ISSN 1727-7337 (print)
ISSN 2663-2217 (online)

УДК 681.5.015:621.452.3 **doi: 10.32620/aktt.2026.4sup2.13**

С. Р. ВЯЛОВ[1,2]

[1] *АТ «Елемент», Одеса, Україна*
[2] *Національний університет «Одеська політехніка», Одеса, Україна*

ОБҐРУНТУВАННЯ ФУНКЦІОНАЛЬНОЇ СТРУКТУРИ ПРОГРАМНОГО ЗАСОБУ ДЛЯ АВТОМАТИЗОВАНОЇ ПЕРЕВІРКИ АЛГОРИТМІВ І ФУНКЦІЙ ЗАХИСТУ РЕГУЛЯТОРА ГАЗОТУРБІННОГО ДВИГУНА

У статті розглянуто підхід до обґрунтування функціональної структури програмного засобу для автоматизованої перевірки алгоритмів і функцій захисту регулятора газотурбінного двигуна. Актуальність дослідження зумовлена тим, що критичні перехідні режими запуску та зупинки потребують безпечної, повторюваної й формалізованої перевірки, а натурне відтворення таких режимів є ризикованим і недостатньо відтворюваним. Метою роботи є обґрунтування функціональної структури програмного засобу, призначеного для візуального та формального задання, виконання й документування сценаріїв автоматизованої перевірки алгоритмів захисту регулятора ГТД. Методика дослідження базується на системному аналізі функцій електронного регулятора, узагальненні типових захисних алгоритмів, визначенні вимог до сценарію перевірки та формалізації груп функціональних вузлів програмного середовища. Запропоновано робочу функціональну класифікацію алгоритмів захисту для задач автоматизованої перевірки: функціонально-контрольні, тестово-сценарні, реконфігураційні та алгоритми, пов'язані з імітацією відмов і переходом на резервні значення. Показано, що з аналізу вимог до сценарію випливає мінімальний склад функціональних можливостей програмного засобу: задання початкових умов, формування тестових впливів, робота з аналоговими та дискретними сигналами, перевірка логічних і часових умов, реєстрація реакції регулятора, взаємодія зі стендом-імітатором і формування протоколу. Обґрунтовано доцільність подання сценарію у вигляді сукупності функціональних вузлів, серед яких вузли керування потоком виконання, роботи зі змінними, математичної та логічної обробки, часової організації, формування команд, взаємодії з обладнанням, логування, валідації та контролю помилок. Запропонована структура задає каркас для формалізованої побудови конкретних сценаріїв перевірки.

Ключові слова: *газотурбінний двигун; регулятор ГТД; алгоритми захисту; автоматизована перевірка; програмний засіб; візуальне та формальне задання сценаріїв; стенд-імітатор; функціональна структура.*

Вступ

Сучасні електронні регулятори газотурбінних двигунів виконують не лише функції автоматичного керування, а й захисту в критичних перехідних режимах запуску та зупинки. Саме ці режими є методично найскладнішими: їх небезпечно відтворювати на натурному двигуні, вони потребують повторюваності умов і вимагають формалізованого критерію PASS/FAIL. Тому актуальність дослідження визначається потребою в безпечній, відтворюваній і автоматизованій перевірці алгоритмів захисту в стендово-імітаційному середовищі [1–9].

Перевірка захисних алгоритмів є складним завданням, оскільки вона потребує відтворення штатних, граничних і нештатних ситуацій. Частина таких ситуацій не може бути безпечно реалізована на реальному двигуні без ризику пошкодження агрегатів або порушення умов випробування. Тому доцільним є використання стендово-імітаційних засобів як середовища, в якому сценарій перевірки може бути заданий, виконаний і задокументований з контролем часових і логічних умов, без створення реальної аварійної ситуації. Тобто графічне середовище тут є засобом, а не метою дослідження.

Ручне або напівавтоматизоване виконання перевірок має суттєві обмеження: залежність від оператора, складність точного повторення тестових умов, імовірність помилок при формуванні сигналів, обмежені можливості контролю часових критеріїв і значні витрати часу на документування. Особливо це проявляється для алгоритмів, у яких рішення про спрацювання захисту залежить від поєднання параметрів, дискретних ознак і часових умов [1, 10, 11].

У попередніх роботах було розглянуто автоматизацію перевірок електронних регуляторів ГТД [1], застосування стендів-імітаторів для відпрацювання та випробувань регуляторів [2, 12–14], математичне моделювання робочих процесів ГТД [15], HIL- і real-time-підходи до дослідження систем керування та паливних контурів газотурбінних двигунів [10, 16, 17], а також підходи до FADEC, виявлення відмов і керування резервуванням [11]. У зарубіжних джерелах

останніх років акцент змістився до поєднання transient-моделювання, HIL-верифікації та керування обмеженнями: огляд Yang et al. [3], робота Khamvilai et al. [4], дослідження Salehi і Montazeri-Gh [5], Kratz [6], Du et al. [7], Enalou et al. [8] та Foss [9] підтверджують, що перевірка захистів потребує не лише моделі, а й формалізованого сценарного контуру. Водночас ці джерела переважно описують стенд, модель або окрему випробувальну технологію, але не визначають склад графічного програмного середовища, призначеного для формування, виконання та документування сценаріїв перевірки захисних алгоритмів.

Аналіз попередніх досліджень

У попередніх дослідженнях стенди-імітатори розглядаються насамперед як апаратно-програмні комплекси, що відтворюють аналогові, дискретні та цифрові сигнали двигуна і дають змогу перевіряти взаємодію регулятора з імітованим об'єктом керування [2, 12, 14]. Ці роботи підтверджують доцільність стендово-імітаційного підходу, однак їх основний акцент зосереджено на складі стенда, відтворенні сигналів і випробуванні конкретних регуляторів, а не на структурі програмного середовища для побудови сценаріїв перевірки захистів.

Праці, присвячені метрологічному забезпеченню випробувань електронних регуляторів, деталізують питання імітації датчиків, точності вимірювальних каналів, атестації стендів і достовірності формування сигналів [13, 14]. Для даної роботи ці результати важливі як джерело вимог до точності, повторюваності та протоколювання випробувань. Водночас вони не розкривають, які функціональні вузли має містити графічний редактор сценаріїв і як ці вузли повинні взаємодіяти під час перевірки захисної логіки.

Окремий напрям становлять HIL- і real-time-підходи до дослідження систем керування ГТД, паливних контурів, виконавчих механізмів і моделей двигуна [10, 16, 17], а також дослідження FADEC із засобами виявлення відмов і керування резервуванням [11]. Матеріали HIL-досліджень підтверджують можливість безпечного відтворення критичних режимів, відмов датчиків і виконавчих ланцюгів у замкненому контурі. Роботи з FADEC додатково підкреслюють значення відмовостійкої логіки для безпечної роботи системи керування. Проте їх предметом зазвичай є модель двигуна, апаратна платформа або конкретний контур керування, тоді як задача формалізації сценарної перевірки алгоритмів захисту регулятора залишається розглянутою лише частково.

Таким чином, наявні джерела охоплюють стендово-імітаційну базу випробувань, метрологічне забезпечення та HIL-моделювання, але не дають повної формалізації програмного рівня, який поєднує опис сценарію, виконання, взаємодію зі стендом і критерії PASS/FAIL. Саме тому дана робота зосереджується на побудові такого середовища та на визначенні складу його функціональних вузлів.

Постановка задачі дослідження

Об'єктом дослідження є процес автоматизованої перевірки алгоритмів захисту регулятора газотурбінного двигуна в умовах стендово-імітаційного випробування. Предметом дослідження є функціональна структура програмного засобу, призначеного для графічного формування, виконання та документування сценаріїв такої перевірки.

Метою дослідження є обґрунтування функціональної структури програмного засобу автоматизованої перевірки алгоритмів захисту регулятора ГТД на основі аналізу функцій регулятора, типових захисних алгоритмів і вимог до стендово-імітаційного відтворення критичних режимів.

Для досягнення поставленої мети необхідно класифікувати алгоритми захисту, що підлягають автоматизованій перевірці; сформувати вимоги до автоматизованого сценарію; визначити групи даних, сигналів і команд, з якими має працювати програмний засіб; обґрунтувати склад функціональних вузлів; запропонувати узагальнену структуру сценарію перевірки та критерії оцінювання результату.

Методика дослідження базується на системному аналізі функцій електронного регулятора, узагальненні типових сценаріїв перевірки захисних алгоритмів і формалізації функціональних вимог до перспективного програмного засобу візуального та формального задання сценаріїв перевірки. При визначенні вимог враховано підходи до автоматизації випробувань регуляторів [1], принципи HIL-відтворення режимів і відмов у системах керування ГТД [10, 16, 17], а також сучасні зарубіжні підходи до transient-моделювання, HIL-верифікації та управління обмеженнями [3–9].

Результати

Функції регулятора як основа вимог до програмного засобу

Функціональну структуру програмного засобу доцільно визначати на основі задач, які виконує регулятор ГТД під час штатних, граничних і нештатних режимів роботи. Для автоматизованої перевірки найбільше значення мають функції, пов'язані з переходом системи з одного стану до іншого залежно від вхідних параметрів, дискретних сигналів, часових умов і результатів внутрішнього контролю.

ISSN 1727-7337 (print)
ISSN 2663-2217 (online)

Алгоритми захисту регулятора часто мають комбінований характер. Вони можуть включати перевірку порогового значення, підтвердження ознаки протягом заданого часу, аналіз стану інших сигналів, формування вихідної команди, перехід на резервний канал або завершення поточного режиму. Отже, мінімальний склад можливостей програмного засобу має включати роботу з аналоговими параметрами двигуна, обробку дискретних сигналів, формування тестових впливів і команд, контроль логічних і часових умов, реєстрацію реакції регулятора, протоколювання результатів і причин невідповідності [1].

З аналізу функцій регулятора випливають основні групи вимог до програмного засобу: робота з аналоговими параметрами двигуна; обробка дискретних сигналів; формування тестових впливів і команд; контроль логічних і часових умов; реєстрація реакції регулятора; протоколювання результатів і причин невідповідності. Ці вимоги формують основу для визначення складу функціональних вузлів програмного середовища.

Класифікація алгоритмів захисту, що підлягають автоматизованій перевірці

У межах цієї роботи алгоритми захисту доцільно розглядати як ширший клас функцій контролю та діагностування, оскільки їх спільною метою є виявлення, попередження або локалізація небезпечних станів регулятора ГТД. Водночас для задач автоматизованої перевірки практично доцільно використовувати не загальнотеоретичну, а робочу функціональну класифікацію, яка відображає спосіб організації сценарію перевірки, а не підміняє усталені терміни діагностики.

Функціонально-контрольні алгоритми аналізують штатні сигнали, порогові рівні та логічні умови в межах нормального режиму роботи. Для їх перевірки необхідно відтворити штатні або граничні значення параметрів, зафіксувати момент виконання умови та перевірити факт і час спрацювання. До цієї групи належать, зокрема, захисти за частотою обертання, температурою, тиском або крутним моментом.

Тестово-сценарні алгоритми залежать не тільки від абсолютного значення параметра, а й від характеру його зміни за часом і послідовності подій. У цьому випадку сценарій має відтворювати часову послідовність параметрів, контролювати швидкість зміни, тривалість ознаки та поєднання кількох умов. Такі сценарії корисні для перевірки як спрацювання за небезпечних умов, так і неспрацювання на допустимих граничних режимах.

Алгоритми, пов’язані з імітацією відмов і реконфігурацією, описують випадки втрати, спотворення, шуму або фіксації вимірювальних каналів і подальшого переходу системи до резервної структури. Для їх перевірки програмний засіб має підтримувати імітацію несправності, контроль формування ознаки відмови, перевірку переходу на інший канал і реєстрацію стану системи після реконфігурації.

Контрольно-тестові алгоритми доцільно розглядати не як протилежну до тестових алгоритмів групу, а як поєднання контрольного аналізу штатних сигналів і примусово сформованої тестової дії. Для таких перевірок важливими є запуск тестової процедури, контроль її етапів, аналіз результату і протоколювання виявлених невідповідностей.

Таблиця 1

Узагальнення вимог до перевірки основних груп алгоритмів захисту

Група алгоритмів	Що перевіряється	Необхідні функції програмного засобу
Функціонально-контрольні алгоритми аналізують штатні сигнали, порогові рівні та логічні умови у межах нормального режиму роботи. Для їх перевірки необхідно відтворити штатні або граничні значення параметрів, зафіксувати момент виконання умови та перевірити факт і час спрацювання. До цієї групи належать, зокрема, захисти за частотою обертання, температурою, тиском або крутним моментом.	Досягнення граничного значення частоти, температури, тиску або іншого параметра	Задання параметра, контроль порога, вимірювання часу реакції, реєстрація команди
Тестово-сценарні алгоритми залежать не тільки від абсолютного значення параметра, а й від характеру його зміни у часі та послідовності подій. У цьому випадку сценарій має	Зміна параметра або групи параметрів у часі, ознаки помпажу чи погасання	Формування часової послідовності, контроль швидкості зміни, логічні комбінації

Група алгоритмів	Що перевіряється	Необхідні функції програмного засобу
Алгоритми, пов'язані з імітацією відмов і реконфігурацією, описують випадки втрати, спотворення, шуму або фіксації вимірювальних каналів і подальшого переходу системи до резервної структури. Для їх перевірки програмний засіб має підтримувати імітацію несправності, контроль формування ознаки відмови, перевірку переходу на інший канал і реєстрацію стану системи після реконфігурації.	Відмова датчика, каналу, дискретного сигналу або виконавчого ланцюга	Імітація несправності, контроль ознаки відмови, повідомлення або блокування
Реконфігураційні	Перехід на дублюючий канал або резервний режим після підтвердженої відмови	Формування відмови, контроль послідовності переходу, перевірка стану після реконфігурації
Контрольно-тестові	Виконання процедури самоконтролю без створення реальної аварійної ситуації	Запуск тесту, контроль етапів, реєстрація результату і протоколювання

Вимоги до автоматизованого сценарію перевірки

Автоматизований сценарій перевірки доцільно розглядати як формалізовану послідовність дій, спрямовану на відтворення тестової ситуації, реєстрацію реакції регулятора та оцінювання її відповідності заданим критеріям. На відміну від ручної перевірки, такий сценарій має забезпечувати однозначність виконання, повторюваність умов, автоматичний контроль часових інтервалів і документування результатів. Така постановка узгоджується з вимогами до автоматизації випробувань регуляторів [1], HIL-відтворення режимів роботи ГТД [10, 16, 17] та підходами до FADEC і відмовостійкого керування [11].

Типовий сценарій включає ініціалізацію перевірки, встановлення початкових умов, формування тестового впливу, контроль досягнення умови спрацювання, очікування та реєстрацію реакції регулятора, перевірку критеріїв правильності, формування результату і протоколювання. Для кожного етапу потрібні відповідні функціональні дії: задання параметра, зміна параметра, очікування умови, перевірка логічного виразу, формування команди, зчитування сигналу, затримка, перехід до іншої гілки сценарію та запис повідомлення до журналу.

Отже, вимоги до сценарію безпосередньо визначають вимоги до програмного засобу. Якщо сценарій має працювати з параметрами, умовами, командами, затримками, логічними переходами, обладнанням і протоколом, то програмне середовище повинно містити відповідні функціональні вузли.

Важливою вимогою є також відокремлення структури сценарію від числових значень, які можуть змінюватися залежно від типу виробу, умов випробування або версії алгоритму. Тому сценарій має підтримувати використання зовнішніх таблиць параметрів, шаблонів перевірок і налаштувань, що дозволяє застосовувати одну логічну структуру для різних варіантів випробувань без переписування алгоритму.

Функціональна структура програмного засобу

На верхньому рівні програмний засіб доцільно розглядати як сукупність взаємопов'язаних шарів: сценарного редактора і бібліотеки вузлів; виконувального ядра з керуванням часом; адаптера обміну зі стендом-імітатором; підсистем валідації, реєстрації та протоколювання. Така декомпозиція відокремлює опис перевірки, часову логіку й апаратно-залежний обмін сигналами та дозволяє використовувати однакові сценарні шаблони для різних варіантів виробу.

Графічний редактор забезпечує формування сценарію у вигляді функціональних вузлів, пов'язаних лініями передачі потоку виконання та даних. Бібліотека вузлів має покривати мінімальний цикл перевірки: задання початкових умов, формування тестового впливу, контроль логічних і часових умов, реєстрацію реакції регулятора, оцінювання результату та формування протоколу.

Модуль виконання інтерпретує побудований сценарій, синхронізує часові затримки й умови переходів, передає команди до стенда-імітатора та приймає від нього аналогові, дискретні й діагностичні сигнали. Інтерфейс обміну з обладнанням має виконувати роль апаратної абстракції: сценарій оперує іменованими параметрами та командами, а конкретні канали введення-виведення задаються в конфігурації стенда.

ISSN 1727-7337 (print)
ISSN 2663-2217 (online)

Модуль валідації має перевіряти коректність побудованого сценарію до його запуску: наявність початкового і кінцевого вузлів, відсутність нез'єднаних гілок, визначеність змінних, сумісність типів даних, наявність обов'язкових параметрів і відсутність потенційних нескінченних циклів. Модуль логування та генератор протоколу фіксують ключові події, значення параметрів, час реакції, результат PASS/FAIL і причину невідповідності.

Таблиця 2

Основні групи функціональних вузлів програмного засобу

Група вузлів	Призначення	Приклади дій
Керування потоком	Формування структури сценарію	Початок, завершення, цикл, перехід, аварійний вихід
Дані та змінні	Збереження параметрів і проміжних результатів	Задання змінної, зміна значення, використання глобального параметра
Математична й логічна обробка	Обчислення, порівняння та перевірка умов	Поріг, формула, логічний вираз, розгалуження
Часова організація	Контроль затримок і часу реакції	Затримка, очікування події, вимірювання інтервалу
Команди та обладнання	Передавання впливів і обмін зі стендом	Подати команду, зчитати параметр, керувати джерелом живлення
Файли та протокол	Зберігання налаштувань і результатів	Імпорт параметрів, запис журналу, формування протоколу
Валідація та контроль помилок	Перевірка коректності сценарію і безпечне завершення при невідповідності	Перевірити зв'язність сценарію, типи даних і обов'язкові параметри; зупинити перевірку, записати причину, сформувати негативний результат

Узагальнена структура сценарію перевірки

Узагальнений сценарій перевірки алгоритму захисту включає вибір захисної функції, встановлення початкових умов, формування тестового впливу, контроль досягнення умови спрацювання, реєстрацію реакції регулятора, перевірку критеріїв правильності, формування результату та протоколювання. Кожний етап може бути реалізований окремим функціональним вузлом або їх комбінацією.

На етапі вибору захисної функції визначаються контрольовані параметри, вхідні впливи, очікувана реакція і критерії оцінювання. Початкові умови задають стан регулятора, стенда-імітатора, активних каналів, дискретних сигналів і засобів реєстрації. Тестовий вплив створює умову, за якої має спрацювати захист: зміну числового параметра, появу дискретної ознаки, імітацію відмови або запуск тестової процедури.

Після досягнення умови спрацювання сценарій фіксує момент її появи, контролює реакцію регулятора та визначає час реакції. Далі фактична реакція порівнюється з очікуваною за логічними, часовими та функціональними критеріями. Результат перевірки подається як «перевірку пройдено», «перевірку не пройдено», «перевірку не виконано» або «перевірку перервано».

Узагальнену структуру сценарію автоматизованої перевірки алгоритму захисту регулятора ГТД у контурі стенда-імітатора доцільно подати у вигляді блок-схеми, наведеної на рис. 1. Вона відображає логіку виконання конкретної перевірки, тоді як функціональна структура програмного засобу визначається модулями редактора, виконання, валідації, обміну зі стендом, реєстрації та протоколювання, описаними вище.

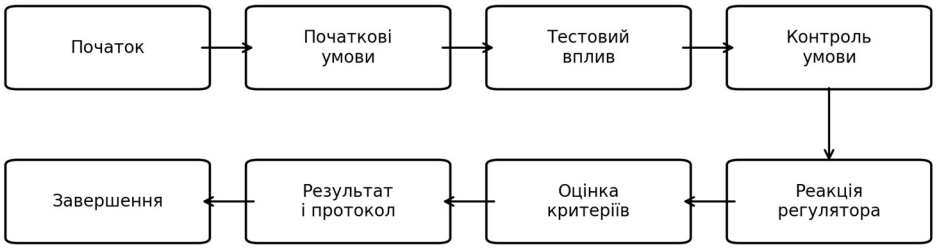

Рис. 1. Узагальнена структура сценарію автоматизованої перевірки алгоритму захисту регулятора ГТД

Запропонована структура є універсальною й може бути адаптована до різних груп алгоритмів захисту. Для параметричних алгоритмів основну роль відіграють блоки формування зміни параметра, контролю порогового значення та реєстрації команди захисту. Для динамічних алгоритмів важливими є блоки відтворення часової послідовності параметрів і аналізу швидкості їх зміни. Для діагностичних алгоритмів ключовими є блоки імітації несправності та контролю формування ознаки відмови. Для реконфігураційних алгоритмів структура сценарію має додатково включати контроль переходу системи на резервний канал або резервний режим роботи.

Таким чином, узагальнена структура сценарію перевірки дозволяє встановити зв'язок між функціями регулятора, типом алгоритму захисту, вимогами до тестового впливу та складом функціональних вузлів програмного засобу. Це забезпечує методичну основу для побудови різних сценаріїв автоматизованої перевірки в межах єдиного програмного середовища.

Критерії оцінювання результату перевірки

Автоматизована перевірка повинна завершуватися формалізованим висновком щодо правильності роботи захисного алгоритму. Критерії оцінювання мають враховувати логічну, часову та функціональну відповідність реакції регулятора очікуваному результату. Логічна відповідність означає спрацювання тільки за наявності заданої умови; часова відповідність — появу реакції у допустимому інтервалі; функціональна відповідність — формування саме тієї команди, ознаки або послідовності дій, яка передбачена алгоритмом.

До основних критеріїв належать досягнення умови спрацювання, наявність очікуваної реакції, допустимий час реакції, правильність команди або ознаки, відсутність хибного спрацювання, правильність послідовності дій, перехід системи у безпечний або заданий стан і повнота протоколювання. Невиконання хоча б одного з обов'язкових критеріїв має призводити до негативного результату із зазначенням причини невідповідності.

Для прикладу, під час перевірки захисту від розкрутки вільної турбіни сценарій задає початкове значення частоти обертання, формує її зростання до граничного значення, фіксує момент виконання умови, очікує появу ознаки або команди захисту, визначає час реакції та перевіряє перехід системи в безпечний стан. Навіть такий параметричний алгоритм потребує використання вузлів задання параметрів, умов, часової організації, реєстрації сигналів, оцінювання результату і протоколювання, що підтверджує доцільність графічного формування сценаріїв на основі уніфікованих функціональних блоків.

Для динамічних захистів критерії оцінювання мають додатково враховувати не тільки кінцеву реакцію регулятора, але й послідовність проміжних дій. Наприклад, при моделюванні помпажу або погасання камери згоряння необхідно контролювати появу ознаки, запуск відновлювальної дії, обмеження або припинення небезпечного режиму та відсутність неочікуваних команд. Такий підхід забезпечує перевірку не окремого сигналу, а логіки захисного алгоритму в цілому.

Узагальнений перелік критеріїв оцінювання результатів автоматизованої перевірки наведено в таблиці 3.

Таблиця 3

Критерії оцінювання результатів автоматизованої перевірки

Критерій	**Позитивний результат**	**Негативний результат**
Умова спрацювання	Параметр або ознака досягли заданої умови	Умова не сформована або сформована некоректно
Реакція регулятора	З'явилася очікувана команда, ознака або перехід стану	Очікувана реакція відсутня
Час реакції	Реакція сформована у допустимому інтервалі	Реакція запізнилася або виникла передчасно
Правильність дії	Команда або послідовність дій відповідає логіці захисту	Сформована неправильна або неповна реакція
Хибне спрацювання	За відсутності критичної умови захист не активується	Захист активувався без заданої умови
Протоколювання	Зафіксовано ключові параметри, події, час і висновок	Протокол неповний або не дозволяє відновити хід перевірки

ISSN 1727-7337 (print)
ISSN 2663-2217 (online)

Таким чином, критерії оцінювання результату є необхідною складовою функціональної структури програмного засобу. Вони дозволяють перетворити автоматизований сценарій із простої послідовності команд на інструмент об'єктивної перевірки алгоритмів захисту регулятора ГТД. Наявність формалізованих критеріїв забезпечує повторюваність оцінювання, зменшує вплив суб'єктивних рішень оператора та створює основу для автоматичного формування протоколу випробування.

Приклад застосування підходу

Для демонстрації запропонованого підходу доцільно розглянути перевірку захисту від розкрутки вільної турбіни на перехідних режимах запуску та зупинки ГТД. Імітаційна модель у такому випадку не повинна відтворювати повну фізику двигуна, але має відтворювати потрібні траєкторії параметрів, часові вікна, логічні переходи та граничні умови, що визначають область спрацювання захисту. Окремо слід перевіряти як спрацювання за небезпечних умов, так і неспрацювання на допустимих граничних режимах.

На першому етапі сценарію задаються початкові умови: вихідне значення частоти обертання, стан дискретних сигналів, активний канал керування, готовність стенда-імітатора і засобів реєстрації. Початкове значення параметра має бути нижчим за граничне, а режим, навантаження або фазу польоту слід задавати як окремі сценарні параметри відповідно до області активності конкретного захисту, щоб уникнути передчасного спрацювання до початку тестового впливу.

На другому етапі програмний засіб формує тестовий вплив, що імітує зростання частоти обертання вільної турбіни. Зміна параметра може задаватися поступово або стрибкоподібно. Поступове зростання зручне для точного визначення моменту досягнення порога, а стрибкоподібна зміна може використовуватися для перевірки реакції регулятора на швидке виникнення небезпечного стану. За наявності серії прогонів така структура сценарію дає змогу накопичувати статистику хибних спрацювань, пропусків спрацювання та, за потреби, формувати матрицю рішень або ROC-представлення без попереднього припущення про наявність числових результатів.

Умова спрацювання може бути подана у вигляді $n_{ст} \geq n_{ст.гр}$, де $n_{ст}$ – поточне значення частоти обертання вільної турбіни, а $n_{ст.гр}$ – граничне значення. Після виконання цієї умови сценарій фіксує момент $t_{ум}$, очікує появу ознаки або команди захисту, фіксує момент $t_{реакц}$ і визначає час реакції $\Delta t = t_{реакц} - t_{ум}$. Для оцінювання результату доцільно використовувати детерміновані критерії PASS/FAIL: факт спрацювання, допустиме вікно часу спрацювання, відповідність параметричних меж, безпечний перехід системи та повноту протоколу.

Перевірка вважається успішною, якщо гранична умова була досягнута, очікувана ознака або команда з'явилася у допустимий час, сформована реакція відповідає логіці алгоритму, а система перейшла у безпечний стан. Якщо реакція відсутня, запізнюється, має неправильний характер або не припиняє розвиток небезпечної ситуації, сценарій формує негативний результат із зазначенням причини. Повноцінне ймовірнісне оцінювання потребує серії сценаріїв і прогонів, щоб надалі обчислювати частоту хибних спрацювань, пропусків спрацювання та пов'язані з ними показники якості.

Розглянутий приклад показує, що навіть відносно простий захист потребує використання різних груп функціональних вузлів: задання параметрів, формування тестового впливу, перевірки умов, контролю часу, реєстрації сигналів, оцінювання результату, протоколювання та, за потреби, імітації втрати, спотворення, шуму або фіксації вимірювальних каналів. Це підтверджує доцільність побудови програмного засобу за принципом графічного формування сценаріїв на основі уніфікованих функціональних блоків.

Висновки

У роботі запропоновано узагальнену функціональну структуру програмного засобу автоматизованої перевірки алгоритмів захисту регулятора газотурбінного двигуна як каркас, у якому зв'язки між вузлами задаються відповідно до конкретного алгоритму перевірки. Структуру подано як поєднання модулів редактора сценаріїв, бібліотеки вузлів, валідації, виконання, обміну зі стендом, реєстрації даних, протоколювання та обробки помилок.

Класифікацію алгоритмів захисту подано як робочу функціональну класифікацію для задач автоматизованої перевірки: функціонально-контрольні, тестово-сценарні, реконфігураційні та контрольно-тестові алгоритми. Для кожної групи визначено тип тестового впливу, контрольовану умову та характер очікуваної реакції регулятора, а сам поділ використано для узгодження поведінки алгоритму з вимогами до сценарію перевірки.

Визначено сім основних груп функціональних вузлів: керування потоком виконання, дані та змінні, математична й логічна обробка, часова організація, команди та обладнання, файли і протокол, валідація та контроль помилок. Такий набір вузлів забезпечує побудову сценарію від ініціалізації до формування підсумкового висновку, але конкретні зв'язки між ними мають задаватися сценарієм перевірки.

Сформовано узагальнений сценарій перевірки у вигляді послідовності функціональних блоків: початкові умови, тестовий вплив, контроль умови, реакція регулятора, оцінка критеріїв, результат і протокол. Для оцінювання результату виділено шість узагальнених критеріїв: умова спрацювання, реакція регулятора, час реакції, правильність дії, відсутність хибного спрацювання та протоколювання.

Отримані результати можуть бути використані як методична основа для подальшої реалізації програмного засобу у складі стендово-імітаційного комплексу. Подальша робота має бути спрямована на формалізацію інтерфейсів вузлів, наповнення бібліотеки типових сценаріїв, реалізацію модуля валідації та експериментальну перевірку запропонованої структури на реальних або стендових алгоритмах захисту.

Конфлікт інтересів

Автор заявляє, що немає конфлікту інтересів щодо цього дослідження, фінансового, особистого, авторського чи іншого, який міг би вплинути на дослідження та його результати, представлені в цій статті.

Фінансування

Дослідження проводилося без фінансової підтримки.

Наявність даних

Рукопис не має супутніх даних.

Використання штучного інтелекту

Автор підтверджує, що не використовував технології штучного інтелекту при створенні даної роботи.

Автор прочитав і погодилися з опублікованою версією рукопису.

Література

1. Бурунов, Д. С. Автоматизація випробувань електронних регуляторів ГТД / Д. С. Бурунов, А. О. Таранішин // Авіаційно-космічна техніка і технологія. – 2018. – № 8 (152). – С. 108–112. DOI: https://doi.org/10.32620/aktt.2018.8.16.

2. Development of the helicopter turboshafted engine regulator / G. Ranchenko, A. Buryachenko, V. Grudinkin et al. // Aerospace Technic and Technology. – 2023. – № 4 sup 1 (189). – P. 52–57. DOI: https://doi.org/10.32620/aktt.2023.4sup1.07.

3. Gas turbine engine transient performance and heat transfer effect modelling: A comprehensive review, research challenges, and exploring the future / Y. Yang, T. Nikolaidis, S. Jafari, P. Pilidis // Applied Thermal Engineering. – 2024. – Vol. 236. – 24 p. DOI: https://doi.org/10.1016/j.applthermaleng.2023.121523.

4. Hardware-in-the-Loop Simulation Testbed Development for Distributed Turbine Engine Control Systems / T. Khamvilai, M. Pakmehr, G. Lu et al. // AIAA SCITECH 2022 Forum. – 2022. – 10 p. DOI: https://doi.org/10.2514/6.2022-1233.

5. Salehi, A. Design and HIL-based verification of the fuel control unit for a gas turbine engine / A. Salehi, M. Montazeri-GH // Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering. – 2020. – Vol. 234, iss. 8. – P. 1460–1470. DOI: https://doi.org/10.1177/0954410020910593.

6. Kratz, J. L. Transient Optimization of a Gas Turbine Engine / J. L. Kratz // NASA Technical Reports Server. – 2023. Available et: https://ntrs.nasa.gov/citations/20220016302 (дата звернення: 17.07.2026). – Назва з екрана.

7. An Output-Based Limit Protection Strategy for Turbofan Engine Propulsion Control with Output Constraints / J. Qin, M. Pan, W. Xu, J. Huang // Energies. – 2019. – Vol. 12, iss. 21. – 24 p. DOI: https://doi.org/10.3390/en12214043.

8. Nikolaidis, T. Advanced Constraints Management Strategy for Real-Time Optimization of Gas Turbine Engine Transient Performance / T. Nikolaidis, Z. Li, S. Jafari // Applied Sciences. – 2019. – Vol. 9, iss. 24. – 26 p. DOI: https://doi.org/10.3390/app9245333.

9. Foss, S. J. Overspeed Protection System: Design, Verification and Testing / S. J. Foss // Nexus Controls, a Baker Hughes business : technical white paper. – 2020. – Available et: https://www.bakerhughes.com/sites/bakerhughes/files/2023-02/bhcs38712_overspeed_protection_system_wp_r2.pdf (дата звернення: 17.07.2026). – Назва з екрана.

10. Kulikov, G. G. Hardware-in-the-loop testing technology for integrated control and condition monitoring systems of aircraft gas turbine engines / G. G. Kulikov, V. Yu. Ar'kov, A. I. Abdulnagimov // Russian Aeronautics. – 2008. – Vol. 51, iss. 1. – P. 47–52. DOI: https://doi.org/10.3103/S106879980801008X.

11. Yamane, H. A Study on Aircraft Engine Control Systems for Integrated Flight and Propulsion Control. Part 1: FADEC System and Engine Control / H. Yamane, Y. Matsunaga, T. Kusakawa // Journal of the Japan Society for Aeronautical and Space Sciences. – 2008. – Vol. 56, iss. 649. – P. 80–87. DOI: https://doi.org/10.2322/jjsass.56.80.

12. Стенд-імітатор авіадвигуна як універсальний засіб забезпечення розробки та випробувань регуляторів / А. Г. Буряченко, Д. С. Бурунов, В. М. Грудінкін, А. О. Таранішин // Авіаційно-космічна техніка і технологія. – 2020. – № 7 (167). – С. 83–88. DOI: https://doi.org/10.32620/aktt.2020.7.12.

13. Буряченко, А. Г. Метрологічне забезпечення випробувань електронних регуляторів ГТД – стенд-імітатор двигунних датчиків / А. Г. Буряченко, І. К. Лопащенко // Авіаційно-космічна техніка і технологія. – 2018. – № 7 (151). – С. 101–106. DOI: https://doi.org/10.32620/aktt.2018.7.15.

ISSN 1727-7337 (print)
ISSN 2663-2217 (online)

14. Буряченко, А. Г. Стенд-імітатор турбовального двигуна АІ-450М для випробувань регулятора двигуна. Метрологічне забезпечення та атестація стенда / А. Г. Буряченко, В. М. Грудінкін, Д. С. Бурунов // Вісник двигателебудування. – 2015. – № 2. – С. 95–101.

15. Математичне моделювання робочих процесів авіаційного газотурбінного двигуна ТВ3-117 для контролю і діагностики його технічного стану / С. І. Владов, Ю. М. Шмельов, Л. М. Пилипенко та ін. // Вісник Херсонського національного технічного університету. – 2020. – № 1(72), ч. 1. – С. 18–34.

16. Zhang, H. Hardware-in-the-Loop Simulation Study on the Fuel Control Strategy of a Gas Turbine Engine / H. Zhang, M. Su, S. Weng // Journal of Engineering for Gas Turbines and Power. – 2005. – Vol. 127, iss. 3. – P. 693–695. DOI: https://doi.org/10.1115/1.1805012.

17. Application of Neural Network Technology and High-performance Computing for Identification and Real-time Hardware-in-the-loop Simulation of Gas Turbine Engines / G. I. Pogorelov, G. G. Kulikov, A. I. Abdulnagimov, B. I. Badamshin // Procedia Engineering. – 2017. – Vol. 176. – P. 402–408. DOI: https://doi.org/10.1016/j.proeng.2017.02.338.

References

1. Burunov, D. S., Taranishin, A. O. Avtomatizatsiya ispytaniy elektronnykh regulyatorov GTD [Automation of verification of electronic GTE regulators]. *Aviacijno-kosmicna tehnika i tehnologia – Aerospace Technic and Technology*, 2018, no. 8 (152), pp. 108–112. DOI: https://doi.org/10.32620/aktt.2018.8.16 (in Ukrainian).

2. Ranchenko, G., Buryachenko, A., Grudinkin, V., Burunov, D., Danilov, V. Development of the helicopter turboshafted engine regulator. *Aerospace Technic and Technology*, 2023, no. 4, sup 1 (189), pp. 52–57. DOI: https://doi.org/10.32620/aktt.2023.4sup1.07.

3. Yang, Y., Nikolaidis, T., Jafari, S., Pilidis, P. Gas turbine engine transient performance and heat transfer effect modelling: A comprehensive review, research challenges, and exploring the future. *Applied Thermal Engineering*, 2024, vol. 236. 24 p. DOI: https://doi.org/10.1016/j.applthermaleng.2023.121523.

4. Khamvilai, T., Pakmehr, M., Lu, G., Yang, Y., Feron, E. M., Behbahani, A. R. Hardware-in-the-Loop Simulation Testbed Development for Distributed Turbine Engine Control Systems. *AIAA SCITECH 2022 Forum*, 2022. 10 p. DOI: https://doi.org/10.2514/6.2022-1233.

5. Salehi, A., Montazeri-Gh, M. Design and HIL-based verification of the fuel control unit for a gas turbine engine. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 2020, vol. 234, no. 8, pp. 1460–1470. DOI: https://doi.org/10.1177/0954410020910593.

6. Kratz, J. L. *Transient Optimization of a Gas Turbine Engine. NASA Technical Reports Server*, 2023. Available at: https://ntrs.nasa.gov/citations/20220016302 (accessed 17 July 2026).

7. Qin, J., Pan, M., Xu, W., Huang, J. An Output-Based Limit Protection Strategy for Turbofan Engine Propulsion Control with Output Constraints. *Energies*, 2019, vol. 12, iss. 21. 24 p. DOI: https://doi.org/10.3390/en12214043.

8. Nikolaidis, T., Li, Z., Jafari, S. Advanced Constraints Management Strategy for Real-Time Optimization of Gas Turbine Engine Transient Performance. *Applied Sciences*, 2019, vol. 9, iss. 24. 26 p. DOI: https://doi.org/10.3390/app9245333.

9. Foss, S. J. *Overspeed Protection System: Design, Verification and Testing. Technical White Paper. Nexus Controls, a Baker Hughes business*, 2020. Available at: https://www.bakerhughes.com/sites/bakerhughes/files/2023-02/bhcs38712_overspeed_protection_system_wp_r2.pdf (accessed 17 July 2026).

10. Kulikov, G. G., Ar'kov, V. Yu., Abdulnagimov, A. I. Hardware-in-the-loop testing technology for integrated control and condition monitoring systems of aircraft gas turbine engines. *Russian Aeronautics*, 2008, vol. 51, no. 1, pp. 47–52. DOI: https://doi.org/10.3103/S106879980801008X.

11. Yamane, H., Matsunaga, Y., Kusakawa, T. A study on aircraft engine control systems for integrated flight and propulsion control. Part 1: FADEC system and engine control. Journal of the Japan Society for Aeronautical and Space Sciences, 2008, vol. 56, no. 649, pp. 80–87. DOI: https://doi.org/10.2322/jjsass.56.80.

12. Buryachenko, A. G., Burunov, D. S., Grudinkin, V. M., Taranishin, A. O. Stend-imitator aviadvigatelya kak universal'noye sredstvo obespecheniya razrabotki i ispytaniy regulyatorov [Stand imitator of the engine as a universal means of ensuring the development and testing of regulators]. *Aviacijno-kosmicna tehnika i tehnologia – Aerospace Technic and Technology*, 2020, no. 7 (167), pp. 83–88. DOI: https://doi.org/10.32620/aktt.2020.7.12 (in Ukrainian).

13. Buryachenko, A. G., Lopashchenko, I. K. Metrologicheskoe obespechenie ispytaniy elektronnykh regulyatorov GTD – stend-imitator dvigatel'nykh datchikov [Metrological support for testing electronic GTE regulators – engine sensor simulator stand]. *Aviacijno-kosmicna tehnika i tehnologia – Aerospace Technic and Technology*, 2018, no. 7 (151), pp. 101–106. DOI: https://doi.org/10.32620/aktt.2018.7.15 (in Ukrainian).

14. Buryachenko, A. G., Grudinkin, V. M., Burunov, D. S. Stend-imitator turboval'nogo dvigatelya AI-450M dlya ispytaniy regulyatora dvigatelya. Metrologicheskoe obespechenie i attestatsiya stenda [Turboshaft engine AI-450M simulator for engine regulator testing. Metrological support and stand certification]. *Visnyk dvyhunobuduvannia – Herald of aeroenginebuilding*, 2015, no. 2, pp. 95–101 (in Russian).

15. Vladov, S. I., Shmelov, Yu. M., Pylypenko, L. M., Podhornykh, N. V., Nazarenko, N. P., Tutova, N. V., Dieriabina, I. O. Matematychne modeliuvannia robochykh protsesiv aviatsiinoho hazoturbinnoho

dvyhuna TV3-117 dlia kontroliu i diahnostyky yoho tekhnichnoho stanu [Mathematical modeling of operating processes in the TV3-117 aviation gas turbine engine for technical condition monitoring and diagnostics]. *Visnyk Khersonskoho natsionalnoho tekhnichnoho universytetu – Visnyk of Kherson National Technical University*, 2020, no. 1 (72), part 1, pp. 18–34. DOI: https://doi.org/10.35546/kntu2078-4481.2020.1.1.2. (in Ukrainian).

16. Zhang, H., Su, M., Weng, S. Hardware-in-the-Loop Simulation Study on the Fuel Control Strategy of a Gas Turbine Engine. *Journal of Engineering for Gas Turbines and Power*, 2005, vol. 127, iss. 3, pp. 693–695. DOI: https://doi.org/10.1115/1.1805012.

17. Pogorelov, G. I., Kulikov, G. G., Abdulnagimov, A. I., Badamshin, B. I. Application of neural network technology and high-performance computing for identification and real-time hardware-in-the-loop simulation of gas turbine engines. *Procedia Engineering*, 2017, vol. 176, pp. 402–408. DOI: https://doi.org/10.1016/j.proeng.2017.02.338.

Надійшла до редакції 09.07.2026, отримано в доопрацьованому вигляді 30.07.2026.
Дата ухвалення 19.08.2026, дата публікації 04.09.2026

SUBSTANTIATION OF THE FUNCTIONAL STRUCTURE OF A SOFTWARE TOOL FOR AUTOMATED VERIFICATION OF GAS TURBINE ENGINE REGULATOR PROTECTION ALGORITHMS AND FUNCTIONS

Serhii Vialov

The article considers an approach to substantiating the functional structure of a software tool for automated verification of gas turbine engine regulator protection algorithms and functions. The relevance of the study is driven by the need to safely and reproducibly verify critical start-up and shut-down transients, as direct engine testing is risky and difficult to standardize. The aim of the study is to substantiate the functional structure of a prospective software tool intended for visual and formal scenario authoring, execution, and documentation of automated verification scenarios. The research methodology is based on the systematic analysis of regulatory functions, the generalization of typical protection algorithms, the definition of requirements for automated test scenarios, and the formalization of the main groups of functional nodes. A classification of protection algorithms is proposed: parametric, dynamic, diagnostic, reconfiguration, and self-test algorithms. It is shown that the software tool should provide a minimal verification workflow: initial condition setting, generation of test inputs, processing of analog and discrete signals, verification of logical and timing conditions, registration of regulator response, interaction with a bench simulator, and report generation. The expediency of representing a verification scenario as a set of functional nodes is substantiated. The main node groups include flow control, variables and data, mathematical and logical processing, time control, command generation, equipment interaction, file operations, logging, reporting, validation, and error handling. A generalized scenario structure and criteria for evaluating the results are proposed. The results obtained can serve as a basis for further development of software tools for the automated verification of gas turbine engine regulators as part of a bench-simulation complex.

Keywords: gas turbine engine; GTE regulator; protection algorithms; automated verification; software tool; visual scenario authoring; bench simulator; functional structure.

Вялов Сергій Русланович – аспірант, НУ «Одеська політехніка», інженер-електронік, АТ «Елемент», Одеса, Україна.

Serhii Vialov – Postgraduate Student, Odesa Polytechnic University, Electronic Engineer, JSC "Element", Odesa, Ukraine, e-mail: vialov0608@gmail.com. ORCID: 0009-0006-7784-889.