

С. Р. ВЯЛОВ

АТ «Елемент», Одеса, Україна

Національний університет «Одеська політехніка», Одеса, Україна

ЗАСТОСУВАННЯ ВІЗУАЛЬНОГО ПРОГРАМУВАННЯ ДЛЯ АВТОМАТИЧНИХ ВИПРОБУВАНЬ РЕГУЛЯТОРІВ ГТД

Розглянуто різні методи програмування, що використовуються для розробки систем автоматичних випробувань регуляторів газотурбінних двигунів. Проведено детальний аналіз текстового, візуального та скриптового методів програмування, зокрема, з'ясовано їхні переваги та недоліки в контексті автоматичних випробувань. Текстове програмування забезпечує високу гнучкість та потужність, дозволяючи створювати оптимізовані алгоритми для складних завдань. Візуальне програмування надає інтуїтивність та зручність у розробці, зменшуючи ймовірність помилок та скорочуючи час розробки за рахунок використання графічних елементів. Скриптове програмування відзначається швидкістю та легкістю налаштування, що робить його зручним для автоматизації рутинних задач та швидких ітерацій. Проведено порівняльний аналіз ефективності цих методів у різних контекстах автоматичних тестувань, що включало оцінку продуктивності, надійності, масштабованості та зручності впровадження. Представлено результати експериментальних досліджень, що підтверджують ефективність кожного методу для конкретних завдань у системах автоматичних випробувань регуляторів газотурбінних двигунів. Зокрема, з'ясовано, що текстове програмування є оптимальним для складних та критичних систем, де необхідна висока продуктивність та можливість детального контролю. Скриптове програмування виявилось найбільш придатним для автоматизації рутинних задач та швидкого налаштування тестових сценаріїв. Візуальне програмування підходить для швидкої розробки та зручності використання, особливо коли важлива інтуїтивність та зменшення ймовірності помилок. Висновки надають рекомендації щодо вибору найбільш оптимального методу програмування для конкретних завдань у системах автоматичних випробувань. Зокрема, для реалізації автоматичних перевірок блоків розробки АТ «Елемент» доцільно обрати метод візуального програмування, враховуючи його переваги у простоті інтерпретації створюваних алгоритмів, демонстрації поточних значень параметрів у режимі реального часу та гнучкості редагування алгоритмів без необхідності компіляції після внесення змін. Отримані результати сприятимуть підвищенню точності, швидкості та надійності процесів автоматичних випробувань регуляторів газотурбінних двигунів.

Ключові слова: ГТД; автоматизація; візуальне програмування системи автоматичного тестування; оптимізація алгоритмів.

Вступ

Системи автоматичних випробувань регуляторів газотурбінних двигунів (ГТД) відіграють важливу роль у забезпеченні надійності та ефективності цих складних технічних пристроїв. Вони дозволяють проводити ретельну перевірку функціонування різних компонентів регуляторів газотурбінних двигунів при різних умовах експлуатації. Ефективність таких систем значною мірою залежить від методів програмування, що використовуються для їх розробки та впровадження.

У сучасній практиці існує кілька підходів до програмування систем автоматичних випробувань регуляторів газотурбінних двигунів, зокрема текстове програмування, візуальне програмування та

скриптове програмування. Текстове програмування, яке передбачає написання коду на мовах програмування високого рівня, таких як C++, забезпечує високу гнучкість та потужність. Візуальне програмування, що базується на використанні графічних елементів для створення програм, пропонує зручність та інтуїтивність у розробці, зменшуючи ймовірність помилок і скорочуючи час розробки. Скриптове програмування, що використовує скриптові мови, забезпечує швидкість і легкість у налаштуванні та модифікації тестових сценаріїв, що робить його особливо зручним для автоматизації рутинних задач.

Ця стаття присвячена порівнянню методів програмування для систем автоматичних випробувань регуляторів газотурбінних двигунів.

Постановка задачі дослідження

Об'єктом дослідження є методологія та інструменти програмування, які використовуються для автоматизації випробувань регуляторів газотурбінних двигунів.

Предметом дослідження є ефективність, надійність та економічність застосування візуального програмування в контексті автоматичних випробувань регуляторів газотурбінних двигунів.

Метою цієї наукової статті є проведення порівняльного аналізу методів програмування, зокрема текстового, візуального та скриптового методів, які застосовуються в системах автоматичних випробувань регуляторів газотурбінних двигунів. У статті будуть досліджені різні підходи до програмування, включаючи їхні переваги, недоліки та ефективність у контексті тестування та валідації регуляторів газотурбінних двигунів. Аналіз включатиме оцінку продуктивності, надійності, масштабованості та зручності впровадження кожного методу. У роботі представлено аналіз теоретичних аспектів кожного підходу, а також результати експериментальних досліджень, що демонструють їхню практичну ефективність. Отримані результати допоможуть визначити оптимальні методи програмування, які можуть бути рекомен-

довані для підвищення точності, швидкості та надійності процесів автоматичних випробувань регуляторів ГТД.

Порівняльний аналіз методів програмування

Вибір методу програмування для систем автоматичних випробувань регуляторів газотурбінних двигунів залежить від низки факторів, таких як складність завдання, навички користувачів, що працюватимуть із системою, вимоги до продуктивності, гнучкості та можливості інтеграції з іншими системами.

Необхідно виконати порівняльний аналіз трьох підходів: візуального програмування, скриптового програмування та традиційного (текстового) програмування.

1. Візуальне програмування

Візуальне програмування – процес подання програмних структур (алгоритмів і даних) за допомогою багатовимірних елементів (значків, форм, блоків) з використанням візуальної мови програмування [1]. Приклад побудови алгоритмів в LabView наведено на рисунку 1, приклад побудови в BSimple – на рисунку 2.

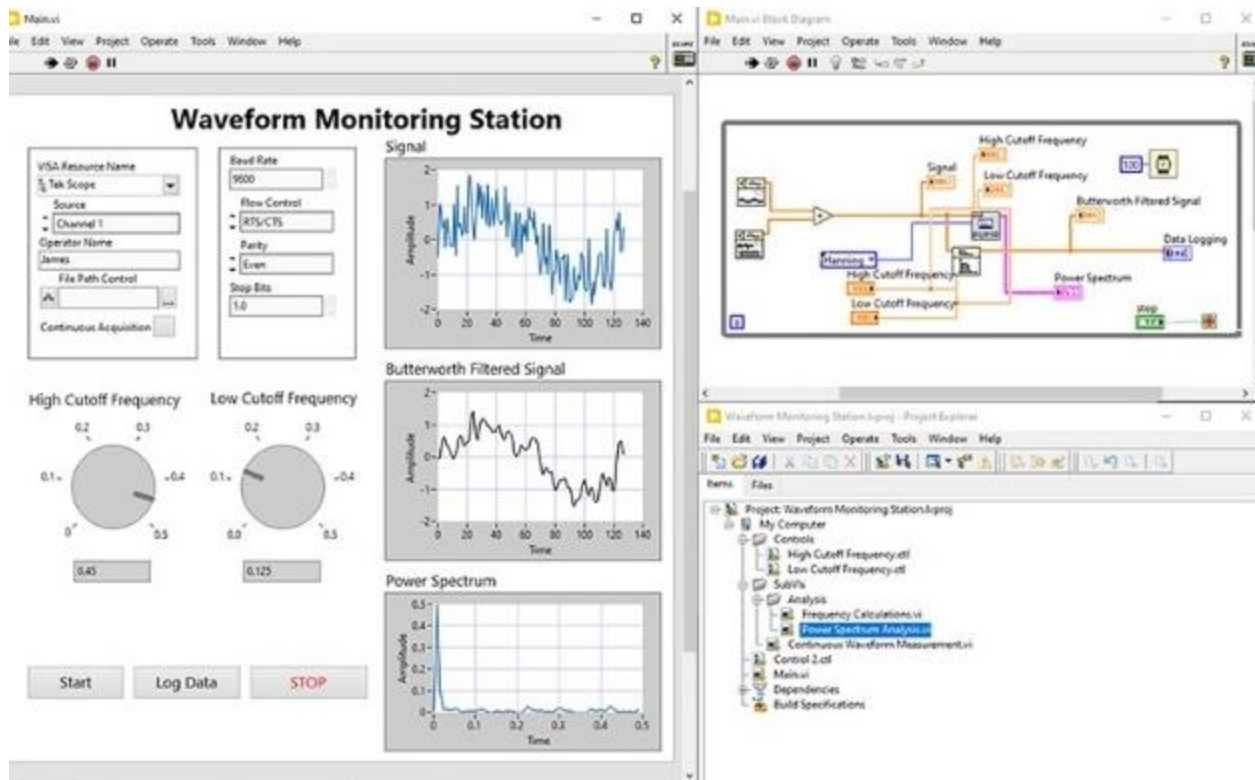


Рис. 1. Приклад побудови алгоритмів в LabView

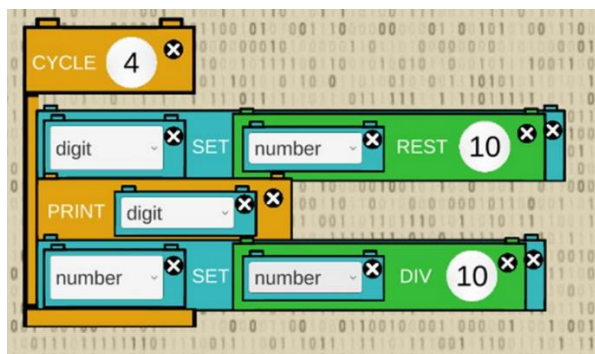


Рис. 2. Приклад побудови алгоритмів в BSimple

Серед основних переваг є зручність використання. Завдяки графічним блокам та інтерфейсам, користувачі з невеликим досвідом програмування можуть легко створювати та редагувати програми. Наприклад, інструменти як LabVIEW від National Instruments широко застосовуються для розробки систем автоматичних випробувань завдяки їх інтуїтивно зрозумілому графічному інтерфейсу [2]. Крім того, швидкість розробки збільшується завдяки наявності готових модулів і бібліотек для типових завдань, що особливо корисно на етапі прототипування або при швидкому налаштуванні випробувальних стендів. Графічне відображення програми також дозволяє зменшити кількість синтаксичних помилок, оскільки більшість дій виконується через інтуїтивно

зрозумілий інтерфейс, що знижує ризик помилок, пов'язаних з некоректним написанням коду.

Проте, візуальне програмування має й певні недоліки. Воно може бути обмежене у функціональності, що робить його менш придатним для складних алгоритмічних завдань або специфічних налаштувань системи. Продуктивність таких програм може бути менш ефективною з точки зору використання системних ресурсів, що критично для систем, які працюють у реальному часі та потребують високої продуктивності. Інтеграція з іншими системами або бібліотеками також може бути складнішою через обмеження, що накладаються середовищем візуального програмування. Наприклад, підключення специфічних сенсорів або використання унікальних комунікаційних протоколів може вимагати додаткових зусиль.

2. Скриптове програмування

Скриптове програмування - гнучкий та зручний метод для швидких ітерацій та інтеграції з іншими системами. Велика кількість доступних бібліотек робить його привабливим вибором для багатьох задач, але менша продуктивність може стати обмеженням для високопродуктивних систем. Приклад побудови алгоритмів в MethanOl наведений на рисунку 3.

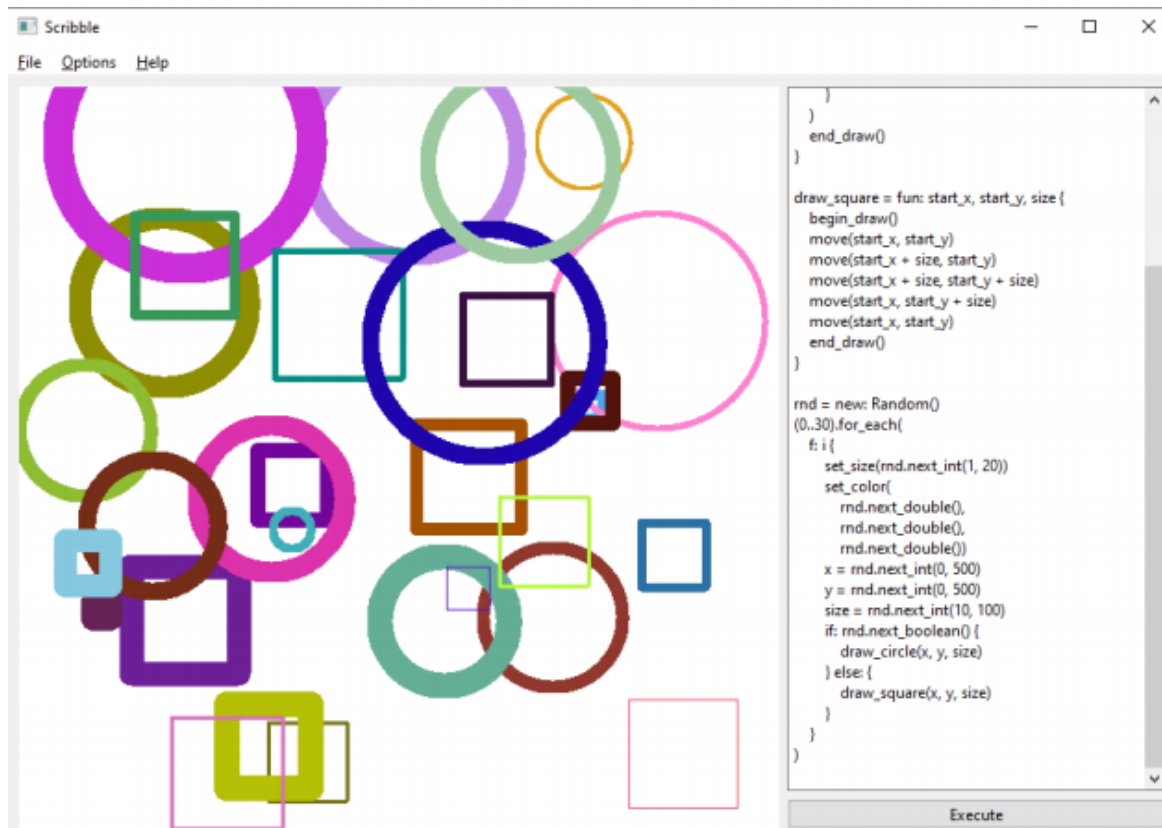


Рис. 3. Приклад побудови алгоритмів в MethanOl

До переваг належить гнучкість, яка дозволяє легко писати та змінювати код. Це особливо корисно для швидких ітерацій та експериментів, що важливо при розробці та тестуванні нових функцій або випробувальних сценаріїв. Скриптові мови, такі як Python або MethanOl забезпечують цю гнучкість [4,5]. Крім того, існує велика кількість доступних бібліотек і фреймворків для скриптових мов, що можуть бути використані для вирішення різноманітних завдань: від обробки даних і візуалізації до автоматизації процесів та інтеграції з іншими системами. Простота інтеграції також є важливою перевагою, оскільки скриптові мови часто мають хорошу підтримку для інтеграції з іншими системами та технологіями. Наприклад, Python має бібліотеки для роботи з мережею, базами даних, апаратними інтерфейсами, що дозволяє легко підключати додаткові компоненти до системи автоматичних випробувань.

Однак, скриптові мови мають і недоліки. Продуктивність скриптових мов зазвичай нижча, оскільки вони інтерпретуються, що може призводити до зниження швидкості виконання порівняно з компільованими мовами. Це може бути критичним для завдань, що вимагають високої швидкості виконання та обробки великої кількості даних у реальному часі. Крім того, управління великими проектами може стати менш зручним через необхідність керування великою кількістю скриптів і залежностей, що може ускладнити підтримку та масштабування проекту.

3. Традиційне програмування (компільовані мови)

Традиційне програмування забезпечує високу продуктивність та надійність, що є критичними для систем реального часу. Проте, складність розробки та підтримки вимагає високої кваліфікації програмістів і може збільшувати час розробки.

Серед переваг компільованих мов можна виділити високу продуктивність. Вони забезпечують ефективне використання ресурсів і високу швидкість обробки даних, що є важливим для систем, які працюють в реальному часі. Крім того, традиційні мови програмування надають повну гнучкість у реалізації будь-яких алгоритмів і структур даних, що дозволяє створювати оптимізовані та спеціалізовані рішення для конкретних завдань. Надійність також є значною перевагою, оскільки ці мови підходять для критичних систем, де важлива стабільність і надійність. Вони забезпечують детальний контроль над пам'яттю та ресурсами, що дозволяє уникнути багатьох помилок на рівні апаратного забезпечення.

Однак, компільовані мови мають і певні недоліки. Вони відрізняються складністю, вимагаючи ви-

сокої кваліфікації програмістів. Це може бути проблемою для команд із різноманітним рівнем підготовки, збільшуючи час навчання нових членів команди та ускладнюючи процес розробки. Крім того, час розробки на традиційних мовах може бути довшим через необхідність написання більшого обсягу коду та його відладки, що уповільнює процес розробки та впровадження нових функцій. Менш інтуїтивний процес розробки також є недоліком, оскільки це ускладнює спілкування між членами команди та обмін знаннями, що може бути проблемою при великій команді або при необхідності швидкого навчання нових співробітників.

Результати порівняльного аналізу

Для вибору методу програмування були враховані наступні чинники:

1. Складність і масштаб проекту. Для невеликих та середніх проектів з високою вимогою до зручності та швидкості розробки можна використовувати візуальне програмування. Для середніх та великих проектів, що потребують гнучкості та швидкої інтеграції з іншими системами, скриптове програмування буде підходящим вибором. Для великих проектів із високими вимогами до продуктивності та надійності найкраще підходить традиційне програмування.

2. Рівень підготовки команди. Команди з різноманітним рівнем підготовки можуть ефективніше використовувати візуальне або скриптове програмування. Висококваліфіковані програмісти можуть максимально використовувати переваги традиційного програмування.

3. Вимоги до продуктивності. Для завдань, що потребують високої продуктивності, необхідно вибрати традиційне програмування. Для завдань, де продуктивність не є критичною, можна обрати скриптове або візуальне програмування.

4. Інтеграція та масштабованість. Для систем, що потребують легкої інтеграції та розширюваності, скриптове програмування може бути найбільш підходящим.

В результаті порівняльного аналізу було отримано висновок щодо ефективності використання візуального програмування з метою автоматизації випробувань регуляторів газотурбінних двигунів. Доцільною є реалізація даного методу при проведенні випробувань регуляторів розробки АТ «Елемент». Дану гіпотезу було сформовано на основі наступних факторів:

- Розробка алгоритмів автоматичної перевірки може виконуватись персоналом без навичок традиційного програмування.

- Простота інтерпретації створюваних алгоритмів внаслідок представлення їх у графічному виді;
- Демонстрація поточних значень параметрів в реальному часі;
- Гнучкість редагування алгоритмів перевірки внаслідок відсутності компіляції після внесення змін в алгоритм, що дозволяє позбавитись використання додаткових середовищ розробки.

Висновки

На основі проведеного аналізу можна зробити висновки щодо вибору методу програмування для систем автоматичних випробувань регуляторів газотурбінних двигунів.

Текстове програмування є найбільш оптимальним вибором для складних і критичних систем, де необхідна висока продуктивність та можливість детального контролю. Воно дозволяє створювати надійні та ефективні програми, але вимагає значного часу на розробку та глибоких знань мов програмування.

Скриптове програмування виявилось найбільш придатним для автоматизації рутинних задач та швидкого налаштування тестових сценаріїв. Воно забезпечує високу гнучкість у модифікації та налагодженні, але може бути менш ефективним для складних задач, що потребують високої продуктивності та масштабованості.

Візуальне програмування підходить для швидкої розробки та зручності використання, особливо коли важлива інтуїтивність та зменшення ймовірності помилок. Цей метод є ефективним для задач середньої складності, де важливо швидко створювати та модифікувати тестові сценарії.

Для реалізації автоматичних перевірок блоків розробки АТ «Елемент», що призначені для управління авіаційними двигунами, доцільно обрати метод візуального програмування. Дане рішення характеризується рядом переваг, серед яких основними є простота інтерпретації створюваних алгоритмів, демонстрація поточних значень параметрів у режимі реального часу, гнучкість редагування алгоритмів, а також відсутність вимог до володіння навичками традиційного програмування персоналом.

Задача проведення подальших досліджень у даному напрямку є актуальною, і буде розвинена в наступних роботах.

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 та погодився з опублікованою версією рукопису.

Література

1. Осадча, К. П. Аналіз понятійного апарату парадигми візуального програмування [Текст] / К. П. Осадча, М. С. Марчук // Науковий вісник Мелітопольського державного педагогічного університету імені Богдана Хмельницького. Сер.: Педагогіка. – 2016. – № 1.16. – С. 178-182.
2. Kodosky, J. LabVIEW [Text] / J. Kodosky // Proceedings of the ACM on Programming Languages. – 2020. – Vol. 4, Issue HOPL. – Article no. 78. – P. 1-54.
3. Кухарець, Л. С. Візуальна мова програмування BSimple з трансляцією у Unity C# [Текст] : дис. ... магістр: захищена 06.12.21: / Кухарець Лілія Савеліївна. – Київ, 2021 – 129 с.
4. Python Software Foundation. The Python Language Reference [Електронний ресурс] / Офіційний сайт компанії Python. – Режим доступу: <https://docs.python.org/3/reference/index.html> – 21.06.2024.
5. Стахов, К. В. Розробка інтерпретатора вбудовуваної скриптової мови програмування з можливістю визначення довільної предметно-орієнтованої граматики [Текст] : дис. ... магістр: захищена 15.12.23: Стахов Кирил Віталійов. – Запоріжжя, 2023 – 111 с.

References

1. Osadcha, K. P., & Marchuk, M. S. Analiz ponyatiynoho aparatu paradyhmy vizual'noho prohrumuvannya. [Analysis of the conceptual apparatus of the visual programming paradigm]. Naukovyy visnyk Melitopol'skoho derzhavnoho pedahohichnoho universytetu imeni Bohdana Khmel'nyts'koho. Seriya: Pedahohika [Scientific Bulletin of Melitopol State Pedagogical University named after Bohdan Khmelnytskyi. Series: Pedagogy]. Melitopol, 2016, pp. 178-182.
2. Kodosky, J. LabVIEW. Proceedings of the ACM on Programming Languages, 2020, vol. 4, iss. HOPL, Article no. 78, pp. 1-54.
3. Kukharets', L. S. Vizual'na mova prohrumuvannya BSimple z translyatsiyeyu u Unity C#.

Diss. magistr. [Visual programming language BSimple with translation to Unity C#. M.Sc. diss]. Kyiv, 2021. 129 p.

4. Python Software Foundation. *The Python Language Reference*. Available at: <https://docs.python.org/3/reference/index.html> (accessed 21.06.2024).

5. Stakhov, K. V. *Rozrobka interpretatora vbudovuvanoi skryptovoyi movy prohramuvannya z*

mozlyvystyu vyznachennya dovil'noyi predmetno-orientovanoyi hramatyky. Diss. magistr. [Development of an interpreter for an embedded script programming language with the possibility of defining an arbitrary subject-oriented grammar. M.Sc. diss]. Zaporizhzhya, 2023. 111 p.

Надійшла до редакції 12.06.2024, розглянута на редколегії 15.08.2024

APPLICATION OF VISUAL PROGRAMMING FOR AUTOMATIC TESTS OF REGULATORS OF GAS TURBINE ENGINES

Serhii Vialov

Various programming methods for the development of automatic test systems for gas turbine engine regulators were considered. A detailed analysis of text, visual, and script programming methods was conducted, and the advantages and disadvantages were identified in the context of automatic testing. Text-based programming ensures high flexibility and power, allowing for the creation of optimized algorithms for complex tasks. Visual programming offers intuitiveness and convenience in development, reducing the likelihood of errors and shortening development times by using graphical elements. Script programming is characterized by speed and ease of customization, making it suitable for automating routine tasks and rapid iterations. A comparative analysis of the effectiveness of these methods in different automatic testing contexts was conducted, including an assessment of performance, reliability, scalability, and implementation ease. The experimental results confirmed the effectiveness of each method for specific tasks in automatic testing systems for gas turbine engine regulators. Specifically, we found that text-based programming is optimal for complex and critical systems that require high performance and detailed control. Script programming was the most suitable method for automating routine tasks and quickly setting up test scenarios. Visual programming is appropriate for rapid development and ease of use, especially when intuitiveness and error reduction are crucial. The conclusions provide recommendations for selecting the optimal programming method for specific tasks in automatic testing systems. For implementing automatic checks of the development units of JSC "Element," the visual programming method is advisable, considering its advantages in the simplicity of interpreting created algorithms, real-time display of current parameter values, and flexibility in editing algorithms without the need for recompilation after changes. The obtained results will enhance the accuracy, speed, and reliability of automatic testing processes for gas turbine engine regulators.

Keywords: GTE; automation; visual programming; automatic testing systems; algorithm optimization.

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

Serhii Vialov – PhD Student, Odesa Polytechnic University; Electronic Engineer, JSC "Element", Odesa, Ukraine,
e-mail: vialov0608@gmail.com, ORCID: 0009-0006-7784-8896.