

## **Машинне навчання в задачі автокалібрування рухомих елементів робототехнічних систем на прикладі керування кроковими двигунами**

*В роботі розглянуто використання машинного навчання для автокалібрування рухомих елементів робототехнічних систем, зокрема, для керування двома кроковими двигунами, оскільки вони є популярними у робототехніці завдяки їхній точності і контролю за кутом обертання. Описано алгоритм автокалібрування, який використовує машинне навчання для покращення точності руху крокових двигунів. Представлено приклад, де порівнювалася точність руху крокових двигунів, які були налаштовані з використанням автокалібрування на основі машинного навчання. Запропонований підхід показує, що використання машинного навчання для автокалібрування рухомих елементів робототехнічних систем, зокрема крокових двигунів, може покращити їхню точність і допомогти забезпечити більш ефективну роботу робототехнічної системи.*

### **Вступ**

Автокалібрування рухомих елементів роботів є однією з важливих задач у робототехніці, яка полягає у налаштуванні параметрів рухомих елементів робота для досягнення максимальної точності та ефективності роботи[1].

Калібрування рухомих елементів роботів є ключовим етапом в їх розробці та експлуатації, оскільки несвоєчасне чи неналежне калібрування може призвести до збоїв у роботі, зменшення точності та швидкості його руху, а також збільшення ризику виникнення аварійних ситуацій[3]. Однак, традиційні методи калібрування рухомих елементів роботів, такі як використання датчиків чи програмування параметрів руху вручну, можуть бути складними у використанні.

Оскільки, застосування методів машинного навчання стає все більш популярним у робототехніці, для вирішення цього завдання доцільно використати машинне навчання, оскільки воно є однією з ключових технологій, яка дозволяє роботам і автоматизованим системам взаємодіяти з навколишнім середовищем та здійснювати складні функції з максимальною точністю та ефективністю[2]. Наприклад, методи глибинного навчання можуть використовуватися для автоматичного визначення оптимальних параметрів руху робота, виходячи з даних, зібраних під час його роботи, а також для прогнозування можливих відхилень у його роботі. В результаті це дозволить автоматизувати та спростити процес калібрування рухомих елементів роботів, забезпечуючи максимальну точність та швидкість роботи[4].

Тому предметом дослідження є застосування методів машинного навчання, що дозволять ефективно вирішувати задачі калібрування рухомих елементів робототехнічних систем.

### Постановка проблеми

Традиційно калібрування виконується вручну фахівцями, що вимагає великих зусиль та часу. Однак, з розвитком технологій машинного навчання з'явилися можливості автоматизувати цей процес та знизити витрати на нього[5].

Застосування методів машинного навчання для калібрування рухомих елементів роботів має декілька переваг. Вони можуть працювати з великими обсягами даних, що дозволяє отримати більш точні результати та зменшити кількість помилок. Крім того, використання методів машинного навчання може дозволити роботам адаптуватися до змін у середовищі та забезпечити максимальну ефективність.

Одним з можливих методів машинного навчання для калібрування рухомих елементів роботів є Q-Learning. Цей метод є одним з видів підсиленого навчання, де робот взаємодіє з навколишнім середовищем та навчається приймати рішення на основі отриманих винагород. У контексті калібрування рухомих елементів, Q-Learning може використовуватися для побудови моделі, яка визначає оптимальні параметри руху робота на основі даних про його роботу.

Для застосування Q-Learning необхідно створити модель, яка буде відслідковувати стан робота та його дії, а також, яка буде визначати винагороди за кожну дію (див.рис.1). В даному випадку, дії включають зміну параметрів руху робота, а винагороди відображають точність його руху.

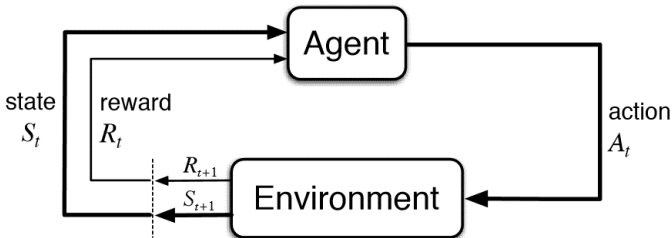


Рис.1. Навчання з підкріпленням, агент і середовище.

Оптимальні параметри руху на основі даних, які він отримує під час своєї роботи. Це можна зробити шляхом ітеративного вибору дій та оновлення моделі на основі отриманих винагород.

На початку процесу навчання модель може бути заповнена випадковими значеннями параметрів руху робота. Робот буде рухатися з цими параметрами та отримувати винагороди за свою роботу. Далі, на основі отриманих даних, модель буде оновлюватися та вибирати нові параметри руху. Цикл навчання буде продовжуватися до досягнення оптимальних параметрів руху, які максимізують винагороди за роботу робота.

Однією з переваг Q-Learning є те, що він може навчити робота адаптуватися до змін у середовищі, так як він використовує винагороди, отримані за роботу в різних умовах. Це дозволяє роботу продовжувати працювати ефективно навіть при зміні параметрів середовища.

Алгоритм нечіткого Q-навчання:

- Спостереження за станом  $x$
- Для кожного запусненого правила вибирається одна дія відповідно до стратегії використання.
- Відбувається розрахунок  $a(x)$

$$a(\mathbf{x}) = \frac{\sum_{i=1}^H w_i(\mathbf{x}) a_i}{\sum_{i=1}^H w_i(\mathbf{x})}$$

- Відбувається розрахунок відповідного значення  $Q(x, a)$

$$Q(\mathbf{x}, a) = \frac{\sum_{i=1}^H w_i(\mathbf{x}) a_i q[\mathbf{x}, i^\dagger]}{\sum_{i=1}^H w_i(\mathbf{x})}$$

Де  $q[x, i]$  є відповідним значенням  $q$  спрацьованого правила  $i$  для вибору дії  $i$  за алгоритмом. Застосування  $a(x)$  і спостереження за новим правилом  $x'$

- Розрахунок винагороди
- Оновлення значень  $q$  відповідно до:

$$\Delta q[\mathbf{x}, i^\dagger] = \eta \Delta Q \frac{w_i(\mathbf{x})}{\sum_{i=1}^H w_i(\mathbf{x})}$$

$$\Delta Q = R(\mathbf{x}, a, \mathbf{x}') + \gamma \cdot Q(\mathbf{x}', a^*) - Q(\mathbf{x}, a), \quad Q(\mathbf{x}', a^*) = \frac{\sum_{i=1}^N w_i(\mathbf{x}') a_i q[i, i^*]}{\sum_{i=1}^N w_i(\mathbf{x}')}$$
 and  $q[i, i^*]$

Загалом, використання методів машинного навчання для калібрування рухомих елементів роботів є перспективним напрямом розвитку робототехніки. Використання Q-Learning може допомогти забезпечити максимальну точність та ефективність роботів у різних галузях діяльності.

### Вирішення проблеми

Об'єктом дослідження є вертикальний плотер (див.рис.2), що складається з двох крокових двигунів Nema17, зовнішньої камери та користувацького інтерфейсу, який був розроблений на мові програмування Python. Крокові двигуни відповідають за переміщення рухомих елементів плотера, зовнішня камера служить для спостереження за процесом роботи та збору даних для подальшого аналізу. Користувацький інтерфейс забезпечує можливість взаємодії з плотером та контролює його роботу.

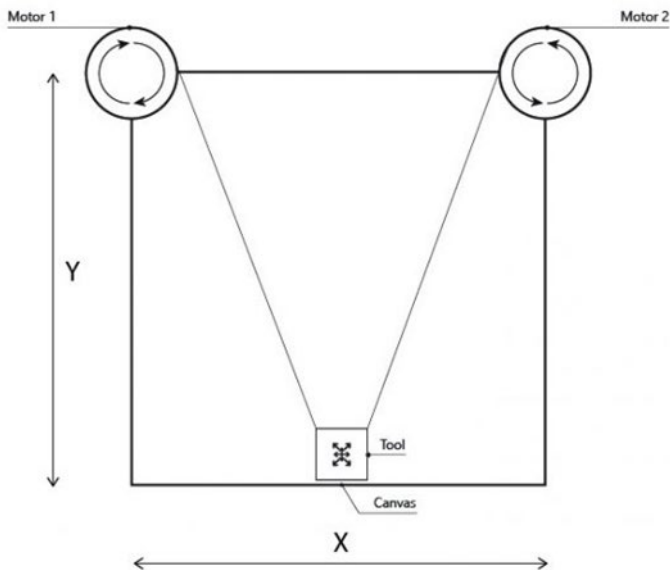


Рис. 2. Схематична конструкція стенду для дослідження

Для контролю роботи системи використовується алгоритм пошуку об'єкту рухомого елемента, який складається з наступних етапів:

1. Відокремлення зображення рухомого елемента від фону. Цей етап зазвичай використовує алгоритми віднімання фону або порівняння зразків для виявлення об'єкту на зображенні.

2. Визначення кольору рухомого елемента. Цей етап може використовувати різні моделі кольору, такі як RGB, HSV або HSL, для визначення кольорних характеристик рухомого елемента.

3. Визначення координат рухомого елемента на зображенні та передача цих даних до системи навчання. Цей етап включає в себе визначення положення рухомого елемента на зображенні та передачу цих даних до системи навчання для подальшого аналізу та обробки.

Після визначення координат рухомого елемента на зображенні та передачі цих даних до системи навчання, алгоритм використовує Q-Learning, для визначення необхідних управляючих сигналів по переміщення рухомого елемента на потрібну позицію.

Під час роботи системи двигуни забезпечують рух в чотирьох можливих напрямках: за годинниковою та проти годинникової стрілок для кожного з двигунів.

Робот отримує нагороду або покарання залежно від правильності напрямку руху до цілі. Напрямок до цілі надає позитивну нагороду, а напрямок від цілі - негативну. Крім того, позитивна нагорода надається, якщо робот досяг меж робочої зони та поставив об'єкт керування в ціль, а негативна, якщо робот не дістався до цілі або не поставив об'єкт керування в ціль (див. рис. 3).

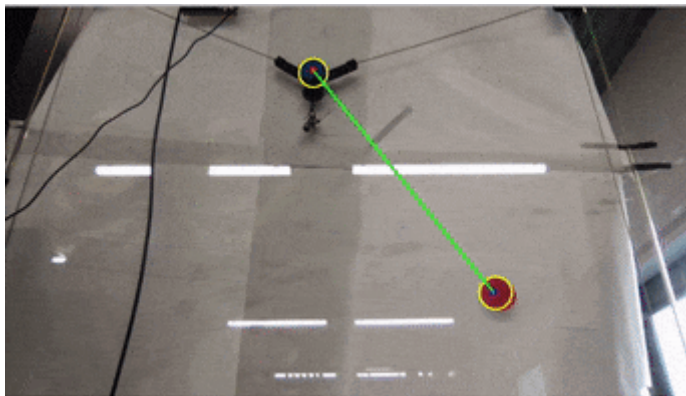


Рис. 3. Візуалізація роботи системи

### **Висновок**

Отже, застосування наведеного в тексті методу має декілька переваг. Один з головних плюсів полягає в тому, що двигуни можна розміщувати у роботі будь-яким чином. Крім того, завдяки алгоритму самоналаштування система може швидко зрозуміти, де знаходяться межі її робочої зони, що дозволить більш просте та пряме керування роботом.

Таким чином, застосування цієї технології може покращити ефективність та точність робота, а також зробити його більш зручним у використанні.

### **Список літератури**

1. Kober, J., Bagnell, J. A., & Peters, J. (2013). Reinforcement learning in robotics: A survey. *International Journal of Robotics Research*, 32(11), 1238-1274.
2. Sadati, N., Fathabadi, A., & Elahi, M. (2020). Deep reinforcement learning for robotic systems: A review. *Robotics and Autonomous Systems*, 124, 103333.
3. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529-533.
4. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
5. Gu, S., Holly, E., Lillicrap, T., & Levine, S. (2017). Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. *Proceedings of the IEEE international conference on robotics and automation*, 3389-3396.