

УДК62-519

## ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ УПРАВЛІННЯ РОБОТОТЕХНІЧНИМИ ПРИСТРОЯМИ НА БАЗІ ML- ТЕХНОЛОГІЙ

Ю. О. Лебеденко, кандидат технічних наук, доцент  
*Київський національний університет технологій та дизайну*

К. Р. Калініна, студент

*Київський національний університет технологій та дизайну*

О. П. Манойленко, кандидат технічних наук, доцент

*Київський національний університет технологій та дизайну*

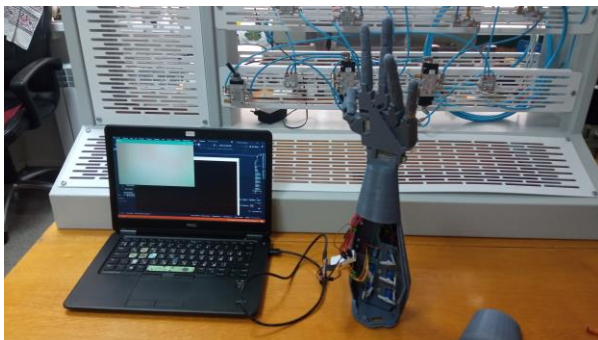
М. А. Дідик, студент

*Київський національний університет технологій та дизайну*

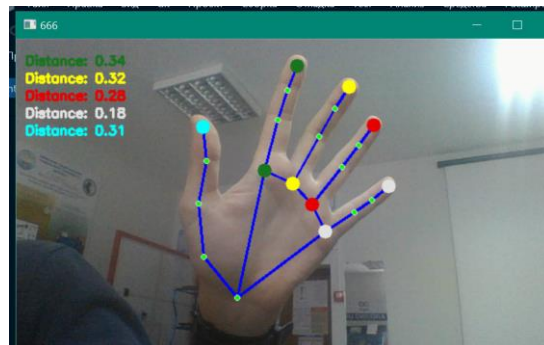
Ключові слова: штучний інтелект, Arduino, маски кисті, Python, рука-маніпулятор, машинний зір.

В роботі представлено системи керування маніпулятором «Cobot»[1], яка базується на ML-технологіях машинного зору та телеметрії. Система включає платформу Arduino [2], та програмне керування з алгоритмами штучного інтелекту[3] на основі машинного зору.

Особливість запропонованого методу керування є взаємодія з оператора з робототехнічними системами через web-камеру, що не потребує додаткових витрат на підключення, що робить систему досить зручною і водночас дешевою. З іншого боку процес передачі сигналів через внутрішні шини інтерфейсу до операційної системи ПК забезпечує ефективність та швидкість реакції. Система керування складається з web-камери, яка зв'язана через послідовний порт USB з контроллером USB (Рис. 1, а).



а)



б)

Рисунок 1 – а - Підключення ПК та руки -маніпулятора; б - Результат телеметрії кроссплатформеного фреймворку «MediaPipe» у вигляді орієнтації «скелета» на поверхні кисті руки оператора

Схема підключення та принцип роботи схеми керування на базі машинного зору наведена на рисунку 2. Програма запускає веб-камеру на комп'ютері, використовуючи бібліотеку «OpenCV», щоб виявити об'єкт (у нашому випадку, руку оператора). Якщо завдання виконано успішно, програма сегментує руку, використовуючи бібліотеку «MediaPipe», і

накладає маску скелета. Камера зчитує рухи руки оператора, і відповідно до позиції точок в коді шукається потрібна команда, що передається на плату Arduino Nano. Arduino Nano, у свою чергу, також шукає цю команду у своєму коді та передає її на сервопривод, який потрібно активувати. Такий процес контролю рухів руки оператора та передачі команд на плату Arduino Nano відбувається за допомогою телеметрії - системи для вимірювання та передачі даних в режимі реального часу.

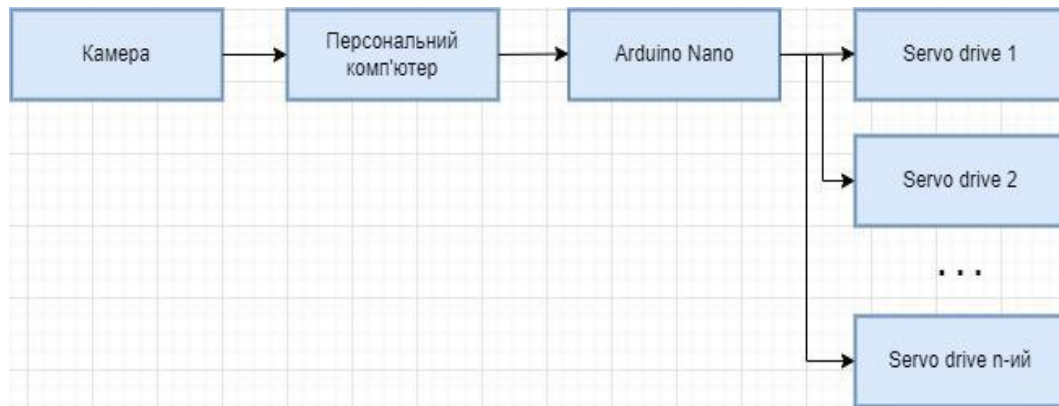


Рисунок 2 – Блок-схема керування сервоприводами на базі технологій машинного зору

Принцип роботи базується на телеметрії геометрії «скелета кисті» (Рис. 3), створеної на базі візуального скелета [4], де кожній точці 0-20 «скелета» кисті відповідають відповідні суглоби біомеханічної кисті руки-маніпулятора (шарніри).

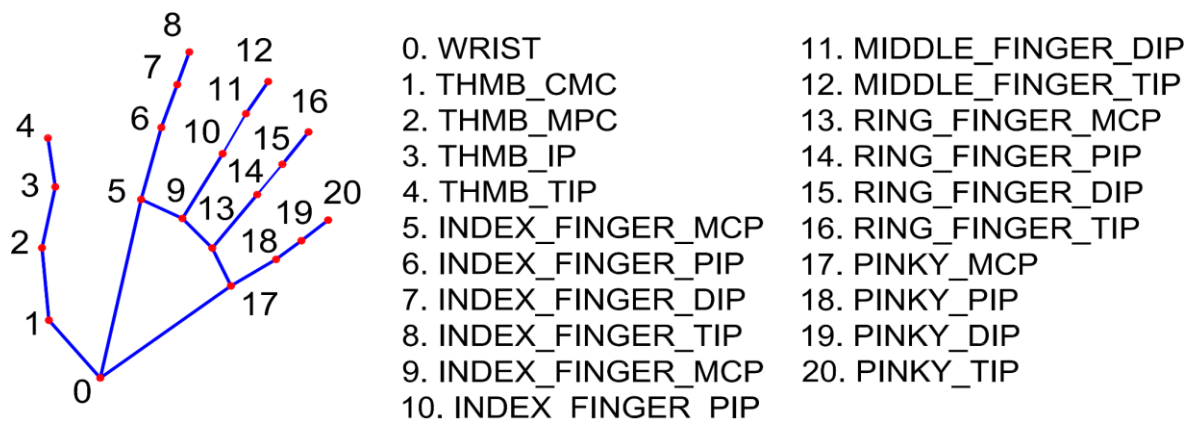


Рисунок 3 – Модель для розмітки «скелета» кисті

Програмне забезпечення, розроблене з використанням мов програмування Python [5] та Arduino. В програмному коді використано чотири основні фреймворки: OpenCV [6] для роботи з зображеннями, MediaPipe [7] для виявлення рук та жестів, NumPy [8] для обробки числових даних та Serial [9] для передачі даних через послідовний порт. Окрім цього стандартних фреймворків в програмному коді застосовані нестандартні фреймворки, які завантажуються вручну через командний рядок Windows.

Також фреймворки створюють потужність у розпізнаванні та обробці даних для подальшого керування маніпулятором за допомогою машинного зору оснований на телеметрії отриманого зображення з web-камери.

У програмному коді Python використовуються алгоритми, що дозволяють розпізнавати рухи руки оператора та відправляти відповідні керуючі сигнали до мікроконтролера Arduino Nano. Мікроконтролер, отримавши ці сигнали, активує відповідні серводвигуни для приводу механічних систем.

Висновок. Розроблена система керування маніпулятором «Cobot» на основі ML-технологій машинного зору та телеметрії є ефективною та інтуїтивно зрозумілою для керування рукою-маніпулятором і не потребує додаткової підготовки оператора. Результати роботи показують можливість та доступність використання машинного зору для керування подібними системами. Розроблене програмне забезпечення на базі Python та Arduino відкриває можливості для подальших досліджень, зокрема в плані підвищення надійності роботи та оптимізації швидкості обробки зображень та передачі даних. Запропоновані алгоритми керування можуть бути успішно застосовані для управління робототехнічними системами та мехатронними елементами.

#### Список використаних джерел

1. Michael A. Peshkin, J. Edward Colgate, Witaya Wannasuphoprasit, Carl A. Moore, R. Brent Gillespie, Prasad Akella Cobot Architecture. IEEE transactions on robotics and automation, 2001, 377–390
2. Arduino.cc офіційний довідник [Електронний ресурс] – Режим доступу <https://docs.arduino.cc/learn/>
3. Sulaiman Alsheibani Artificial Intelligence Adoption: AI-readiness at Firm-Level / Sulaiman Alsheibani, Yen Cheung, Chris Messom // Twenty-Second Pacific Asia Conference on Information Systems, 2018, 1–12
4. Лихошерстов Д.О., Лебедев Д.Ю. Порівняльний аналіз шляхів визначення зображення на відеоряді засобами машинного навчання. записки ТНУ імені В.І. Вернадського, 2022, 20–26. [https://tech.vernadskyjournals.in.ua/journals/2022/5\\_2022/3.pdf](https://tech.vernadskyjournals.in.ua/journals/2022/5_2022/3.pdf)
5. Python - офіційний довідник [Електронний ресурс] – Режим доступу <https://www.python.org/>
6. OpenCV-Python Tutorials [Електронний ресурс] – Режим доступу [https://docs.opencv.org/3.4/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/3.4/d6/d00/tutorial_py_root.html)
7. MediaPipe офіційний довідник: Gesture recognition task guide [Електронний ресурс] – Режим доступу [https://developers.google.com/mediapipe/solutions/vision/gesture\\_recognizer#configurations\\_options](https://developers.google.com/mediapipe/solutions/vision/gesture_recognizer#configurations_options)
8. NumPy офіційний довідник [Електронний ресурс] – Режим доступу <https://numpy.org/devdocs/user/>
9. PySerial офіційний довідник [Електронний ресурс] – Режим доступу <https://pyserial.readthedocs.io/en/latest/>