло з високою швидкістю та безпосереднього створення тяги, або для використання іншої турбіни для живлення зовнішнього споживача, такого як пропелер, електрогенератор або компресор.

Це основна конструкція газотурбінного двигуна. Ця конструкція відповідає циклу Брайтона, циклу, характерному для двигуна внутрішнього згорання постійного тиску. Існують переважно дві форми, в яких цей цикл може бути розрахований: ідеальний і реальний. Останній забезпечує кращі результати, хоча для його обчислення потрібні ідеальні значення циклу та методи для кращого наближення результатів.

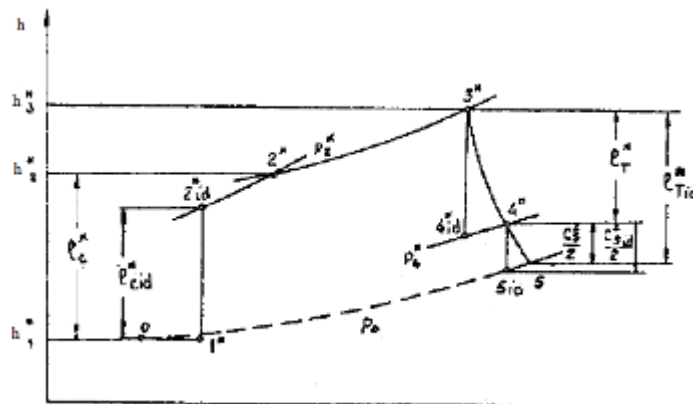


Рис. 1.1 - Робочий цикл турбового двигуна

На зображенні вище графічне зображення реального циклу роботи газотурбінного двигуна. Кожен стан двигуна, названий «1*», «2*» тощо, використовується для розрахунку термодинамічного параметру, що відповідає повній ентропії. «2*_{id}», «4*_{id}» і так далі означає стан ідеального циклу.

Вважається, що вони обчислюються за допомогою ізентропічного перетворення і знаходяться на ізобарній лінії з наступним станом реального циклу.

Кожен стан циклу визначається для частини двигуна між двома конструктивними компонентами. Ці стани включають:

- 0* - стан, що відповідає ділянці перед впуском двигуна;
- 1* - стан, що відповідає ділянці між впуском і компресором;
- 2*, 2*_{id} – стани, що відповідають перетину між компресором і камерою згоряння;
- 3* - стан, що відповідає стану між камерою згоряння та турбіною компресора;
- 4*, 4*_{id} – стан, що відповідає ділянці між турбіною компресора та силовою турбіною або соплом;
- 4T*, 4T*_{id} – стан, що відповідає ділянці між силовою турбіною та вихлопним соплом, цей стан необов'язковий і розраховується лише за наявності силової турбіни;
- 5*_{id}, 5 – стан, що відповідає перетину після вихлопного патрубка.

Значення термодинамічних параметрів кожного стану були розраховані шляхом припущення значення щонайменше двох термодинамічних параметрів, а потім обчислення решти з використанням наведених нижче поліномів. Усі властивості відповідають стану повної ентальпії:

- 0* - параметри передбачаються на рівні моря при температурі 15 °C і тиску 101325 Па;
- 1* - вхідний потік повітря вважається ізотермічним і з коефіцієнтом втрати тиску;
- 2*_{id} – для ідеального стану передбачається тиск через коефіцієнт тиску і що ентропія ідентична стану 1*;

– 2* - передбачається, що тиск ідентичний стану 2*id і зміна ентальпії, оскільки робота всередині компресора для реального стану нижча, ніж зміна вздовж компресора для ідеального стану з ефективністю компресора ;

– 3* - температура зчитується з [11] і розраховується тиск, припускаючи втрати всередині камери згоряння;

– 4*id – аналогічно стану 2*id, передбачається, що ентропія стану ідентична ентропії 3*, а робота, створена турбіною, або зміна ентальпії більша за роботу, яку споживає компресор, щоб компенсувати втрати валової передачі та потоку всередині турбіни, тому припускаються два коефіцієнта корисної дії: механічний і пов'язаний з турбіною;

– 4* - розраховується, якщо припустити, що тиск ідентичний 4*id, і припустити, що робота дорівнює лише значенню, збільшеному на механічний ККД;

– 4*Т - передбачається, що тиск вище атмосферного тиску на рівні моря, який передбачається в стані 0*, у рази;

– 5*id – передбачається, що тиск дорівнює атмосферному тиску на цій висоті, а ентропія ідентична стану 4* або 4*Т, в цьому стані розраховується ідеальна швидкість рідини;

– 5 – передбачається, що тиск ідентичний стану 5*id і сопло має недоліки потоку, тому до швидкості, яка призвела до stat 5*id, необхідно застосувати коефіцієнт.

Для кращої апроксимації термодинамічних параметрів використовуються поліноми, які обчислюють ентропію, як функцію температури. Ці поліноми перевіряються за допомогою термодинамічних значень.

$$i_{air}^{p=1}(T) = 16.7748 + 0.901616 T + 1.47424 * 10^{-4} T^2 - 1.94118 * 10^{-8} T^3$$

$$s_{air}^{p=1}(T) = 5.64896 + 4.09865 * 10^{-3} T - 2.31958 * 10^{-6} T^2 + 5.45627 * 10^{-10} T^3$$

$$i_{sh}^{p=1}(T) = 14.9623 + 0.920391 T + 2.16424 * 10^{-4} T^2 - 3.10594 * 10^{-8} T^3$$

$$s_{sh}^{p=1}(T) = 5.90016 + 4.01925 * 10^{-3} T - 2.10016 * 10^{-6} T^2 + 4.73272 * 10^{-8} T^3$$

Для розрахунку циклу необхідно оцінити два функціональні параметри. Це коефіцієнт тиску компресора та витрата повітря. Для коефіцієнта тиску можна розрахувати оптимальне значення. Зміни потужності, що забезпечує двигун, у відношенні тиску можна побачити на наступному зображенні.

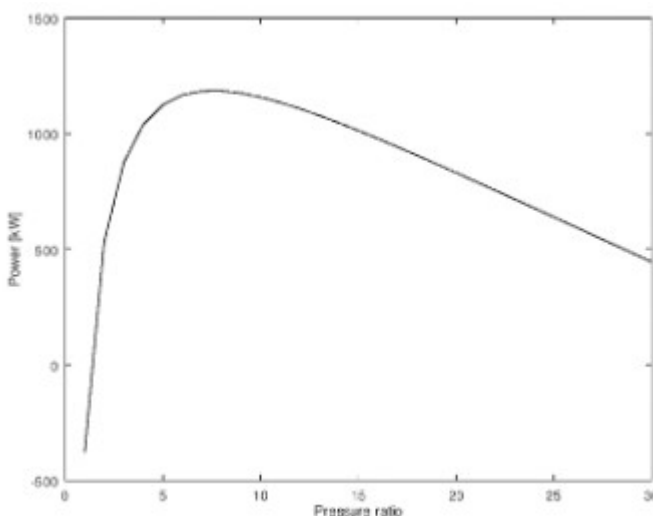


Рис. 1.2 - Залежність потужності двигуна від тиску

Для розрахунку циклу необхідно оцінити два функціональні параметри. Це коефіцієнт тиску компресора та витрата повітря. Для коефіцієнта тиску можна розрахувати оптимальне значення. Залежність потужності, що забезпечує двигун, від тиску можна побачити на рисунку 1.2.

Можна помітити, що існує значення коефіцієнта тиску, для якого потужність двигуна знаходиться на піковому значенні. Це значення вважається оптимальним коефіцієнтом тиску. Що стосується витрати повітря, то правильним значенням для двигуна вважається таке, для якого розрахункова потужність майже дорівнює реальній потужності двигуна.

Для розрахунку зазначених параметрів необхідно прийняти область значень співвідношення тисків і масової витрати для початкового розрахунку. Область

знаходиться між значеннями від 1 до 30 з кроком 1 для обох параметрів. Після того, як початковий результат параметрів обчислюється з передбачуваною областю, подальші значення обчислюються з більш обмеженою областю навколо більш точного кроку початкового результату.

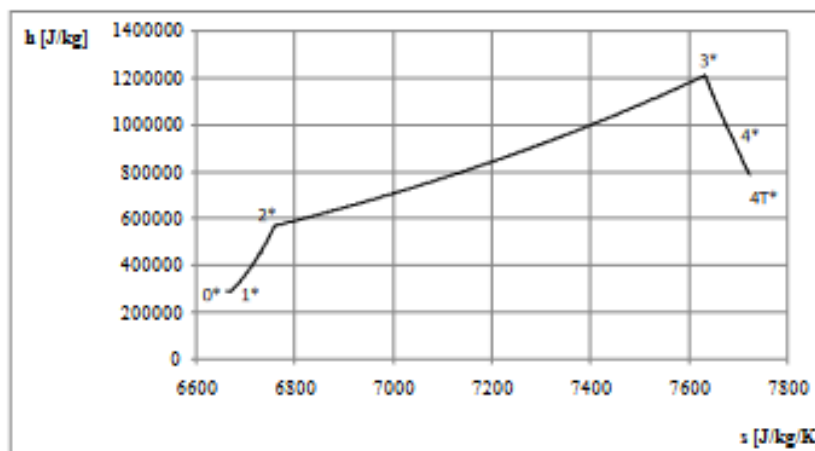


Рис. 1.3 - Стани термодинамічного циклу за винятком стану після випускного сопла

Розраховуються всі стани термодинамічного циклу, за винятком стану після випускного сопла. Ці стани та пов'язані з ними перетворення, наведені в системі h - s , представлені на рис. 1.3.

1.2. Аналіз методів імітаційного моделювання

Моделювання є потужним методом для проектування та оцінки складних систем і процесів. Знання принципів моделювання є важливим для багатьох аналітиків і менеджерів проектів, оскільки вони беруть участь у державних дослідженнях та розробках. У цій роботі представлено повний опис цих принципів, пропонуючи читачеві низку кроків. Процес починається з ретельного формулювання проблеми та побудови моделі і продовжується моделюючими

експериментами, інтерпретацією результатів, валідацією моделі, документуванням, підтвердженням результатів та остаточною реалізацією висновків дослідження. До кожного кроку потрібно підходити системно; жоден не може бути безпечно опущений. На кожному кроці визначаються критичні проблеми, а також представлені рекомендації щодо успішного завершення моделювання та симуляційних досліджень.

Побудова абстракцій систем (моделей) для полегшення експериментування та оцінки (симуляції) є і мистецтвом, і наукою. Ця техніка особливо корисна при вирішенні проблем складних систем, де простіших рішень немає. Методи моделювання також дозволяють проводити експерименти, які в іншому випадку були б громіздкими або неможливими. Наприклад, комп'ютерне моделювання стало причиною багатьох досягнень у таких галузях, як біологія, метеорологія, космологія, динаміка населення та військова ефективність. Без симуляції вивчення цих суб'єктів може бути загальмовано через відсутність доступу до реальної системи, необхідність вивчати систему протягом тривалих періодів часу, складність набору суб'єктів для експериментів або всі ці фактори. Оскільки ця техніка пропонує рішення цих проблем, вона стала надзвичайно потужним інструментом для вивчення тонкощів сучасного все більш складного світу.

Якими б потужними не були методи моделювання, їх безсистемне застосування може призвести до помилкових висновків. У цій статті представлено структурований набір рекомендацій, які допоможуть практикуючому уникнути невдачі та успішно застосувати методологію моделювання.

Однак рекомендації – це все, що можна запропонувати. Незважаючи на міцну основу в математиці, інформатиці, ймовірності та статистиці, ця дисципліна залишається інтуїтивною.

Для вивчення систем використовуються моделювання. Система визначається як сукупність сутностей, які складають об'єкт або процес, що представляє інтерес. Щоб полегшити експериментування та оцінку системи,

будується уявлення, яке називається моделлю. Фізичні моделі представляють системи як фактичне обладнання, тоді як математичні моделі представляють системи як набір обчислювальних або логічних асоціацій. Моделі також можуть бути статичними, представляючи систему в певний момент часу, або динамічними, що представляють, як система змінюється з часом. Набір змінних, який описує систему в будь-який конкретний момент часу, називається вектором стану. Загалом, вектор стану змінюється у відповідь на системні події. У безперервних системах вектори стану змінюються постійно, але в дискретних системах вони змінюються лише скінченну кількість разів. Якщо присутня хоча б одна випадкова величина, модель називається стохастичною; коли випадкові величини відсутні, модель називається детермінованою. Іноді математичні співвідношення, що описують систему, досить прості, щоб їх розв'язати аналітичним шляхом. Аналітичні рішення надають точну інформацію щодо продуктивності моделі. Коли модель не може бути розв'язана аналітично, вона часто може імітувати системні операції – процес, який називається моделюванням, що дозволяє оцінити продуктивність чисельно.

Загалом, системи моделюються як такі, що мають ряд вхідних даних (або стимулів) X_1, X_2, \dots, X_r , кількість виходів (або відповідей) Y_1, Y_2, \dots, Y_s ; і ряд системних параметрів (або умов) P_1, P_2, \dots, P_c . Хоча кожна система унікальна, вхідні дані часто є непередбачуваними явищами, а параметри системи часто виникають як засіб для налаштування відповідей у певний бажаний спосіб. Таким чином, вхідні дані зазвичай моделюються як випадкові процеси, а параметри системи як регульовані умови. Систему часто розглядають як функцію f , яка виробляє вихід Y з входів X і системних параметрів P , тобто $y = f(x, p)$.

Цей процес складається з таких основних етапів.

Формулювання проблеми. Постановка проблеми описує досліджувану систему та перелічує вхідні дані, системні параметри та вихідні дані. Створення формулювань проблеми починається з дослідження системи, що може бути

непростим завданням, оскільки в складних системах може виникнути так багато взаємопов'язаних явищ. Один із способів боротьби зі складністю — розглядати всю систему як ієрархію підсистем. Починаючи з найвищого рівня, всю систему можна рекурсивно розкласти на менші підсистеми, поки складність кожної підсистеми не стане керованою. Кожна підсистема характеризується своїм входом, виходом і обробкою, яка перетворює вхідні дані у вихідні, а також відношенням підсистеми до всієї системи та інших підсистем. Щоб отримати таку характеристику, може знадобитися читання системних посібників, обговорення з системними експертами, багато сесій спостереження, використання складного обладнання для збору даних або всі ці дії разом.

Зазвичай припущення робляться для спрощення аналізу системи. Наприклад, оцінка своєчасності повідомлення в мережі зв'язку може не вимагати моделювання деталей обробки на комутаційних станціях. Час, необхідний для виконання обробки, може замінити фактичний біт, маніпуляції перемикачів в моделі. Список припущень зазвичай створюється під час процесу формулювання моделі, але записи можуть також впливати з уроків, отриманих під час будь-яких попередніх спроб виправити, змінити чи оцінити систему чи подібну систему.

Моделювання передбачає побудову абстрактного представлення системи. Розробник моделі повинен визначити, які елементи системи є суттєвими, що часто є складним завданням. Виняток суттєвих елементів призводить до недійсної моделі, але включення непотрібних елементів надмірно ускладнює модель і збільшує вартість оцінки.

Оскільки кожна система унікальна, можна запропонувати лише три широкі рекомендації для інтуїтивно зрозумілого та майстерного процесу формулювання моделей. По-перше, аналітик повинен отримати досвід, ознайомившись з літературою з моделювання. Більшість цієї літератури містить конкретні моделі та конкретні методики, лише деякі з яких можуть бути корисними в майбутніх проектах моделювання. Менша частина містить корисні списки проблем, які

розробник моделей повинен завжди враховувати. По-друге, аналітик повинен культивувати використання як право-мозкового (образного та творчого), так і ліво-мозкового (раціонального та дисциплінованого) мислення. Моделювання є найбільш успішним, коли використовуються обидва. По-третє, аналітик повинен почати з простої моделі і додавати деталі лише тоді, коли процес перевірки покаже, що вони потрібні.

Що стосується вхідних даних, імітаційні моделі класифікуються як моделі керовані трасуванням, або керовані розподілом. Моделі керовані трасуванням використовують дані, які були зібрані під час вимірювань реальної системи. Справжня система повинна існувати і бути вимірюваною. Керовані розподілом моделі використовують розподіли ймовірностей (наприклад, експоненціальний, нормальний або Пуассоновський), отримані з даних сліду, або з припущень, коли дані сліду недоступні.

Процес отримання розподілів ймовірностей включає вибір типу розподілу, оцінку значень для всіх параметрів розподілу та оцінку результуючого розподілу на узгодження з даними трасування. Вибір відповідного типу розподілу допомагає візуальна оцінка графіків і використання статистичних атрибутів, включаючи коефіцієнти варіації (відношення квадратного кореня дисперсії до середнього для безперервних випадкових величин) і лексичних відношень (відношення дисперсія від середнього для дискретних випадкових величин). Причина використання конкретного розподілу може бути вагомою, оскільки час прибуття моделюється як процес Пуассона. Значення параметрів розподілу часто оцінюються за допомогою методу максимальної правдоподібності. Цей метод передбачає знаходження значень одного або кількох параметрів розподілу, які максимізують спільну функцію ймовірності (ймовірності), сформовану на основі даних трасування. Оцінка результуючого розподілу для узгодження з даними слідів включає візуальну оцінку графіків і тести на придатність. Тести на придатність – це спеціальні тести статистичних гіпотез, які намагаються оцінити узгодження між

даними слідів і отриманими розподілами. Деякі з найбільш поширених тестів на придатність - це χ^2 -квадратний тест, тест Колмогорова-Смімова і тест Андерсона-Дарлінга.

У деяких випадках (дослідження концепції, наприклад) збір даних слідів неможливий. Коли дані недоступні, слід вибрати теоретичні розподіли, які можуть охопити весь діапазон поведінки системи. Моделювання має проводитися для кількох налаштувань параметрів розподілу, а результати повинні бути зазначені з відповідним ступенем невизначеності.

Остання частина етапу формулювання включає пошук відповідної техніки рішення. Цей пошук має бути вичерпним, а вибрана методика повинна мати найвище співвідношення користі та витрат. Як правило, перевага віддається аналітичним рішенням, оскільки вони точні. На жаль, величезна складність багатьох систем унеможливорює використання методів аналітичного рішення. Для таких систем моделювання може бути єдиним ефективним методом вирішення.

Під методом імітаційного моделювання маємо на увазі загальну структуру для відображення реальної системи на її модель. На сьогоднішній день існує три методи:

- системна динаміка;
- моделювання дискретних подій;
- агентне моделювання.

Вибір методу повинен ґрунтуватися на системі, яка моделюється, та меті моделювання, хоча найчастіше на нього найбільше впливає фон або доступний набір інструментів розробника.

Іноді різні частини системи найкраще моделювати за допомогою різних методів. У цьому випадку будується багатометодна модель.

Системна динаміка — полягає у використанні законів фізики, зокрема законів електричних ланцюгів, для опису та дослідження динаміки економічних, а згодом і соціальних систем.

Системна динаміка — метод вивчення динамічних систем.

Математично модель системної динаміки - це система зв'язаних нелінійних диференціальних рівнянь першого порядку.

$$\frac{d(X)}{dt} = F(X, P),$$

де X — вектор запасів;

P — набір параметрів;

F — нелінійна функція.

Обчислення моделей системної динаміки здійснюється за допомогою чисельних методів, які розбивають час на дискретні інтервали довжини dt і обчислюють стан системи через час по одному dt за раз.

На відміну від моделей на основі дискретних подій і агентів, моделі системної динаміки є детермінованими, якщо в них явно не вставлені стохастичні елементи.

Ідея методу дискретного моделювання подій полягає в наступному: розробник моделі розглядає систему, що моделюється, як процес, тобто послідовність операцій, що виконуються між об'єктами.

Операції включають затримки, обслуговування різними ресурсами, вибір гілки процесу, розбиття, об'єднання та деякі інші. Оскільки сутності конкурують за ресурси і можуть бути відкладені, черги присутні практично в будь-якій моделі дискретних подій. Модель задається графічно у вигляді блок-схеми процесу, де блоки представляють операції (є також текстові мови, але вони в меншості). Блок-схема зазвичай починається з блоків «джерела», які генерують сутності та вводять їх у процес, і закінчується блоками «приймач», які видаляють об'єкти з моделі.

Типовий вихідний результат, який очікується від моделі дискретних подій:

- використання ресурсів;

- час перебування в системі або її частині агентом;
- час очікування;
- довжина черги;
- пропускна здатність системи;
- вузькі місця;
- вартість обробки агента та його структура.

Математика моделювання дискретних подій заснована на дискретному часі. Модельний годинник просувається лише тоді, коли в моделі відбувається щось важливе, а саме, коли агент починає або завершує операцію. Будь-яка зміна моделі пов'язана з цими подіями; безперервні зміни апроксимуються миттєвими.

Моделювання на основі агентів є більш сучасним методом моделювання, ніж системна динаміка або моделювання дискретних подій.

Моделювання на основі агентів пропонує розробнику моделювання ще один спосіб поглянути на систему.

Можна не знати, як поводить система в цілому, які ключові змінні та залежності між ними, або просто не бачите, що існує потік процесу, але ви можете мати певне уявлення про те, як об'єкти в системі поведуться окремо. Тому можна почати будувати модель знизу вгору, визначивши ці об'єкти (агенти) та визначивши їх поведінку.

Іноді можна з'єднати агентів один з одним і дозволити їм взаємодіяти; в інших випадках ви можете помістити їх у середовище, яке може мати власну динаміку. Глобальна поведінка системи тоді виникає з багатьох (десятьків, сотень, тисяч, навіть мільйонів) одночасних індивідуальних поведінок.

Поведінка агентів визначається багатьма різними способами. Часто агент має поняття про стан, і його дії та реакції залежать від його стану. У таких випадках поведінку найкраще визначити за допомогою діаграм станів. Іноді поведінка визначається у вигляді правил, які виконуються під час особливих подій. У багатьох випадках внутрішню динаміку агента можна найкраще охопити

за допомогою системної динаміки або підходу дискретних подій. У цих випадках ми можемо помістити в агента діаграму запасів і потоків або блок-схему процесу. Аналогічно, процеси поза агентами та динаміка середовища, де вони живуть, часто природно моделюються за допомогою традиційних методів. Таким чином, ми виявили, що великий відсоток моделей на основі агентів є моделями з багатьма методами.

Моделювання на основі агентів не передбачає жодного конкретного рівня абстракції. Якщо агенти є індивідуальними особами, то модель, заснована на агентах, безумовно, є більш детальною, ніж сегментована модель системної динаміки, де індивіди агреговані на основі характеристик. Агент, однак, можна розробити з високим рівнем абстракції. Наприклад, агентами можуть бути конкуруючі проекти, компанії або навіть ідеї чи філософії.

Більшість моделей на основі агентів працюють у дискретному часі – взаємодія, прийняття рішень і зміни стану відбуваються миттєво. У цьому відношенні на низькому рівні механізм моделювання не повинен сильно відрізнятися від того, який використовується для моделювання дискретних подій.

РОЗДІЛ 2

РОЗРОБКА АЛГОРИТМУ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ

2.1. Розробка математичної моделі турбовального агрегату

Математичні моделі дають уявлення про характеристики двигуна та фізику моделі під час моделювання. Закони збереження та рівняння руху служать відправною точкою для розробки цієї динамічної математичної моделі, яка складається з робочої рідини та компонентів обертання. Загальна продуктивність газотурбінного двигуна визначається його основними компонентами: компресором, камерою згоряння, турбіною, впуском/випуском і допоміжним обладнанням двигуна. Математична модель має бути достатньо гнучкою та давати достовірні результати.

При моделюванні компресору використовується карта продуктивності компонентів і співвідношення енергії. Рідина входить в компресор з одного кінця і виходить з високим тиском на іншому кінці. Зміна імпульсу рідини і підвищення тиску обумовлені зовнішнім крутним моментом, що діє на неї. Характеристики компресора виглядають наступним чином:

$$W_c = m * (h(T_i) - h(T_{out})) \quad (2.1)$$

$$\tau_c = \frac{W_c}{\omega} \quad (2.2)$$

$$T_{out} = T_i + \frac{T_i}{\eta_{c,isen}} (PR_c^{\frac{\gamma-1}{\gamma}} - 1) \quad (2.3)$$

$$PR_c = \frac{P_{out}}{P_i} \quad (2.4)$$

Рівняння (2.7) являє собою роботу компресора та отримано за рахунок збереження енергії без теплопередачі. Рівняння (2.8) представляє рівняння

крутного моменту, а рівняння (2.9) використовує газовий закон для обчислення температури на виході. Рівняння (2.10) дає співвідношення тиску для компресора.

Характеристики компресора представлені у вигляді карт компонентів. Перевага використання карти компонентів полягає в тому, що вона включає всі втрати для конкретного проектного випадку. Гнучким способом представлення карти компонентів у моделюванні є форма таблиць. Представлення характеристик компресора у форматі таблиці як функції допоміжної координати (β_c) та відносної швидкості показано в співвідношеннях (2.5)-(2.7).

$$PR_c = Fn(\beta_c, \% N_c) \quad (2.5)$$

$$m \frac{\sqrt{\theta}}{\delta} = Fn(\beta_c, \% N_c) \text{ при } \theta = \frac{T_o}{288.15}; \delta = \frac{P_o}{101325} \quad (2.6)$$

$$\eta_{c, isen} = Fn(\beta_c, \% N_c) \quad (2.7)$$

Продуктивність камери згоряння визначається кількістю виробленого тепла та втратою тиску в камері. Для відомих умов температури на вході та на виході, та втрати тиску розраховуються на основі рівнянь, наведених нижче. Рівняння (2.14) являє собою відношення зміни ентальпії до тепла, доданого в систему, і визначає ефективність камери згоряння.

$$\eta_b = \frac{1}{m_f * HV} (m_{out} * h(T_{out}) - m_c * h(T_c)) \quad (2.8)$$

$$P_{out} = P_c (1 - \Delta P R_b) \text{ при } \Delta P R_b = (\Delta P R_b)_{des} * \left(\frac{\frac{m_c * \sqrt{T_c R}}{P_c}}{\left(\frac{m_c * \sqrt{T_c R}}{P_c} \right)_{res}} \right)^2 \quad (2.9)$$

$$m_{out} = m_c + m_f \quad (2.10)$$

Рівняння (2.15) дає розрахунок тиску на виході з камери згоряння. За формулою (2.10) обчислюється вихідна масова витрата.

Подібно до моделювання компресора, для моделювання турбіни також використовують карти продуктивності компонентів і співвідношення енергія/імпульс. Газ високого тиску і температури розширюється через турбіну і генерує потужність для обертання валу. Вироблена потужність використовується для роботи компресора та інших допоміжних компонентів. Основні рівняння турбіни наведені нижче.

$$W_T = m * (h(T_i) - h(T_{out})) \quad (2.11)$$

$$\tau_T = \frac{W_T}{\omega} \quad (2.12)$$

$$T_{out} = T_i - T_i * \eta_{T, isen} * \left(1 - \left(\frac{1}{PR_T} \right)^{\frac{\gamma-1}{\gamma}} \right) \quad (2.13)$$

$$PR_T = \frac{P_{out}}{P_i} \quad (2.14)$$

Рівняння (2.17) розраховує роботу турбіни. Це похідне від енергозбереження без втрати тепла в навколишнє середовище. Аналогічно ентальпії розраховуються з кореляції кривої. Рівняння (2.18) дає співвідношення крутного моменту, а рівняння (2.19) використовує газовий закон для обчислення температури на виході. Рівняння (2.20) визначає коефіцієнт тиску.

Дотримуючись тієї ж процедури, що описана вище, карта характеристик турбіни перетворюється в табличну форму з використанням допоміжної координати та відносної швидкості.

$$PR_T = Fn(\beta_T, \% N_T) \quad (2.15)$$

$$m \frac{\sqrt{\theta}}{\delta} = Fn(\beta_T, \% N_T) \text{ при } \theta = \frac{T_o}{288.15}; \delta = \frac{P_o}{101325} \quad (2.16)$$

$$\eta_{T,isen} = Fn(\beta_T, \% N_T) \quad (2.17)$$

Щоб передбачити перехідну поведінку газової турбіни, необхідно враховувати постійні часу, пов'язану з інерцією ротора.

$$\frac{d\omega}{dt} = \frac{\tau_{drive} - \tau_{driven}}{I} \quad (2.18)$$

Під час перехідного процесу незбалансований крутний момент, прикладений до валу, призводить до прискорення або уповільнення ротора.

Насос із зовнішнім приводом використовується для подачі палива у вхідний канал камери згоряння. Потужність накачування залежить від різниці тисків, об'ємного потоку та ефективності насоса. Він визначається наступним рівнянням.

$$P_{Pump, power} = \frac{\Delta P_{Tot, Pressure} * Q}{\eta_{pump}} \quad (2.19)$$

Моделювання вхідних і випускних каналів однакове і відповідає наведеному нижче співвідношенню для врахування втрати тиску.

$$P_{out} = P_i (1 - \Delta P R_b) \quad \text{при} \quad \Delta P R_b = (\Delta P R_b)_{des} * \left(\frac{\frac{m_i * \sqrt{T_i R}}{P_i}}{\left(\frac{m_i * \sqrt{T_i R}}{P_i} \right)_{res}} \right)^2 \quad (2.20)$$

Газова турбіна в основному перетворює енергію палива в корисну роботу. Газова турбіна працює на термодинамічному принципі Брайтона. Повітря, що надходить через вхідне повітря, стискається в компресорі і додається паливо для

підвищення температури газу під тиском. Потім він розширюється в турбіні для виробництва енергії.

Умови відповідності параметрів:

- швидкість валу компресора і турбіни повинна бути рівною;
- масовий витрата турбіни - це лише сума масового потоку повітря з вихідного отвору компресора і масового витрати палива в камері згоряння;
- припускають, що втрата тиску в камері згоряння є постійним невеликим відсотком тиску на вході в камеру згоряння.

2.2. Розробка алгоритму роботи імітаційної моделі

Серед основних алгоритмів моделювання можна виділити два – моделювання за принципом особливих станів та принцип постійного збільшення модельного часу.

Алгоритм моделювання за принципом особливих станів зводиться до таких дій:

- визначається подія з мінімальним часом - найбільш рання подія;
- модельному часу надається значення часу настання найбільш ранньої події;
- визначається тип події;
- залежно від типу події вживаються дії, спрямовані на завантаження пристроїв та просування заявок відповідно до алгоритму їх обробки, та обчислюються моменти наступу майбутніх подій; ці дії називають реакцією моделі на події.

Перелічені дії повторюються до закінчення часу моделювання.

Далі буде розглядатись алгоритм за принципом постійного збільшення модельного часу (рис.2.1).



Рис. 2.1 блок-схема алгоритму моделювання за принципом dt

Алгоритм моделювання, заснований на застосуванні dt , використовується для ширшого кола систем, ніж алгоритм, побудований за принципом спеціальних станів. Однак за його реалізації виникають проблеми визначення величини dt . Для моделювання приладових систем на системному рівні переважно застосовуються принцип спеціальних станів.

Для моделювання двигуна був обраний алгоритм за принципом dt .

2.3. Розробка структури імітаційної моделі

Основою програмної моделі буде клас `Engine`. Його атрибутами будуть об'єкти підкласів, кожен з яких відповідає одній із складових систем турбовального двигуна. Також, деякі атрибути містять поточний стан двигуна. Методи класу призначені для повернення стану та керування двигуном. Метод цього класу `update` викликає методи `update` всіх об'єктів, які зберігаються в його атрибутах.

Кожний клас містить атрибути, які зберігають стан компонента або системи, які відповідають об'єктам даного класу. Методи класів на основі формул та залежностей моделі двигуна оновлюють стан об'єкта. Також кожний клас містить метод `update`, який викликає всі інші методи, які відповідають за оновлення стану та метод, призначений для повернення поточного стану.

Також для функціонування моделі повинні бути задані деякі параметри.

Для ув'язування параметрів двигуна на розрахунковому режимі та визначення розмірів його проточної частини (площі характерних перерізів тракту, діаметри тощо) повинна бути задана необхідна вихідна інформація, що включає:

- дані про умови польоту (H , M н) та характеристику вхідного пристрою ($\sigma_{вх}$);
- параметри робочого процесу ($T^*_{г}$, m , $\pi^*_{в}$, $\pi^*_{к}$, $T^*_{ф}$ або інші еквівалентні параметри, наприклад, \bar{p}^* , α_{Σ} тощо);
- будь-який параметр, що характеризує абсолютний розмір двигуна – тяга R , витрата повітря $G_{в\Sigma}$, площа входу $F_{в}$ тощо;
- режимні параметри кожного каскаду компресора (наведені витрати, наведені частоти обертання, значення $\pi^*_{к}$, $\eta^*_{к}$, $\Delta K_{у}$);
- наведені швидкості в характерних перерізах тракту $\lambda_{ф.х}$, $\lambda_{кан. вх}$ тощо;

- розрахункові значення коефіцієнтів повного тиску (σ_f , σ_{II} , $\sigma_{к.с.}$ та ін.), коефіцієнтів повноти згоряння (η_g , η_f);
- параметри, що визначають зовнішній вигляд вентилятора, каскадів компресора та турбіни (розрахункові окружні швидкості, середні коефіцієнти навантаження, втулкові відносини або інші еквівалентні дані).

Зазвичай завдання не обмежується розрахунковим режимом, і для подальшого визначення параметрів двигуна в нерозрахункових умовах (наприклад, при побудові висотно-швидкісних та дросельних характеристик) вихідна інформація повинна містити дані про конкретні або узагальнені характеристики вузлів двигуна.

Усю вихідну інформацію можна поділити на чотири групи.

Першу групу складають константи (фізичні або формальні), наприклад, нижча теплотворна здатність палива H_u , вміст атмосферного повітря d , коефіцієнти апроксимації термодинамічних функцій і характеристик вузлів, необхідна точність вирішення системи визначальних рівнянь ϵ і т.д.

До другої групи відносяться параметри, що вважаються постійними на прийнятому рівні моделювання.

Третя група параметрів – це початкові наближення для незалежних змінних.

Четверта група вихідної інформації включає дані, яким повинні задовольняти двигун та його вузли. Так, наприклад, якщо задана тяга двигуна R , то розрахунковому режимі підбирається відповідне значення площі входу. Якщо у вихідній інформації задана величина наведеної швидкості в холодному перерізі форсажної камери, то це означає, що на розрахунковому режимі повинна бути обрана необхідна площа прохідного перерізу на вході в форсажну камеру, що задовольняє поставленою умовою.

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ

3.1. Опис програмних модулів

Програма складається з модулів `model.py` та `test.py`.

Модуль `model` містить класи, які представляють математичну модель двигуна забезпечують обчислення його стану.

```
import dataclasses
from abc import ABC
from dataclasses import dataclass
from typing import Optional
from numpy import *
from math import *
import sys, os
```

Створюються допоміжні функції, які використовуються під час обчислень

```
def rotationMatrix(angle, axis):
    c, mc = cos(angle), 1-cos(angle)
    s = sin(angle)
    l = sqrt(dot(array(axis), array(axis)))
    x, y, z = array(axis)/l
    r = array([
        [x*x*mc+c, x*y*mc-z*s, x*z*mc+y*s, 0],
        [x*y*mc+z*s, y*y*mc+c, y*z*mc-x*s, 0],
        [x*z*mc-y*s, y*z*mc+x*s, z*z*mc+c, 0],
        [0, 0, 0, 1]])
    return r.T
```

```
def scaleMatrix(sx, sy, sz):
    s = array([[sx, 0, 0, 0],
               [0, sy, 0, 0],
               [0, 0, sz, 0],
               [0, 0, 0, 1]])
    return s

def translationMatrix(tx, ty, tz):
    t = array([[1, 0, 0, tx],
               [0, 1, 0, ty],
               [0, 0, 1, tz],
               [0, 0, 0, 1]])
    return t

def lookAtMatrix(ex, ey, ez, cx, cy, cz, ux, uy, uz):
    e = array([ex, ey, ez])
    c = array([cx, cy, cz])
    up = array([ux, uy, uz])

    #normalize
    uplength = sqrt(dot(up, up))
    up = up / uplength

    # viewdirection
    f = c - e
    flength = sqrt(dot(f, f))
    f = f / flength

    s = cross(f, up)
    slength = sqrt(dot(s, s))
    s = s / slength
```

```

u = cross(s, f)
la = array([
    [s[0], s[1], s[2], -dot(s,e)],
    [u[0], u[1], u[2], -dot(u,e)],
    [-f[0], -f[1], -f[2], dot(f,e)],
    [0, 0, 0, 1]])

return la

def perspectiveMatrix(fov, aspect, zNear, zFar):
    f = 1.0/tan(fov/2.0)
    aspect = float(aspect)
    zNear = float(zNear)
    zFar = float(zFar)
    p = array([
        [f/aspect, 0, 0, 0],
        [0, f, 0, 0],
        [0, 0, (zFar+zNear)/(zNear-zFar), (2*zFar*zNear)/(zNear-
zFar)],
        [0, 0, -1, 0]])
    return p

```

Наступна частина включає класи, кожний з яких відповідає одній з систем двигуна. Перший клас відповідає компресору. Його функція update виконує оновлення його стану

```

class Compressor():
    def init():
        pass

    def update():
        PRc = Pout/Pin

```

```

Tout = Tin + Tin/niuc * (power(PRc, Gamma-1/Gamma)-1)
self.Wc = m*(Tin-Tout)
self.Tettac = self.Wc / Omega

```

Всі наступні класи відповідають цьому шаблону. Наступний клас відповідає камері внутрішнього згорання. Він так само оновлює його стан та моментальну вихідну потужність

```

class IgnitionChamber():
    def __init__():
        pass

    def update():
        self.mout = m_in +mf
        niub = 1/(mf * H * V) * (mout * Tout - Min * Tin)
        self.Pouti = Pin * (1 - DeltaPRb)

```

Наступний клас відповідає турбіні.

```

class Shaft():
    def __init__():
        pass

    def update():
        PRt = Pout / Pin
        Tout = Tin + Tin/niut * (power(PRt, Gamma-1/Gamma)-1)
        self.Wt = m * (Tin - Tout)
        self.Tettas = self.Wt / Omega

```

Клас, представлений далі відповідає насосу. Обчислюється потужність потоку палива.

```

class FuelPump():
    def __init__():
        pass

    def update():
        self.Ppower = deltaPpressure * Q / niup

```

Обчислюється потужність вихлопу. Повітря вбирається з такою ж потужністю.

```

class Exhaust():
    def __init__():
        pass

    def update():
        self.Poute = Pin(1-DeltaPRb)

```

Далі представлений клас двигуна, який включає в себе об'єкти всіх попередніх класів. Він має додатковий метод, який повертає стан системи.

```

class Engine():
    def __init__():
        self.compressor = Compressor()
        self.chamber = IgnitionChamber()
        self.shaft = Shaft()
        self.pump = FuelPump()
        self.exhaust = Exhaust()

    def update():

```

```

self.compressor.update()
self.chamber.update()
self.shaft.update()
self.pump.update()
self.exhaust.update()

def getState():
    state = []
    state.append(self.compressor.Wc)
    state.append(self.compressor.Tettac)
    state.append(self.chamber.mout)
    state.append(self.chamber.Pouti)
    state.append(self.shaft.Wt)
    state.append(self.shaft.Tettas)
    state.append(self.pump.Ppower)
    state.append(self.Poute)

```

Метод `test` використовується для тестування моделі.

Далі представлено клас, який безпосередньо відповідає за тестування моделі.

```

class TestEverythingTogether:
    def test_everything_ok(self) -> None:
        class Benzine(Gasoline):
            pass

        bezine95 = Benzine(octane=95)
        bezine98 = Benzine(octane=98)

        gas_tank = GasPortion(
            gasoline=bezine95,
            volume_liters=20,

```

```
)
gas_tank += GasPortion(
    gasoline=bezine98,
    volume_liters=20,
)
assert gas_tank.volume_liters == 40
assert gas_tank.gasoline.octane == 96.5

engine = Engine(
    gasoline=bezine98,
)

gas_pump = GasPump(
    connected_engine=engine,
    connected_gas_tank=gas_tank,
    max_flow_lps=0.2,
)

pedal = Pedal(
    connected_gas_pump=gas_pump,
)

pedal.press(
    how_hard=0.8,
    seconds=5,
)

assert engine.rotations == 328.23
pedal.press(
    how_hard=0.8,
    seconds=5,
)
def __add__(self, other:"GasPortion") -> "GasPortion":
```



```

        if type(self.gasoline).__name__ !=
type(other.gasoline).__name__:
            raise UnwantedGasolineMix
            volume_liters = self.volume_liters + other.volume_liters
            octane = self.gasoline.octane * (self.volume_liters /
volume_liters)
            octane += other.gasoline.octane * (other.volume_liters /
volume_liters)
            octane = round(octane, ndigits=2)
            return GasPortion(
                gasoline=dataclasses.replace(self.gasoline,
octane=octane),
                volume_liters=volume_liters,
            )
def produce_a_standard_diesel_engine(cls):
    class Diesel(Gasoline):
        pass

    diesel70 = Diesel(octane=95)
    return cls(
        gasoline=diesel70,
    )
def supply(
    self,
    gas_portion: GasPortion,
    seconds: float,
):
    self.haha = 5
    if gas_portion.volume_liters <= 0 or seconds <= 0:
        return
    self.total_supplied_gas += gas_portion
    flow_speed = min(
        gas_portion.volume_liters / seconds,

```

```

        self.max_flow_speed_lps
    )
        flow_speed_relative = flow_speed /
self.max_flow_speed_lps
        octane_relative = gas_portion.gasoline.octane /
self.gasoline.octane
        rpm = self.max_rpm * flow_speed_relative *
octane_relative
        # now convert rates per minute to rates per second
        rps = rpm / 60
        # and actually rotate the engine:
        rotations = round(rps * seconds, ndigits=2)
        self.rotations += rotations
        return rotations
def apply(
    self,
    voltage: float,
    seconds: float,
) -> GasPortion:
    if voltage < 0:
        raise NegativeVoltage
    if voltage > 14:
        raise TooHighVoltage
    flow_lps = self.max_flow_lps * voltage / 12
    volume = round(min(
        self.connected_gas_tank.volume_liters,
        flow_lps * seconds
    ), ndigits=2)
    self.connected_gas_tank.volume_liters -= volume
    gas_portion = dataclasses.replace(
        self.connected_gas_tank,
        volume_liters=volume
    )

```

```
if self.connected_engine is not None:  
    self.connected_engine.supply(  
        gas_portion=gas_portion,  
        seconds=seconds,  
    )  
return gas_portion
```

3.3. Тестування системи імітаційного моделювання турбовального двигуна

Оскільки модель заснована на даних двигуна, кілька основних значень продуктивності для різних налаштувань потужності порівнюються між трьома наборами даних: моделлю T-MATS двигуна та значеннями з тестування двигуна.

На рис.3.1 показано вимірювану температуру газу MGT, виміряну температуру на виході з турбіни, скориговану з урахуванням факторів температури навколишнього середовища та частоти обертання валу, описаної в інструкції з установки двигунів. Надано хорошу відповідність між блоком двигуна і моделлю T-MATS. Однак можна помітити відмінну відмінність між випробуваними і змодельованими двигунами до 10% для налаштувань низької потужності. Він помітно покращується до високих балів продуктивності, зменшуючи приблизно до 2%. Ця розбіжність також має місце між даними інституту випробувань двигуна та даними випробувань виробника двигуна. Оскільки процедура випробувань виробника не розкрита і невідома, то, ймовірно, помилка криється десь у методі випробування та вимірювання MG.

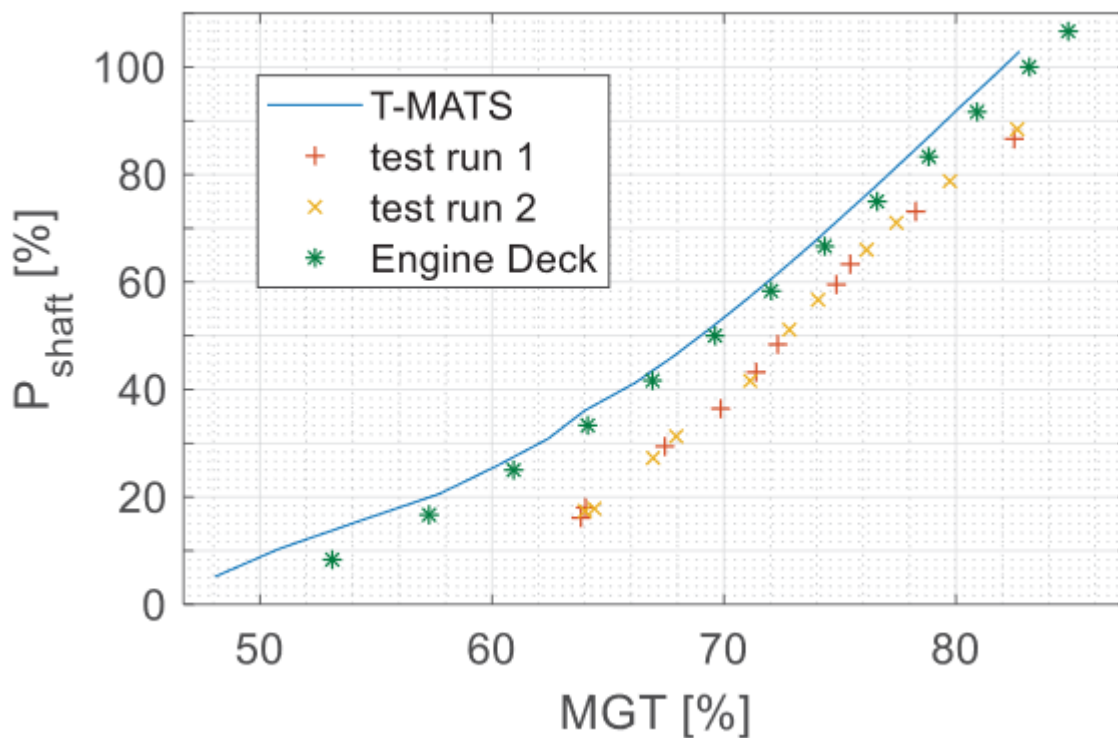


Рис. 3.1 - Виміряна температура газу(MGT) для різних налаштувань потужності для імітованого та реального двигуна

При розгляді швидкості газогенератора N_g можна спостерігати хорошу узгодженість даних. Можна виявити лише невелике зміщення менше ніж на 1% для більш високих налаштувань потужності між моделлю T-MATS, випробуваним двигуном і підключеним двигуном. Це вважається відносно хорошою угодою.

Те ж саме можна стверджувати для коефіцієнта тиску компресорів π на малюнку 8. Розбіжності між моделлю T-MATS, блоком двигуна та випробуваним двигуном завжди рівні або менше приблизно 2%. Це відхилення збільшується до свого максимуму приблизно на 2% для найвищої вихідної потужності.

Результати моделювання циклу двигуна порівнюються з даними випробувань двигуна. Для того, щоб забезпечити це, на випробувальному стенді були використані спеціальні екрани тиску на вході у впуск двигуна, щоб піддати двигун загальним викривленням тиску різного розміру та інтенсивності.

Щоб створити викривлення, перфоровані листи сплаву вмонтовані у впускну камеру двигуна. Пленум зазвичай забезпечує рівномірний розподіл повітря навколо всього радіального впускного отвору двигуна. Для зміни інтенсивності екрани неодноразово виготовлялися з однаковим шаблоном свердління, але з різними діаметрами отворів, а також для деяких випробувань частково або повністю покривалися для найбільшої інтенсивності.

Порівняні спотворення, змодельовані за допомогою T-MATS, забезпечені динамічною складовою, оскільки випробовувані викривлення перфорованого листа мають опору в залежності від швидкості потоку. Швидкість потоку на вході змінюється лінійно з потоком повітряної маси через двигун. Це, в свою чергу, визначається потужністю двигуна. В результаті імітовані спотворення повинні бути сильнішими для високих навантажень двигуна, ніж для низьких. Приблизну оцінку інтенсивності спотворення проводили на аеродинамічній моделі входу. Ця процедура необхідна, оскільки немає можливості повністю виміряти спотворення в AIP на самому двигуні.

Щоб визначити реакцію двигунів на спотворення впуску, зафіксували стаціонарні робочі точки в широкому полі продуктивності двигунів. Ці показники розподіляються по всьому діапазону випробувального обладнання з 5-відсотковим кроком крутного моменту двигуна і зберігаються щонайменше протягом хвилини, тоді як останні десять секунд використовуються для усереднення стаціонарних даних. Це необхідно для того, щоб виключити процеси теплової адаптації в двигуні при зміні рівня потужності та послідовної температури. Щоб визначити робочу лінію для кожної конфігурації спотворення, поліном, який спочатку найкраще відповідає даним палуби двигуна, зміщують уздовж осі абсцис (напрямок x). Цей зсув виконується таким чином, що досягається найменший квадрат відстаней до всіх зв'язаних точок вимірюваних даних.

На малюнку нижче показано нормовану норму MGT щодо різних нормованих налаштувань потужності $P_{\text{shaft, norm}}$ для різного розміру спотворення

однакової інтенсивності. Крпка з комою представляє дані моделі T-MATS, а безперервні лінії представляють вимірні робочі лінії, отримані в результаті зсуву лінії палуби двигуна. Нормалізація тестових даних здійснюється за допомогою ISA-умов. Модель T-MATS безпосередньо розраховується для ISA-умов. Наведені контрольні лінії призначені для неспотвореного впуску двигуна. Порівнюючи обидва набори даних, T-MATS і реальний двигун, можна виявити подібний вплив, що зображено диверсифікацією ліній, спотворення у напрямку двигунів MGT. У всякому разі, повна розбіжність між моделлю T-MATS і реальним двигуном у MGT все ще існує.

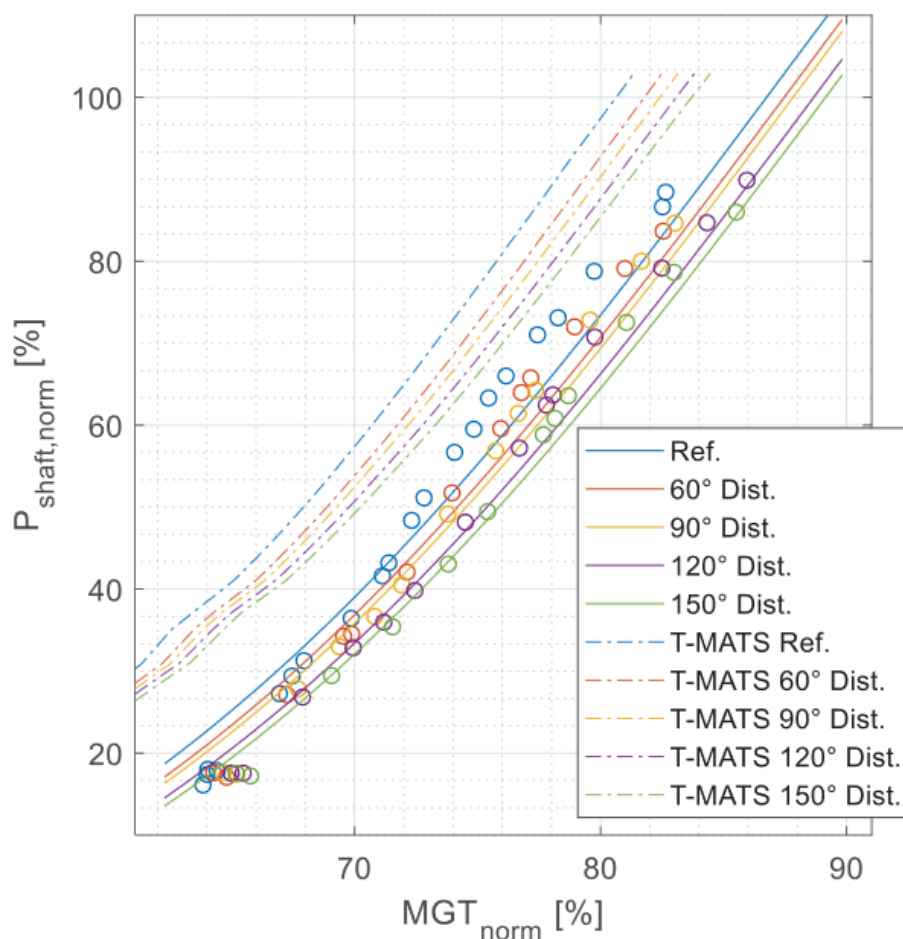


Рис.3.2 - Порівняння робочих ліній MGT для моделі і деки двигуна

На наступному малюнку, в основному, показаний той самий вміст, але для нормованої швидкості газогенератора N_g , норма на осі абсцис, замість MGT.

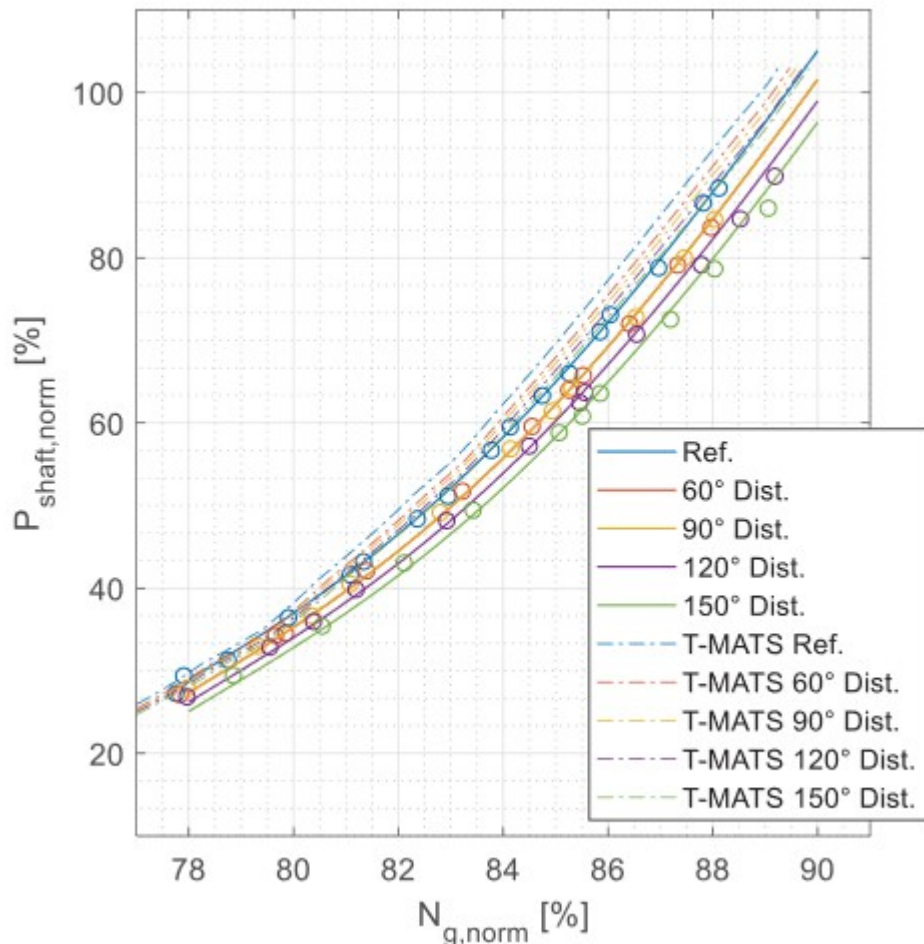


Рис.3.3 - Порівняння робочих ліній швидкості двигуна (N_g) для моделі та двигуна

Безперервні лінії, рівні попередньому малюнку, показують зміщену робочу лінію двигуна, щоб відповідати виміряним даним, відображеним крапками. Вплив викривлень на робочі лінії моделі T-MATS і тестового двигуна здається порівняним. Очевидно, спотворення впливає на цикл T-MATS так само, як і на цикл реальних двигунів.

Для збільшення розміру спотворення Θ , відображеного різними кольорами, досягнутий коефіцієнт тиску π для заданої норми швидкості компресора N_g ,

зменшується. Це можна зазначити для тестованого двигуна на верхньому малюнку, а також для моделі T-MATS на нижньому малюнку. Здається, що кількісний вплив є порівнянним між обома наборами даних, з менш ніж 1% впливом на низькі налаштування потужності та до 2-3% для параметрів високої потужності з високими коефіцієнтами тиску та великою швидкістю газогенератора. Така поведінка відповідає очікуваному ефекту, що викривлення негативно впливає на досяжний коефіцієнт тиску π компресора. Особливо для відцентрового компресора, оскільки двигун оснащений одноступінчастим відцентровим компресором.

Модель T-MATS з реалізованою моделлю паралельного компресора призводить до передбачуваних результатів для компресора, а також для всього циклу двигуна. При огляді на загальні експлуатаційні характеристики він показує оцінку поведінки у випадку викривлення тиску. Термодинамічний цикл напружується зростаючим спотворенням, що є для вимірюваної газової турбіни MGT і частоти обертання газогенератора N_g . Нарешті, загальна ефективність, представлена SFC, зменшується. На роботу компресорів безпосередньо впливає спотворення, що демонструє прогнозований зсув у бік погіршення продуктивності.

Для перевірки впливу спотворення на цикл двигуна було проведено серію випробувань двигуна для різних конфігурацій викривлення загального тиску та порівняння з даними моделі T-MATS. Була показана хороша відповідність. Іншим недоліком є те, що поки модель базується на картах продуктивності турбокомпонентів, які не є оригінальними з базового двигуна, точне положення меж стабільності компресорів невідоме. Прогноз критичного викривлення, що призводить до нестабільності двигуна, в даному випадку не є точним, оскільки він лише показує втрату стабільності для використаного девіантно змодельованого компресора.

Підсумовуючи, це означає, що з представленою моделлю T-MATS можна передбачити першу втрату продуктивності для оцінки кріплення двигуна та ситуації інтеграції. Це могло б допомогти зменшити фінансові та експлуатаційні ризики при проектуванні нового двигуна вертольота. Однак завжди слід критично ставитися до небезпеки збоїв стабільності, принаймні без оригінальної карти продуктивності виробника компресора.

На наступному кроці модель має бути розширена, щоб дати можливість врахувати температурні спотворення. Вони також мають велике значення для двигунів вертольотів. Зразковий потік гарячих вихлопних газів двигуна може легко транспортуватися за допомогою турбулентного ротора вниз до впуску двигуна. Для цих міркувань можна безпосередньо застосувати модель паралельного компресора, але контроль вхідних і граничних умов має бути адаптований. Оскільки температурні спотворення часто мають дуже динамічний характер, модель могла б використовувати T-MATS з урахуванням можливості виконання динамічного моделювання двигуна.

ВИСНОВКИ

На стадії проектування турбогвинтових двигунів основне джерело даних для верифікації методів виявлення дефектів все ще покладається на випробувальні установки компресора або всього двигуна. Це робить розробку моделей всього процесу перенапруги в двигуні нагальною потребою для заміни високовартісного та високоризикованого тестування.

Переваги такого моделювання численні. Використання реального апаратного забезпечення в симуляції усуває потребу в програмній імітації певних частин і значно скорочує час розробки програмного забезпечення. З іншого боку, це усуває необхідність проведення дорогих і часто небезпечних сертифікаційних випробувань реальної системи, зокрема газотурбінного двигуна.

В даній роботі було розглянуто будову та принцип роботи турбовального двигуна. На цій основі було розроблено математичну модель турбовального двигуна.

Розглянуто принципи та різні методи імітаційного моделювання. Після детального аналізу було обрано метод системної динаміки.

Після цього був розроблений алгоритм імітаційного моделювання двигуна та програмну модель та провели тестування.

Модель із реалізованою моделлю паралельного компресора дає передбачувані результати для компресора, а також для всього циклу двигуна. Якщо розглядати загальні експлуатаційні характеристики, то він показує оцінку поведінки у випадку викривлення тиску. Термодинамічний цикл напружується зростаючим спотворенням, що характерно для вимірюваної газової турбіни та швидкості газогенератора. Нарешті, загальна ефективність, представлена через , зменшується. На роботу компресорів безпосередньо впливає спотворення, що демонструє прогнозований зсув у бік погіршення продуктивності.

РЕКОМЕНДАЦІЇ

Розроблене в роботі програмне забезпечення дозволяє виконувати моделювання роботи турбовального двигуна, та деяких основних процесів, що відбуваються в ньому.

Програмне забезпечення може використовуватися як в великому комплексі з іншими програмами, що моделюють поведінку літального апарату, так і окремо, для вивчення поведінки двигуна турбовального двигуна на різних робочих режимах.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Andrei Borshchev, The Big Book of Simulation Modeling, Anylogic, 2013
2. Anu Maria, Introduction To Modeling And Simulation, State University of New York at Binghamton, 1997
3. Averill M. Law, How To Build Valid And Credible Simulation Models, Averill M. Law & Associates, 2009
4. Ballin M., A High Fidelity Real-Time Simulation Of A Small Turboshaft Engine, ResearchGate, 1988
5. Eshwarprasad Thirunavukarasu, Modeling and Simulation Study of A Dynamic GasTurbine System In A Virtual Test Bed Environment, University of South Carolina, 2013
6. Heng Wu, Gas Turbine Power Calculation Method Of Turboshaft Based On Simulation And Performance Model, MATEC Web of Conferences, 2018
7. Linnéusson G. On System Dynamics as an Approach for Manufacturing Systems Development, Chalmers Reproservice, 2009
8. Luke Stuyvenberg, Helicopter Turboshfts, University of Colorado at Boulder, 2012
9. William A. Menner, Introduction to Modeling and Simulation, Johns Hopkins Apl Technical Digest, 1995
10. Альсова О.К. Имитационное моделирование систем, Новосибирский Государственный Технический Университет, 2015
11. Костюков В.М., Капырин Н.И. Модель газотурбинного двигателя и его системы управления для особых значений углов атаки и метеорологических условий, Электронный журнал «Труды МАИ». Выпуск No 49
12. Кузнецов А. В., Макарьянц Г. М. Имитационная модель малоразмерного

газотурбинного двигателя, ResearchGate, 2017

13. Кулагин В.В., Кузьмичев В.С. Теория, расчет и проектирование авиационных двигателей и энергетических установок: учебник. 4-е изд., испр. В 2 кн. Кн. 1. Основы теории ГТД. Рабочий процесс и термогазодинамический анализ. – М.: Инновационное машиностроение, 2018
14. Максимей И.В., Смородин В.С., Демиденко О.М. Разработка имитационных моделей сложных технических систем, ГГУ им. Ф. Скорины, 2014
15. Мицель А.А., Математическое и имитационное моделирование, Изд-во ЮТИ(филиал)ТПУ, 2016
16. Салмина Н.Ю., Имитационное моделирование, «Эль Контент», 2012
17. Юнаков Л. П. Основы теории авиационных газотурбинных двигателей, Балт. гос. техн. ун-т. – СПб., 2013

ДОДАТКИ

Додаток А. Вихідний файл model.py

```

from numpy import *
from math import *

import sys, os
#В цій частині створюються споміжні функції, які
використовуються під час підрахунків
def rotationMatrix(angle, axis):
    c, mc = cos(angle), 1-cos(angle)
    s = sin(angle)
    l = sqrt(dot(array(axis), array(axis)))
    x, y, z = array(axis)/l
    r = array([
        [x*x*mc+c, x*y*mc-z*s, x*z*mc+y*s, 0],
        [x*y*mc+z*s, y*y*mc+c, y*z*mc-x*s, 0],
        [x*z*mc-y*s, y*z*mc+x*s, z*z*mc+c, 0],
        [0, 0, 0, 1]])
    return r.T

def scaleMatrix(sx, sy, sz):
    s = array([[sx, 0, 0, 0],
               [0, sy, 0, 0],
               [0, 0, sz, 0],
               [0, 0, 0, 1]])
    return s

def translationMatrix(tx, ty, tz):
    t = array([[1, 0, 0, tx],
               [0, 1, 0, ty],

```

```

        [0, 0, 1, tz],
        [0, 0, 0, 1]))
    return t

def lookAtMatrix(ex, ey, ez, cx, cy, cz, ux, uy, uz):
    e = array([ex, ey, ez])
    c = array([cx, cy, cz])
    up = array([ux, uy, uz])

    #normalize
    uplength = sqrt(dot(up, up))
    up = up / uplength

    # viewdirection
    f = c-e
    flength = sqrt(dot(f,f))
    f = f / flength

    s = cross(f, up)
    slength = sqrt(dot(s, s))
    s = s /slength

    u = cross(s, f)
    la = array([
        [s[0], s[1], s[2], -dot(s,e)],
        [u[0], u[1], u[2], -dot(u,e)],
        [-f[0], -f[1], -f[2], dot(f,e)],
        [0, 0, 0, 1]])

    return la

def perspectiveMatrix(fov, aspect, zNear, zFar):
    f = 1.0/tan(fov/2.0)

```

```

aspect = float(aspect)
zNear = float(zNear)
zFar = float(zFar)
p = array([
    [f/aspect, 0, 0, 0],
    [0, f, 0, 0],
    [0, 0, (zFar+zNear)/(zNear-zFar), (2*zFar*zNear)/(zNear-
zFar)],
    [0, 0, -1, 0]])
return p

```

#Наступна частина включає класи, кожний з яких відповідає одній з систем двигуна.

#Перший клас відповідає компресору та відповідає за оновлення його стану

```

class Compressor():
    def init():
        pass

    def update():
        PRc = Pout/Pin
        Tout = Tin + Tin/niuc * (power(PRc, Gamma-1/Gamma)-1)
        self.Wc = m*(Tin-Tout)
        self.Tettac = self.Wc / Omega

```

#Наступний клас відповідає камері внутрішнього згорання. Він так само оновлює його

```

#стан та моментальну вихідну потужність
class IgnitionChamber():
    def __init__():

```



```

pass

def update():
    self.mout = m_in +mf
    niub = 1/(mf * H * V) * (mout * Tout - Min * Tin)
    self.Pouti = Pin * (1 - DeltaPRb)

#Наступний клас відповідає турбіні.
class Shaft():
    def __init__():
        pass

    def update():
        PRt = Pout / Pin
        Tout = Tin + Tin/niut * (power(PRt, Gamma-1/Gamma)-1)
        self.Wt = m * (Tin - Tout)
        self.Tettas = self.Wt / Omega

#Насос. Підраховується потужність потоку палива.
class FuelPump():
    def __init__():
        pass

    def update():
        self.Ppower = deltaPpressure * Q / niup

#Підраховується потужність вихлопу. Повітря вбирається з такою ж
потужністю.
class Exhaust():

```

```

def __init__():
    pass

def update():
    self.Poute = Pin(1-DeltaPRb)

```

#Клас двигуна, який включає в себе об'єкти всіх попередніх класів.

```

class Engine():
    def __init__():
        self.compressor = Compressor()
        self.chamber = IgnitionChamber()
        self.shaft = Shaft()
        self.pump = FuelPump()
        self.exhaust = Exhaust()

    def update():
        self.compressor.update()
        self.chamber.update()
        self.shaft.update()
        self.pump.update()
        self.exhaust.update()

    def getState():
        state = []
        state.append(self.compressor.Wc)
        state.append(self.compressor.Tettac)
        state.append(self.chamber.mout)
        state.append(self.chamber.Pouti)
        state.append(self.shaft.Wt)
        state.append(self.shaft.Tettas)
        state.append(self.pump.Ppower)

```

```
state.append(self.Poute)
```

Додаток Б. Вихідний файл test.py

```
from model import Engine, Gasoline, GasPortion, GasPump, Pedal

class TestEverythingTogether:
    def test_everything_ok(self) -> None:
        class Benzine(Gasoline):
            pass

        bezine95 = Benzine(octane=95)
        bezine98 = Benzine(octane=98)

        gas_tank = GasPortion(
            gasoline=bezine95,
            volume_liters=20,
        )
        gas_tank += GasPortion(
            gasoline=bezine98,
            volume_liters=20,
        )
        assert gas_tank.volume_liters == 40
        assert gas_tank.gasoline.octane == 96.5

        engine = Engine(
            gasoline=bezine98,
        )

        gas_pump = GasPump(
            connected_engine=engine,
            connected_gas_tank=gas_tank,
            max_flow_lps=0.2,
        )

        pedal = Pedal(
            connected_gas_pump=gas_pump,
        )

        pedal.press(
            how_hard=0.8,
            seconds=5,
        )

        assert engine.rotations == 328.23
        pedal.press(
            how_hard=0.8,
            seconds=5,
        )
```

