

**Запропоновано концепцію адаптивного клієнта як альтернативу «тонкому» та «товстому» клієнту та його інтеграцію в автоматизовану комп'ютерну систему ідентифікації технічних станів промислових об'єктів. Розроблено теоретичний базис та варіант реалізації адаптивного клієнта на конкретному апаратному та програмному забезпеченні. Зроблено висновок щодо доцільності використання такої концепції в системах технічної діагностики**

**Ключові слова:** система діагностики, адаптивний клієнт, технічний стан, обчислювальна потужність, функціональний модуль

**Предложена концепция адаптивного клиента в качестве альтернативы «тонкому» и «толстому» клиенту и его интеграция в автоматизированную компьютерную систему идентификации технических состояний промышленных объектов. Разработан теоретический базис и вариант реализации адаптивного клиента на конкретном аппаратном и программном обеспечении. Сделаны выводы насчет целесообразности применения данной концепции в системах технической диагностики**

**Ключевые слова:** система диагностики, адаптивный клиент, техническое состояние, вычислительная мощность, функциональный модуль

УДК 004.75

# ЗАСТОСУВАННЯ КОНЦЕПЦІЇ АДАПТИВНОГО КЛІЄНТА В КОМП'ЮТЕРНІЙ СИСТЕМІ ТЕХНІЧНОЇ ДІАГНОСТИКИ

**М. І. Горбійчук**

Доктор технічних наук, професор\*

E-mail: ksm@nung.edu.ua

**М. О. Слабінога**

Аспірант\*

E-mail: mslabinoha@gmail.com

\*Кафедра комп'ютерних систем та мереж  
Івано-Франківський національний технічний  
університет нафти і газу  
вул. Карпатська 15, м. Івано-Франківськ,  
Україна, 76018

## 1. Вступ

При вирішенні задач оперативної ідентифікації технічного стану промислового об'єкта важливим параметром є швидкодія системи при виконанні тієї чи іншої задачі діагностики. Тому важливою задачею при проектуванні системи є мінімізація затрат часових ресурсів при виконанні обчислень для оперативного забезпечення оператора діагностичною інформацією.

Проектована авторами автоматизована комп'ютерна система ідентифікації технічного стану промислових об'єктів спрямована на багаторівневий глибокий аналіз даних з допомогою доступних передових методик та потребує розробки комплексного підходу до вирішення проблеми мінімізації часу виконання того чи іншого етапу аналізу. Загальна схема роботи системи зображена на рис. 1.

Слід зазначити також, що деякі з етапів аналізу вимагають залучення досить великих обчислювальних ресурсів (наприклад, вейвлет-аналіз та аналіз декомпозиції емпіричних мод), тоді як інші етапи не потребують великих ресурсів та дозволяють провести оперативну діагностику, хоча і без виявлення довготермінових трендів у зміні параметрів функціонування об'єкта (наприклад, статистичний аналіз та порівняння значень параметрів із граничними значеннями, поданими в державних стандартах). Тому слід проводити чітку градацію за пріоритетом виконання етапів, виходячи з етапів роботи.

Крім того, до системи ставляться стандартні вимоги, що диктуються сучасними тенденціями розвитку обчислювальної техніки – система має бути мобільною, компактною, програмне забезпечення системи має бути модульним та масштабованим.

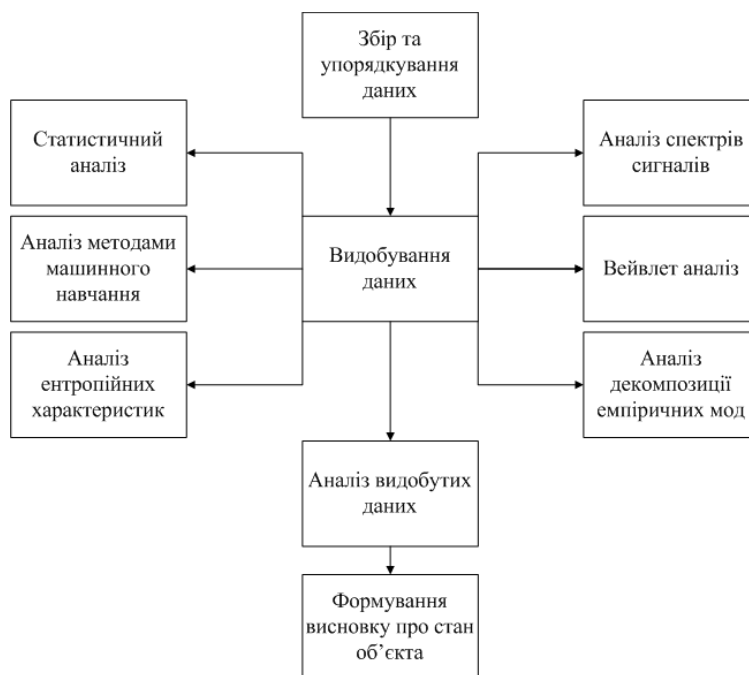


Рис. 1. Схема роботи комп'ютерної системи ідентифікації технічного стану промислових об'єктів

Тому актуальною проблемою в проектуванні клієнт-серверної архітектури системи є оптимальний розподіл навантаження на обчислювальні ресурси клієнта та сервера, який би здійснювався програмним рішенням на базі клієнта.

## 2. Аналіз існуючих рішень та постановка проблеми

Типовим рішенням для градації виконуваних завдань без шкоди мобільності та компактності системи є клієнт-серверна архітектура, що складається із сервера (серверів), що надає частину своїх ресурсів та функціоналу для загального користування, клієнта (клієнтів), що використовує ці ресурси та мережі, що забезпечує обмін даних між ними [1]. Такий підхід дозволяє знизити вимоги до клієнтських комп'ютерів та, як наслідок зробити їх компактнішими. Однак, загальним недоліком цього підходу є те, що сервер в деякі моменти часу може бути недоступним з технічних причин, та, як наслідок, порушити роботу всієї системи.

Особливо це проявляється в підході з «тонким» клієнтом [2], який дозволяє звести розміри та вимоги до клієнта до мінімуму, однак виконує тільки функцію вводу-виводу та відображення результатів, перекладаючи всю обчислювальну роботу на ресурси сервера. Крім того, клієнтів може бути декілька, що спричинює ще більше навантаження на сервер без можливості розподілити його рівномірно. Частково завдання сервера спрощуються при застосуванні концепції «товстого клієнта» [3], який звертається до сервера тільки як до оператора загальної бази даних. Однак при цьому, відповідно, компактність не є пріоритетною і побудова більш потужної машини-клієнта вимагає більших затрат.

Останнім часом розглядаються варіанти застосування даних концепцій з різними змінами в логіці роботи. Зокрема, в роботі [4] автори пропонують предиктивну модель, яка б аналізувала зміни, які стануться в системі після проведення обчислень. Слід також відзначити тенденцію до застосування програм для відображення роботи сервера на мобільних пристроях та впровадження в них елементів адаптивності [5–7]. Крім того, значна увага надається управлінню системами через веб-інтерфейси [8–10] та побудови системи з використанням хмаркових сервісів [11–13].

## 3. Мета за задачі дослідження

Метою даного дослідження була розробка концепції адаптивного клієнта, який оперативно оцінював би обсяг обчислень, який слід виконати і приймав рішення щодо самостійного виконання обчислень або передачі задачі на сервер, виходячи з навантаження на сервер та мережу, а також власних вільних обчислювальних ресурсів.

Задачею дослідження була розробка математичного та програмного апарату оцінювання для подальшої інтеграції реалізованої концепції в автоматизовану комп'ютерну систему ідентифікації технічного стану промислових об'єктів.

## 4. Вибір апаратного та програмного забезпечення для реалізації комп'ютерної системи

Основними критеріями вибору апаратного забезпечення клієнта комп'ютерної системи були:

- компактність;
- вартість;
- швидкодія;
- сумісність із сучасними стандартами вводу-виводу інформації та передачі даних.

В якості клієнта при тестуванні було вирішено використовувати одноплатний комп'ютер Raspberry Pi Model B з ARM-процесором тактової частоти 700 МГц, 512 Мб оперативної пам'яті та відео-модулем Broadcom VideoCore IV. Вартість даного комп'ютера становить \$35, що робить його однією з найдоступніших, найкомпактніших обчислювальних машин даного сегменту [14]. На платі присутні порти HDMI, USB, Composite Video, 3.5mm jack, SD, зв'язок з низькорівневою периферією організовано з допомогою 16 контактів GPIO, SPI, I<sup>2</sup>C, UART, тобто плата дозволяє організувати широкий спектр вводу-виводу. Мережеві можливості плати представлені Ethernet-портом, крім того, USB-порт дозволяє організувати безпроводні мережеві з'єднання Wi-Fi, 3G та EDGE шляхом підключення сторонніх мережевих модулів [15].

Сервером для даної системи може виступати будь-який комп'ютер, обчислювальна потужність якого достатня для виконання задач системи. Для тестування при роботі з даними технологічного процесу газоперекачувального агрегату використовувався персональний комп'ютер з процесором Intel Core i7 з 4 фізичними та 8 віртуальними ядрами з 8 Гб ОЗП.

До програмного забезпечення системи ставилися наступні вимоги:

- відкритість;
- кросплатформенність;
- модульованість;
- наявність бази репозиторіїв з готовою реалізацією окремих методів.

В якості операційної системи сервера використовується Linux Mint, операційної системи клієнта – Raspbian. Встановлення Raspbian додатково дозволяє безкоштовно використовувати бібліотеки знаннево-орієнтованої системи математичних розрахунків Wolfram Mathematica [16]. Для написання основних модулів було використано мову програмування Python. Дана мова програмування відрізняється від інших своєю гнучкістю, простотою роботи зі списками, матрицями та іншими типами масивів [17], кросплатформенністю та великим набором безплатних стандартних бібліотек для проведення наукових обчислень [18], що в більшості випадків дозволяє рівноцінно замінити математичне забезпечення таких комерційних продуктів, як Matlab та Mathcad. Для взаємодії із користувачем було вирішено використовувати веб-інтерфейс, побудований на базі відкритого сервера Django, що дозволяє, ввівши IP-адресу клієнта в адресному рядку браузера, отримувати оперативні дані аналізу на будь-якому ПК чи мобільному пристрої.

Таким чином, затрати на програмне забезпечення були мінімізовані, до того ж, все обране ПЗ доступне для роботи на широкому спектрі архітектур, що дозволяє вибрати оптимальну архітектуру клієнта та сервера в залежності від задач.

### 5. Теоретичні основи та реалізація програмного забезпечення для адаптивного клієнта

Очевидно, що в описаній вище концепції, клієнт має керуватися наступним правилом – якщо прогнозований час виконання на сервері (з урахуванням передачі даних) менший від прогнозованого часу виконання на клієнті, то слід передати запит на виконання завдання на сервер.

В першу чергу, слід вибрати, де зберігати дані і в якому обсязі. В тестованій системі при роботі з даними газоперекачувальних агрегатів в середньому mat-файл даних з одного газоперекачувального агрегату за місяць займав 200 Кб дискового простору. Це значить, що на локальному клієнті можна без проблем зберігати дані про газоперекачувальні агрегати однієї газокompресорної станції за цілий рік. Обчислення довготерміновіших трендів уже постійно буде проводитися на сервері, оскільки такі обсяги даних потребують відповідних обчислювальних ресурсів, а оперативність обчислення результатів на таких термінах є неприциповою.

Отже, міститися на базі клієнта будуть міститися наступні дані:

- програмне забезпечення для проведення обчислень;
- дані з промислового об'єкту за останній рік;
- дані про те, які з діагностичних даних містяться в базі сервера (з постійним оновленням);

Для висновку щодо прогнозованого часу виконання завдання на клієнті та сервері слід враховувати наступні фактори:

- навантаження на клієнт та сервер;
- швидкість передачі в мережі;
- середній час виконання завдання даної складності (з вказаною розмірністю даних та операціями даного типу) на клієнті та сервері.

Для формалізації навантаження на клієнт та сервер введемо наступний коефіцієнт навантаження  $I$ , який дорівнює:

$$I = (1 - I_{HDD})M_{HDD} * (1 - I_{RAM})M_{RAM} * (1 - I_{CPU})M_{CPU}, \quad (1)$$

де  $I_{HDD}$ ,  $I_{RAM}$ ,  $I_{CPU}$  – відсоток навантаження на жорсткий диск, оперативну пам'ять та процесор клієнта (сервера) відповідно, отриманий з допомогою інструменту RSUtils4Python;  $M_{HDD}$ ,  $M_{RAM}$ ,  $M_{CPU}$  – результат бенчмаркінгу (оцінки загальних характеристик машини при нульовому навантаженні)[19] для жорсткого диска, оперативної пам'яті та центрального процесора відповідно.

Порівнявши отримані значення коефіцієнта для клієнта та сервера відповідно, та вибравши більше значення, можна зробити висновок, яка з двох машин на даний момент володіє більшою обчислювальною потужністю з урахуванням інших програм, що виконуються на них. Такий підхід дозволяє також проводити адекватну оцінку можливостей сервера при роботі з декількома клієнтами.

Для обчислення швидкості передачі даних в мережі, передається контрольний пакет даних обсягом 1 кб. Виходячи з даних програмного модуля, що передає пакет, швидкість передачі обчислюється наступним чином:

$$v = I / (t - t_0), \quad (2)$$

де  $t_0$  – час початку передачі,  $t$  – час закінчення передачі,  $I$  – обсяг контрольного пакету.

Для оцінки середнього часу виконання було прийнято рішення використовувати штучну нейронну мережу з використанням алгоритмів навчання з вчителем. Вхідними значеннями мережі є:

- тип завдання (статистичний аналіз, вейвлет-перетворення, аналіз емпіричних компонент тощо), заданий кодом від 1 до 6;
- обсяг даних, з якими здійснюється робота (кількість чисел з плаваючою комою у вхідній матриці).

Цільовими значеннями для тренування було вибрано фактичний час виконання конкретних завдань з матрицями заданої величини, що розділявся на відсоток вільних ресурсів на момент початку виконання завдання.

Тренована нейронна мережа в результаті дає можливість прогнозувати приблизний час, причому, у зв'язку з неповнотою даних, даний підхід є точнішим ніж застосування статистичних алгоритмів регресії [20].

Загальний алгоритм проведення аналізу даних на стороні клієнта відбувається за блок-схемою, зображеною на рис. 2.

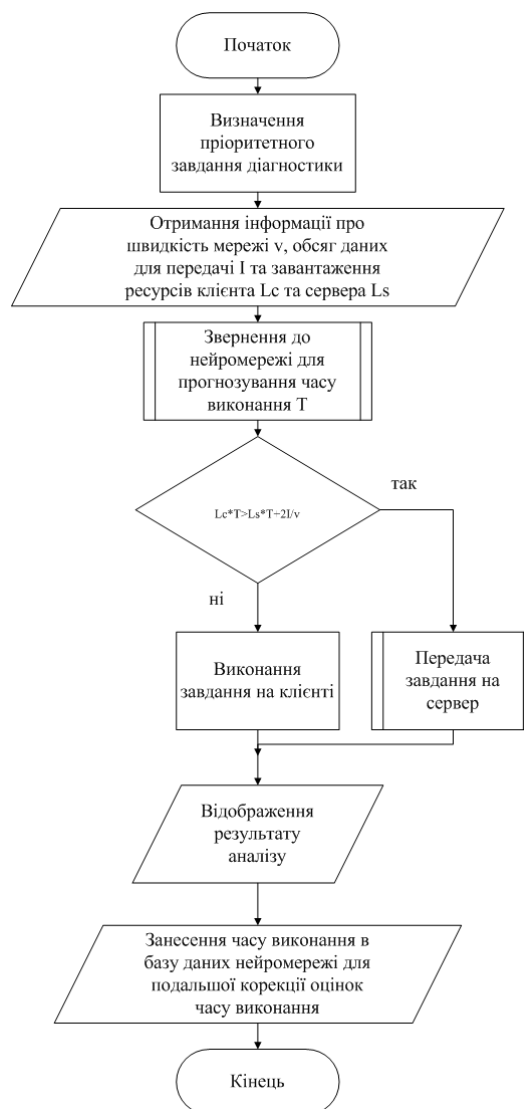


Рис. 2. Алгоритм проведення аналізу даних на стороні клієнта

Очевидно, що при проведенні аналізу даних, який потребує проведення невеликої кількості обчислень, дана концепція буде працювати дещо повільніше. Однак, якщо брати тривалі часові проміжки з великим обсягом аналізу даних, оптимізація розподілу обчислювальних завдань між клієнтом і сервером дасть можливість зменшити час діагностики в порівнянні зі схемами з «тонким» та «товстим» клієнтом.

### 6. Результати дослідження

Для перевірки результатів було проведено аналіз даних газоперекачувального агрегату за 3 місяці. Спочатку брався аналіз з найменшим обчислювальним навантаженням (статистичний). З кожним кроком додався новий вид аналізу з більшим навантаженням на обчислювальні ресурси. Графік середнього часу аналізу на трьох видах клієнтів показаний на рис. 3.

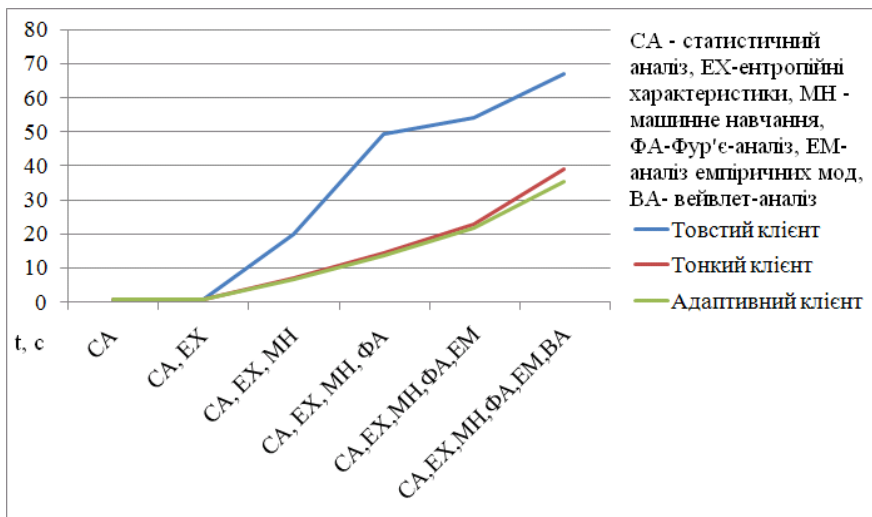


Рис. 3. Результати роботи трьох видів клієнтів для різних видів аналізу

На початкових етапах адаптивний клієнт поступається в швидкодії товстому клієнту за рахунок витрати обчислювальних потужностей на прогнозування часу виконання. Однак при включенні в аналіз методів, що

вимагали великих обчислювальних потужностей, товстий клієнт дав найгірший результат, оскільки швидкодія сервера є в рази кращою. При цьому, адаптивний клієнт мав незначну перевагу за рахунок виконання статистичного та ентропійного аналізу на власних потужностях, без передачі даних на сервер, економлячи час на передачу.

### 7. Висновки

Запропонована концепція адаптивного клієнта на базі одноплатного комп'ютера Raspberry Pi Model B використовує розроблений та реалізований алгоритм, що враховує прогнозований нейронною мережею обсяг обчислень, що мають бути виконані, а також поточний рівень завантаження обчислювальних ресурсів клієнта, сервера та мережі з метою визначення доцільності виконання обчислювального завдання на клієнті або

передачі його на сервер. Перевага такого підходу перед «тонким» та «товстим» клієнтом полягає в тому, що у випадку невеликих завдань клієнт не витрачає час на передачу їх на сервер (як у варіанті «тонкого» клієнта), а у випадку завдань, що вимагають значного обсягу обчислень, клієнт не виконує їх самостійно (як у варіанті «товстого» клієнта). Таким чином, адаптивний клієнт поєднує в собі сильні сторони обох стандартних реалізацій клієнт-серверної архітектури. Крім того, клієнт-серверна архітектура системи дозволяє реалізувати підхід з використанням Web-інтерфейсу та хмаркових обчислень, що відповідає останнім тенденціям у взаємодії користувача та машин з великою обчислювальною потужністю. Функціональний модуль з реалізацією концепції адаптивного клієнта інтегровано в розроблювану авторами автоматизовану комп'ютерну систему ідентифікації технічних станів промислових об'єктів.

### Література

1. Tanenbaum, A. Computer Networks [Text] / A. Tanenbaum. – New Jersey: Prentice Hall, 2010. – 960 с.
2. Kanter, J. Understanding Thin Client/Server Computing [Text] / J. Kanter. – Redmond: Microsoft press, 1997. – 256 с.
3. Boudreau, T. Rich Client Programming: Plugging into the NetBeans Platform [Text] / T. Boudreau. – New Jersey: Prentice Hall, 2007. – 640 с.
4. Sallam, A. Proactive workload management in dynamic virtualized environments [Text] / A. Sallam, K. Li, A. Ouyang, Z. Li // Journal of Computer and Systems Sciences. – 2014. – Vol. 80, Issue 8. – P. 1504–1517. doi:10.1016/j.jcss.2014.04.018
5. Mizouni, R. A framework for context-aware self-adaptive mobile application SPL [Text] / R. Mizouni, M. A. Matar, Z. A. Mahmoud, S. Alzahmi, A. Salah // Expert Systems with applications. – 2014. – Vol. 41, Issue 16. – P. 7549–7584. doi:10.1016/j.eswa.2014.05.049
6. Acker, A. The Short Message service: Standarts, Infrastructure and Innovations [Text] / A. Acker // Telematics and informatics. – 2014. – Vol. 31, Issue 4. – P. 559–568. doi:10.1016/j.tele.2014.01.004
7. Zhao, Y. An Android application for receiving notifications of astrophysical transient events [Text] / Y. Zhao, I. A. Bond, W. L. Sweatman // Astronomy and computing. – 2014. – Vol. 6. – P. 9–27. doi:10.1016/j.ascom.2014.05.001



8. Oriol, M. Quality models for web services: A systematic mapping [Text] / M. Oriol, J. Marco, X. Franch // Information and software technology. – 2014. – Vol. 56, Issue 10. – P. 1167–1182.
9. Sheng, Q. Z. Web-services composition: A decade's overview [Text] / Q. Z. Sheng, X. Qiao, A. V. Vasilakos, C. Szabo, S. Bourne, X. Xu // Information Sciences. – 2014. – Vol. 280. – P. 218–238. doi:10.1016/j.ins.2014.04.054
10. Silva, S. Development of a web-based multi-criteria spatial decision support system for the assessment of environmental sustainability of dairy farms [Text] / S. Silva, L. Al ada-Almeida, L. C. Dias // Computers and Electronics in Agriculture. – 2014. – Vol. 108. – P. 46–57. doi:10.1016/j.compag.2014.06.009
11. Yigit, M. Cloud Computing for Smart Grid applications [Text] / M. Yigit, V. C. Gungor, S. Baktir // Computer Networks. – 2014. – Vol. 70. – P. 312–329. doi:10.1016/j.comnet.2014.06.007
12. Li, Zh. A generic cloud platform for engineering optimization based on OpenStack [Text] / Z. Li, H. Li, X. Wang, K. Li // Advances in Engineering Software. – 2014. – Vol. 75. – P. 42–57. doi:10.1016/j.advengsoft.2014.05.001
13. Yu, H. Granary: A sharing oriented distributed storage system [Text] / H. Yu, F. Zhang, Y. Wu // Future Generation Computer Systems. – 2014. – Vol. 38. – P. 47–60. doi:10.1016/j.future.2013.08.001
14. Upton, E. Raspberry Pi User Guide [Text] / E. Upton. – New York City: John Wiley & Sons, 2012. – 262 p.
15. Corbet, J. Linux Device Drivers [Text] / J. Corbet. – Sebastopol: O'Reilly Media, Inc., 2005. – 640 p.
16. Wolfram, S. The Mathematica Book [Text] / S. Wolfram. – Champaign: Wolfram Media, 2004. – 1488 p.
17. McKinney, W. Python for Data Analysis [Text] / W. McKinney. – Sebastopol: O'Reilly Media, Inc., 2012. – 470 p.
18. Downey, A. Think Python [Text] / A. Downey. – Sebastopol: O'Reilly Media, Inc., 2012. – 300 p.
19. Jarvis, S. High Performance Computing Systems Performance Modeling, Benchmarking and Simulation [Text] / S. Jarvis. – New York City: Springer, 2014. – 240 p.
20. Russel, S. Artificial Intelligence: A Modern Approach [Text] / S. Russell. – New Jersey: Prentice Hall, 2009. – 1152 p.

*Запропоновано перешкодостійкий алгоритм розпізнавання символної інформації без процедур попередньої сегментації та контурної фільтрації, побудований на основі комбінації кореляційного методу та критерію мінімуму кодових відстаней. Проведено порівняльний аналіз з відомими алгоритмами розпізнавання символної інформації. Показано, що запропонований алгоритм вимагає виконання меншого числа арифметичних операцій та має низьку складність технічної реалізації*

*Ключові слова: розпізнавання символної інформації, ковзна кореляція, бінаризація зображення, кодова відстань Хемінга*

*Предложен помехоустойчивый алгоритм распознавания символной информации без процедур предварительной сегментации и контурной фильтрации, построенный на основе комбинации корреляционного метода и критерия минимума кодовых расстояний. Проведен сравнительный анализ с известными алгоритмами распознавания символной информации. Показано, что предложенный алгоритм требует выполнения меньшего числа арифметических операций и обладает низкой сложностью технической реализации*

*Ключевые слова: распознавание символной информации, скользящая корреляция, бинаризация изображения, кодовое расстояние Хэмминга*

УДК 004.932.1

## ПОВЫШЕНИЕ ПОМЕХОУСТОЙЧИВОСТИ «СКОЛЬЗЯЩЕГО» КОРРЕЛЯЦИОННОГО АЛГОРИТМА РАСПОЗНАВАНИЯ ПЕЧАТНЫХ СИМВОЛОВ

О. А. Кушниренко

Аспирант\*

E-mail: kuuk@mail.ru

А. В. Садченко

Кандидат технических наук, доцент\*

E-mail: anjand@mail.ru

А. В. Троянский

Кандидат технических наук, доцент\*

E-mail: troy\_shur@mail.ru

\*Кафедра радиотехнических систем

Одесский национальный

политехнический университет

пр. Шевченко, 1, г. Одесса, Украина, 65044

### 1. Введение

В современных системах машинного зрения актуальна задача распознавания символной информации.

Решение такой задачи необходимо при распознавании регистрационных номеров автомобилей, надписей на железнодорожных вагонах и контейнерах, при идентификации символной информации на печатных платах