

automation becomes not merely a convenience but a necessity for maintaining quality at pace with development. Future work may further explore the integration of lightweight, on-device inference models that perform self-diagnostics directly within the embedded hardware, eliminating the dependency on external evaluation infrastructure. Ultimately, the convergence of machine learning, computer vision, and continuous integration practices establishes a new paradigm for robust, scalable, and intelligent quality assurance in embedded systems engineering.

### References

1. J. M. Zhang, M. Harman, L. Ma, and Y. Liu, "Machine Learning Testing: Survey, Landscapes and Horizons," *IEEE Trans. Softw. Eng.*, vol. 48, no. 1, pp. 1–36, Jan. 2022, doi: 10.1109/TSE.2019.2962027.
2. V. H. S. Durelli et al., "Machine Learning Applied to Software Testing: A Systematic Mapping Study," *IEEE Trans. Rel.*, vol. 68, no. 3, pp. 1189–1212, Sep. 2019, doi: 10.1109/TR.2019.2892517.
3. S. Segura, G. Fraser, A. B. Sanchez, and A. Ruiz-Cortés, "A Survey on Metamorphic Testing," *IEEE Trans. Softw. Eng.*, vol. 42, no. 9, pp. 805–824, Sep. 2016, doi: 10.1109/TSE.2016.2532875.
4. S. Ajourloo, A. Jamarani, M. Kashfi, M. H. Kashani, and A. Najafizadeh, "A systematic review of machine learning methods in software testing," *Appl. Soft Comput.*, vol. 162, p. 111805, 2024, doi: 10.1016/j.asoc.2024.111805.
5. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.
6. V. Garousi, N. Joy, Z. Jafarov, A. B. Keleş, S. Değirmenci, E. Özdemir, and R. Zarringhalami, "AI-powered software testing tools: A systematic review and empirical assessment of their features and limitations," *arXiv preprint arXiv:2409.00411*, 2024.

---

## РОЛЬ АУДІО-ВІЗУАЛЬНОГО ПОЄДНАННЯ ДЛЯ НАВІГАЦІЇ РОБОТІВ У ДИНАМІЧНОМУ СЕРЕДОВИЩІ

Лозицький А., Главчев Д.

Національний технічний університет «Харківський політехнічний інститут»,  
Харків, Україна

Рухомі об'єкти, такі як люди або інші роботи, створюють значні перешкоди для візуальної одометрії, оскільки вони перекривають статичний фон і призводять до збоїв у побудові карт [1]. Традиційні системи розпізнавання об'єктів на основі глибокого навчання вимагають великих обчислювальних ресурсів і потужних графічних процесорів (GPU), що обмежує їх використання на мобільних роботах [2]. Вирішенням стає застосування локалізації джерела звуку (SSL) для виявлення шумових перешкод [1]. Напрямок звуку проєктується на RGB-D зображення камери, формуючи обмежувальну зону навколо об'єкта, після чого значення глибини в цій зоні анулюються [1]. Це дозволяє системі витягувати візуальні ознаки

виключно зі статичного фону, забезпечуючи стабільну роботу SLAM у реальному часі на звичайному процесорі (CPU) [1, 2].

**Метою доповіді** є аналіз ролі аудіо-візуального поєднання для навігації роботів у динамічному середовищі, та дослідження ефективності застосування даного підходу. У ситуаціях, коли візуальна інформація недоступна або обмежена (наприклад, через раптове задимлення або стіни), покладання виключно на зір призводить до неефективного блукання простором [3]. У таких випадках звук слугує ефективним просторовим вказівником [4]. Завдяки аудіо-візуальному поєднанню акустичний контекст використовується для цілеспрямованого виділення тих ділянок візуального зображення, які безпосередньо пов'язані із джерелом звуку [4]. Такий підхід реалізує принцип, за яким звук вирішує, «куди дивитися», а зір уточнює, «як дивитися» [3]. Це значно зменшує час на пошук цілі, запобігає зіткненням та позбавляє систему необхідності покладатися на хибне запам'ятовування середовища [3, 4].

**В доповіді** аналізується питання навігації до рухомого джерела звуку, що є значно складнішою задачею, ніж до статичного, оскільки попередні спостереження швидко втрачають актуальність, вимагаючи від роботи постійно оновлювати свою пам'ять і агрегувати інформацію в часі [5]. Замість того, щоб просто рухатися за градієнтом гучності звуку, сучасні алгоритми дозволяють роботу проактивно аналізувати траєкторію рухомої цілі [5]. Завдяки безпосередньому об'єднанню геометричної карти перешкод (отриманої з датчиків глибини) із динамічними звуковими сигналами, робот може розрахувати найкоротший шлях до найближчої точки перетину з ціллю [6]. Таке просторове аудіо-візуальне злиття радикально оптимізує ефективність руху та дозволяє ефективно перехоплювати рухомі об'єкти в невідомому середовищі [6].

**Як висновок**, можна сказати, що інтеграція аудіальних та візуальних даних є критично важливим кроком для створення надійних автономних систем, здатних функціонувати поза межами ідеальних лабораторних умов. Практичне застосування таких підходів є особливо доречним у складних сценаріях: під час пошуково-рятувальних місій роботи зможуть локалізувати постраждалих за їхнім голосом у задимлених приміщеннях; у сфері безпеки – миттєво реагувати на підозрілі шуми поза зоною прямої видимості; а в логістичній та домашній робототехніці – безпечно маневрувати серед людей. Враховуючи це, подальші дослідження мають зосередитися на оптимізації алгоритмів аудіо-візуального злиття для edge-пристроїв. Це дозволить впроваджувати подібні рішення на компактних мобільних платформах із суттєво обмеженими обчислювальними ресурсами, роблячи їх масовими та енергоефективними.

### References

1. Zhang, T., Zhang, H., Li, X.: Vision-audio fusion SLAM in dynamic environments. *CAAI Trans. Intell. Technol.* 8(4), 1364–1373 (2023). <https://doi.org/10.1049/cit2.12206>.
2. M. Sewtz, "Multi-Sensor and Multi-Modal Localization in Indoor Environments on Robotic Platforms," Ph.D. dissertation, Karlsruhe Institut für Technologie (KIT), 2025.
3. Y. Wang, Y. Yu, F. Sun, L. Wang, and W. Zheng, "Audio-Guided Visual Perception for Audio-Visual Navigation," arXiv preprint arXiv:2510.11760, 2025.

4. Y. Yu, W. Huang, F. Sun, C. Chen, Y. Wang, and X. Liu, "Sound Adversarial Audio-Visual Navigation," arXiv preprint arXiv:2202.10910, 2022.
5. A. Younes, D. Honerkamp, T. Welschehold, and A. Valada, "Catch Me If You Hear Me: Audio-Visual Navigation in Complex Unmapped Environments with Moving Sounds," IEEE Robot. Autom. Lett., vol. 8, no. 2, pp. 928–935, 2023.
6. C. Huang, O. Mees, A. Zeng, and W. Burgard, "Audio Visual Language Maps for Robot Navigation," in Proc. Int. Symp. Experimental Robotics (ISER), Cham: Springer Nature Switzerland, 2023, pp. 105–117.

---

## РОЗГОРТАННЯ ТА НАЛАШТУВАННЯ JENKINS ЯК СИСТЕМИ БЕЗПЕРЕРВНОЇ ІНТЕГРАЦІЇ

Мороз А.В., Дригач К.В.

Харківський національний університет радіоелектроніки, Харків, Україна

У сучасному процесі розробки програмного забезпечення особливого значення набуває автоматизація етапів складання, тестування та розгортання додатків. Безперервна інтеграція (Continuous Integration, CI) є однією з ключових практик DevOps, яка дозволяє розробникам регулярно інтегрувати зміни в спільній репозиторій із подальшою автоматичною перевіркою їх працездатності.

Одним із найпоширеніших інструментів для реалізації CI є Jenkins – система з відкритим кодом, що забезпечує гнучке налаштування процесів автоматизації.

Jenkins являє собою сервер автоматизації, який дозволяє створювати та виконувати конвеєри (pipelines) для обробки коду. Основною перевагою системи є її модульна архітектура, що базується на використанні плагінів. Це дозволяє інтегрувати Jenkins з різними системами контролю версій, такими як Git, інструментами тестування, системами управління контейнерами та хмарними платформами [1].

Розгортання Jenkins може здійснюватися різними способами: на фізичному сервері, у віртуальному середовищі або у контейнері за допомогою Docker. Використання контейнеризації дозволяє спростити процес встановлення, забезпечити ізоляцію середовища та підвищити гнучкість масштабування системи. У середовищах Kubernetes Jenkins може бути розгорнутий як окремий сервіс із можливістю динамічного створення агентів для виконання завдань.

Процес налаштування Jenkins включає створення пайплайнів, які описують послідовність дій для обробки коду. Пайплайни можуть бути визначені у вигляді скриптів (Jenkinsfile), що дозволяє зберігати конфігурацію разом із кодом та забезпечує контроль версій процесів CI/CD. Це сприяє підвищенню прозорості та відтворюваності процесів розробки [2].

Однією з ключових функцій Jenkins є автоматизація тестування. Після кожного коміту система може запускати юніт-тести, інтеграційні тести та інші