DOI 10.70286/ISU-17.12.2025.002

РОЗРОБКА АРХІТЕКТУРИ СИСТЕМИ АВТОНОМНОЇ НАВІГАЦІЇ РОБОТОТЕХНІЧНОЇ ПЛАТФОРМИ

Скідан Владислава Валентинівна
кандидат технічних наук, доцент
Сукало Максим Леонідович
PhD, ст. викладач
Пилипенко Владислава Ігорович
ст. викладач
Радецький Владислав Сергійович
здобувач вищої освіти
Кафедра інформаційних та комп'ютерних технологій
Київський національний університет технологій та дизайну, Україна

Стрімкий розвиток робототехніки та інтелектуальних систем зумовлює зростання вимог до автономності робототехнічних платформ [1], зокрема до їх здатності самостійно орієнтуватися у просторі, планувати маршрути та безпечно взаємодіяти з навколишнім середовищем. Автономна навігація [2]є однією з ключових функціональних складових сучасних мобільних роботів [3].

Розробка ефективної архітектури системи автономної навігації є складним міждисциплінарним завданням, що поєднує методи інженерії програмного забезпечення, штучного інтелекту, обробки сигналів, комп'ютерного зору та керування. Архітектура такої системи повинна забезпечувати інтеграцію сенсорних даних (лідарів, камер, ультразвукових та інерціальних датчиків), алгоритмів локалізації, побудови карти середовища, планування траєкторій і прийняття рішень у реальному часі.

Актуальність теми зумовлена необхідністю створення гнучких, масштабованих і надійних програмно-апаратних рішень, здатних адаптуватися до динамічних умов середовища та різних сценаріїв експлуатації робототехнічних платформ. Правильно спроєктована архітектура навігаційної системи дозволяє зменшити складність розробки, підвищити надійність роботи робота, спростити модифікацію та розширення функціональності в майбутньому.

В ході роботи розроблено архітектуру системи автономної навігації, що базується на методах підкріплювального навчання (Reinforcement Learning, RL). Дана схема відображає повний життєвий цикл створення, навчання та практичного застосування навігаційної моделі – від етапу моделювання в симуляційному середовищі до інтеграції та використання на реальному робототехнічному пристрої.

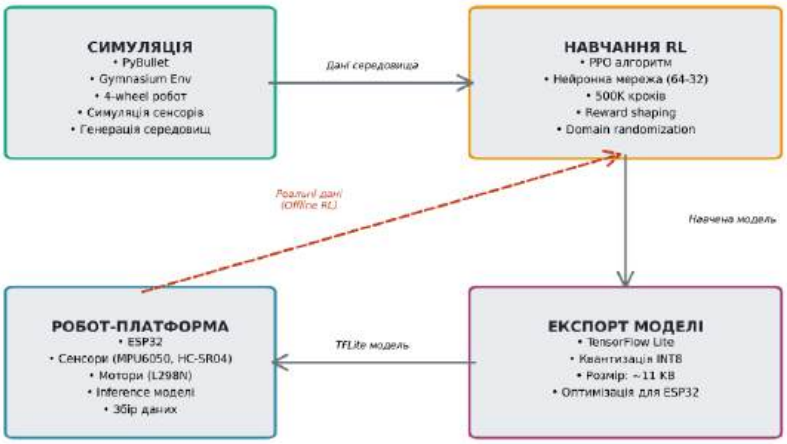

Рисунок 1 – Загальна архітектура системи автономної навігації

У лівій верхній частині подано блок «Симуляція» (рис.2), який відповідає за створення віртуального середовища навчання. Для цього використовується фізичний рушій PyBullet [4] та середовище Gymnasium Env [5], у межах яких моделюється чотириколісний робот, його сенсори та різні варіанти навколишнього середовища. Саме з цього блоку генеруються дані середовища, які передаються до модуля навчання.

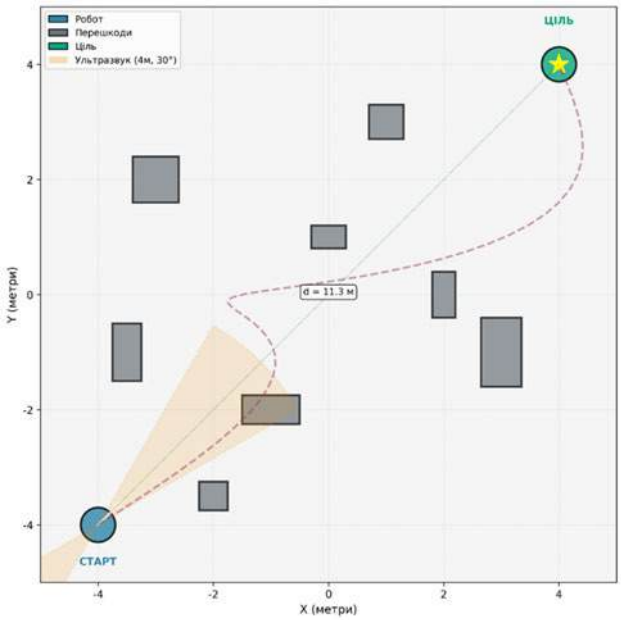

Рисунок 2. Приклад розробленого симуляційного середовища

Праворуч угорі розміщено блок «Навчання RL», у якому реалізовано процес підкріплювального навчання з використанням алгоритму PPO (Proximal Policy Optimization) [6]. Навчання здійснюється на основі нейронної мережі з архітектурою 64–32, із загальною кількістю близько 500 тисяч кроків. Для покращення збіжності застосовуються методи reward shaping та domain randomization, що підвищує стійкість моделі до змін умов середовища. На виході цього блоку формується навчена модель.

У нижній правій частині зображено блок «Експорт моделі», де навчена модель конвертується у формат TensorFlow Lite. На цьому етапі застосовується

INT8-квантизація, що дозволяє зменшити розмір моделі приблизно до 11 КБ та оптимізувати її для виконання на мікроконтролері ESP32 [7].

У нижній лівій частині представлено блок «Робот-платформа», який описує апаратну реалізацію системи. Платформа побудована на мікроконтролері ESP32 (рис.3) та оснащена сенсорами MPU6050 і HC-SR04 [8-9], а також моторним драйвером L298N. На цьому рівні відбувається інференс моделі, збір реальних даних і керування рухом робота. Навчена TFLite-модель передається на платформу для виконання в реальному часі.

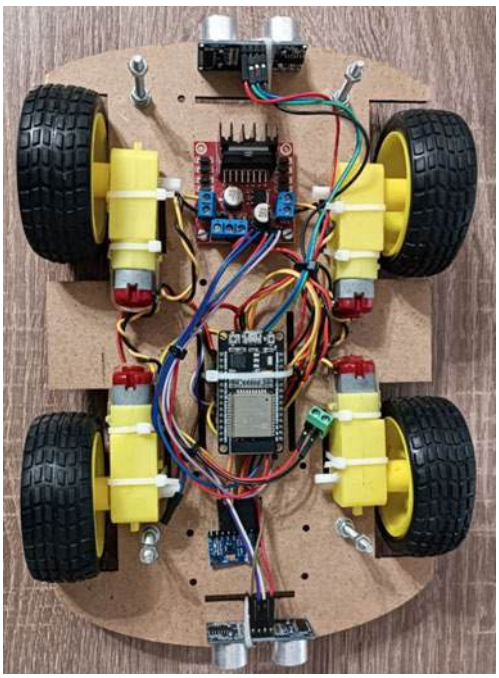

Рисунок 3 – Загальний вигляд платформи

Окремо пунктирною стрілкою показано можливість використання реальних даних (Offline RL), які надходять від робот-платформи назад у модуль навчання. Це дозволяє вдосконалювати модель на основі даних з реального середовища, підвищуючи точність і надійність автономної навігації.

Результати роботи можуть бути використані як основа для подальшої реалізації програмного забезпечення автономної навігації та впровадження його в реальні робототехнічні системи різного призначення.

Список використаних джерел

1. Correll N., Introduction to Autonomous Robots: Mechanisms, Sensors, Actuators, and Algorithms / N. Correll, B. Hayes, C. Heckman, A. Roncone. – 1st Edition : MIT Press, 2022. – 288 p.
2. G. Lozenguez, L. Adouane, A. Beynier, A. I. Mouaddib, P. Martinet, Punctual versus continuous auction coordination for multi-robot and multi-task topological navigation, Autonomous Robots – Springer 40 (4) (2016) 599–613.
3. J.-A. Fernández-Madrigal, J. L. B. Claraco, Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods, IGI Global, Hershey, PA, USA, 2013.

4. Посібник користувача Pybullet Python modul Real-Time Physics Simulation [Електронний ресурс]. – Режим доступу: https://manuals.plus/uk/pybullet/pybullet-python-module-real-time-physics-simulation-user-manual
5. Env - Gymnasium Documentation.[Електронний ресурс]. – Режим доступу: https://gymnasium.farama.org/api/env/
6. PPO(Proximal Policy Optimization). [Електронний ресурс]. – Режим доступу: https://aiengineering.academy/LLM/TheoryBehindFinetuning/PPO/
7. ESP32 / ESP8266. [Електронний ресурс]. – Режим доступу: https://docs.arduino.cc/arduino-cloud/guides/esp32/
8. Staff L. E. How HC-SR04 ultrasonic sensor works & how to interface it with Arduino [Електронний ресурс] / Last Minute Engineers. – Режим доступу: https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/
9. Ультразвуковий датчик відстані HC-SR04 [Електронний ресурс]. – Arduino в Україні. – Режим доступу: https://arduino.ua/prod182-yltrazvykovoi-datchik-rasstoyaniya-hc-sr04.

КВАНТУВАННЯ MLP ДЛЯ АПАРАТНОЇ РЕАЛІЗАЦІЇ ЗАДАЧ ВИЯВЛЕННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Зубрицький Олексій
аспірант
Донченко Євгеній
кандидат. техн. наук., старший викладач
Кафедра автоматизації виробничих процесів
Донбаська державна машинобудівна академія, Україна

У 2020–2025 роках проблема виявлення шкідливого програмного забезпечення залишається надзвичайно актуальною. За даними антивірусних компаній [1], середньодобова кількість виявлених шкідливих файлів у світі зросла з приблизно 340 тис. у 2019 році до понад 410 тис. у 2023 році. Водночас класичні антивірусні методи, зокрема сигнатурний аналіз та евристичні правила, стають недостатньо ефективними для виявлення нових варіантів шкідливого програмного забезпечення [2]. Зловмисники активно застосовують техніки обфускації та поліморфізму, що суттєво ускладнює детектування невідомих загроз традиційними засобами захисту [3]. Штучні нейронні мережі здатні ефективно виявляти складні нелінійні залежності у великих обсягах даних та автоматично формувати релевантні ознаки для задач класифікації malware. Однак застосування нейронних мереж супроводжується значними обчислювальними витратами та високими вимогами до апаратних ресурсів.

FPGA-платформи забезпечують високий рівень паралелізму обчислень і використовують спеціалізовані апаратні блоки (LUT, DSP), що дозволяє