

**АЛГОРИТМ ОПЕРАТИВНОГО НАВЕДЕННЯ ЗАСОБІВ  
ВИМІРЮВАЛЬНО-КЕРУВАЛЬНОГО ВУЗЛА  
КІБЕРФІЗИЧНОЇ СИСТЕМИ  
НА РУХОМИЙ ОБ'ЄКТ**

**Д. О. Кушнір, Я. С. Парамуд**

Національний університет “Львівська політехніка”,  
кафедра електронних обчислювальних машин

© Кушнір Д. О., Парамуд Я. С., 2020

За результатами аналізу літературних джерел встановлено, що одними з основних вузлів кіберфізичних систем є вимірювально-керувальні вузли. Одним із завдань, розв'язання яких покладено на такі вузли, є наведення засобів спостереження за рухомими об'єктами. Запропоновано алгоритм наведення, який полягає в оперативному опрацюванні результатів спостережень, передбаченні найімовірнішого напрямку руху та формуванні команд для максимального наближення зображення рухомого об'єкта до центра інформаційного кадру. Розроблений алгоритм базується на алгоритмі навчання з підкріпленням DDPG. Засоби розпізнавання реалізують можливості моделі YOLOv3. Використані додаткові програмні фільтри для покращення якості розпізнавання. Алгоритм верифіковано на експериментальній фізичній моделі з використанням дрона. Результати експериментальних досліджень підтвердили функціонування алгоритму наведення в реальному часі.

**Ключові слова:** кіберфізична система, вимірювально-керувальний вузол, навчання з підкріпленням, алгоритм наведення, дрон.

**Вступ**

Останнім часом розроблення кіберфізичних систем (КФС) стає все популярнішим. Під кіберфізичною системою розуміють поєднання фізичних процесів та кібернетичних компонентів, які забезпечують організацію вимірювально-обчислювальних процесів, захищене зберігання та обмін вимірювальною і службовою інформацією, організацію та здійснення впливів на фізичні процеси [1]. Інтеграція таких компонентів в одну систему дає нам змогу створювати складні та ефективні технічні та сервісні інструменти.

Водночас, важливою частиною КФС є вимірювально-керувальний вузол (ВКВ). Загалом КФС – це набір ВКВ, яка взаємодіє з реальним середовищем і обладнана засобами спостереження та наведення [2].

Алгоритм оперативного наведення на рухомий об'єкт можна застосувати до одного з модулів ВКВ. Дослідження в цій області є важливою задачею, яку можна використати в системах відстеження рухомих об'єктів. Такі задачі можуть бути покладені на дрони, які мають на собі модуль ВКВ та здатні слідкувати за об'єктом, використовуючи технології штучного інтелекту.

Беручи до уваги вищенаведене, можна стверджувати, що робота у даному напрямку є своєчасною та актуальною.

### Огляд літературних джерел

Існує багато прикладних задач, де дрони можуть бути застосовані для відстеження та моніторингу об'єктів [3–4,12]. Серед них важливо виділити системи, які здатні наводити засоби спостереження на об'єкт та розпізнавати його [6]. Така система використовує алгоритмічний спосіб наведення на об'єкт, що значно обмежує масштабованість системи.

Було визначено, що для навчання системи автоматично відслідковувати рух об'єкта у довільному напрямку доречно використати алгоритми навчання з підкріпленням. Для навчання системи з неперервним простором дій доречно використати один з таких алгоритмів:

- *Deep Q-Network (DQN)* [7] – алгоритм, який комбінує Q-навчання із глибокими нейронними мережами, дає змогу розв'язувати високомасштабовані задачі в ігровій та робототехнічній індустрії.

- *Deep Deterministic Policy Gradient (DDPG)* [8] – це поліпшений алгоритм типу policy gradient, в навчанні з підкріпленням, який використовує модель Актор–Критик. Актор намагається навчитися рухатись у певному напрямку залежно від умов критика.

- *Proximal Policy Optimization (PPO)* [9] – алгоритм наближеної оптимізації стратегії. Основною ідеєю є те, що під час оновлень стратегії за градієнтами, оціненими з використанням IS (Importance Sampling), не можна допускати ситуацію, коли поточна стратегія та стара стратегія дуже відрізняються.

- *Soft Actor–Critic (SAC)* [10] – алгоритм по структурі наслідується від PPO, проте має ключову особливість – під час навчання додано мету підвищувати ентропію в policy gradient. Таким чином агент стає здатним діяти в більш випадкових ситуаціях середовища.

- *Target Candidate Track (TCT)* [11] – система алгоритмів стеження за об'єктами, який ґрунтується на стеженні за рухом об'єкта. Своєю чергою, базується на алгоритмі Калмана. Алгоритм намагається пов'язати положення нового розпізнаного об'єкта з положенням об'єкта у попередньому кадрі. Для того, щоб визначити, чи розпізнаний об'єкт є новим об'єктом чи новим положенням старого об'єкта, використовується відомий Угорський алгоритм, де вартість – це відстань між двома розпізнаними прямокутниками розпізнавання.

### Постановка завдання

Дослідити алгоритми оперативного наведення на рухомий об'єкт. Обраний алгоритм використати для тренування моделі, що використовує методи навчання з підкріпленням. Отриману модель використати для оперативного передбачення напрямку руху об'єкта. Скерувати засоби спостереження у потрібному напрямку залежно від результатів передбачення. Для розпізнавання об'єктів використати систему на базі архітектури моделі YOLOv3, використовуючи згладжувальні та мінімізаційні фільтри [12]. Дослідити роботу системи у реальному часі.

### Основні результати досліджень

Запропоновану схему, яка реалізує наведені вище задачі, показано на рис 1. Координати  $x$ ,  $y$ ,  $z$  будуть обчислені за допомогою алгоритму, що використовує метод машинного навчання з підкріпленням на основі відстані між центром середовища спостереження (модуля ВКВ) та центром розпізнаного об'єкта. Згідно з цими результатами дрон рухається у певному напрямку.

Для розв'язання поставленої задачі доцільно поділити її на два основні кроки:

**Детектування запропонованого об'єкта відслідковування.** Використовуючи згорткову нейронну мережу та зображення, отримане з камери засобів спостереження, визначають, чи присутній об'єкт на зображенні. За моделлю YOLOv3 знаходять координати знайденого об'єкта, які будуть використані для наступного кроку [4].

**Навігація.** Ця частина містить визначення команд, які потрібно надіслати дрону для коректного відслідковування запропонованого об'єкта. Для розв'язання цієї задачі пропонується алгоритм, в основу якого покладено навчання з підкріпленням. Саме цей алгоритм буде розглянуто детально у цій статті.

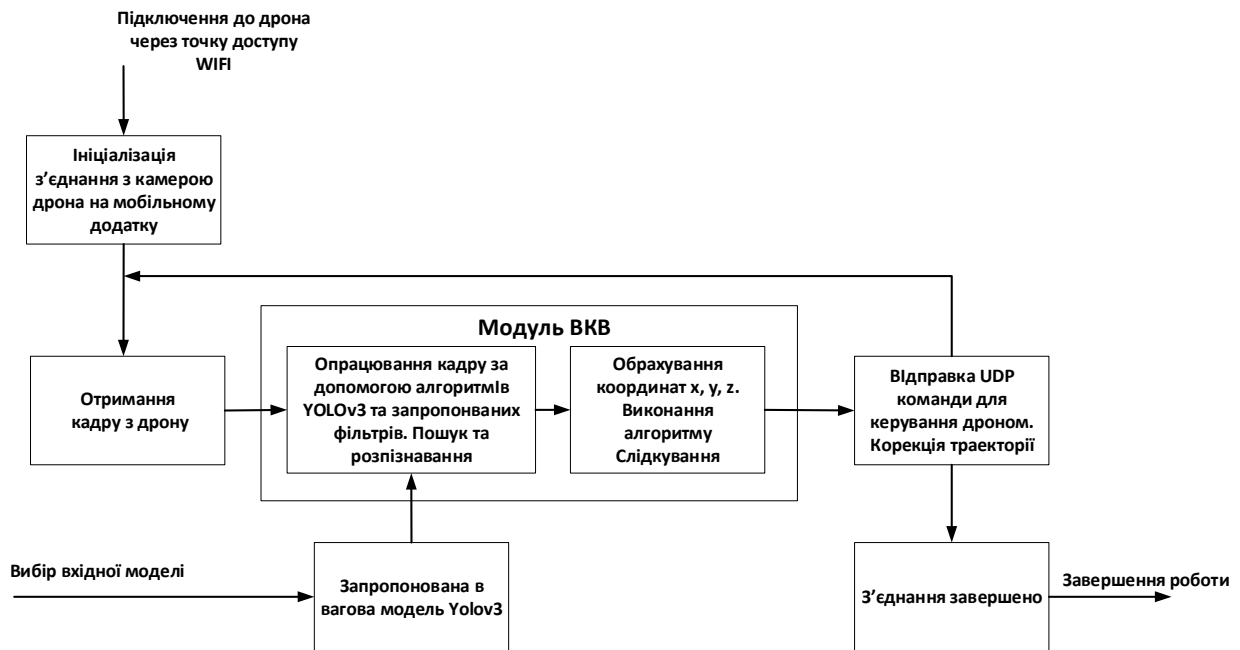


Рис. 1. Узагальнена структурна схема алгоритму пошуку та розпізнавання об'єктів

#### 1. Задання середовища для слідкування

Використовуючи різницю між центром екрана дрона та центром розпізнаного прямокутника, ми отримуємо відносні координати, які можна використати для моделі навчання з підкріпленням.

Середовища для навчання засобів спостереження:

- 1) середовище спостереження:  $0 \leq x \leq 960$ ;  $0 \leq y \leq 720$
- 2) середовище дій:  $-60 < A1 < 60$ ;  $-60 < A2 < 60$ .  $A1$  контролює дії руху ліворуч та праворуч,  $A2$  контролює дії руху донизу та догори по висоті.
- 3) центр екрана:  $x=480$ ,  $y=360$
- 4) засоби спостереження підтримують швидкість у діапазоні  $-100 < S < 100$

Наведені вище параметри необхідні для визначення координат руху  $x$  та  $y$ . (Схематично їх зображено на рис 2.)

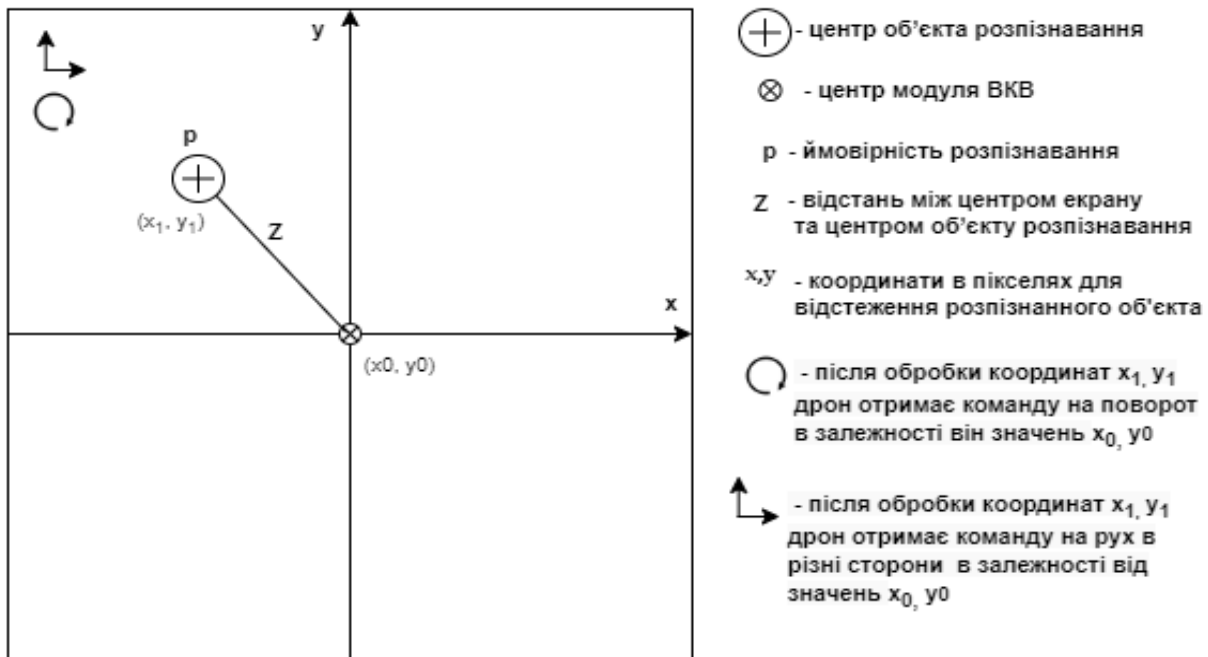


Рис. 2. Середовище керування засобами спостереження

## 2. Визначення руху за координатою z

Важливо зазначити, що алгоритм на основі навчання з підкріпленням застосовуватимуть лише для визначення руху за координатами  $x$  та  $y$ . Для переміщення засобів спостереження за координатою  $z$ , тобто вперед і назад, використовується окремий алгоритм.

Для цього спершу визначаємо відстань від засобів спостереження до об'єкта, використовуючи наведену формулу. Обчислюємо значення, яке являє собою відсоток площі, який займає прямокутник розпізнавання об'єкта відносно розміру екрана. Це значення обчислюють за формулою:

$$area\_p = \left( \frac{area}{frame\_h \times frame\_w} \right) \times 100, \quad (1)$$

де  $area$  – прямокутник з об'єктом розпізнавання;  $frame\_h$  – висота екрана;  $frame\_w$  – ширина екрана;  $area\_p$  – відсоткове співвідношення розмірів екрана пристрою та розмірів прямокутника з об'єктом розпізнавання.

Отримуємо алгоритм руху дрона по траєкторії назад та вперед:

- 1) якщо значення  $area$  менше ніж значення  $screen$  на 25 %, рухаємо дрон уперед;
- 2) якщо значення  $area$  більше ніж значення  $screen$  на 25 %, рухаємо дрон назад;
- 3) якщо значення  $area$  між 25 % та 50 % значення  $screen$  – дрон стоїть на місці.

## 3. Навчання моделі на базі DDPG агента

Для розв'язання поставленої задачі – розпізнавання та відслідковування об'єктів, – на основі аналізу літературних джерел було обрано модель DDPG, оскільки вона підходить для оцінювання стратегії дії (руху дрону у різних напрямках). На рис. 3 зображено структурну схему розробленої моделі.

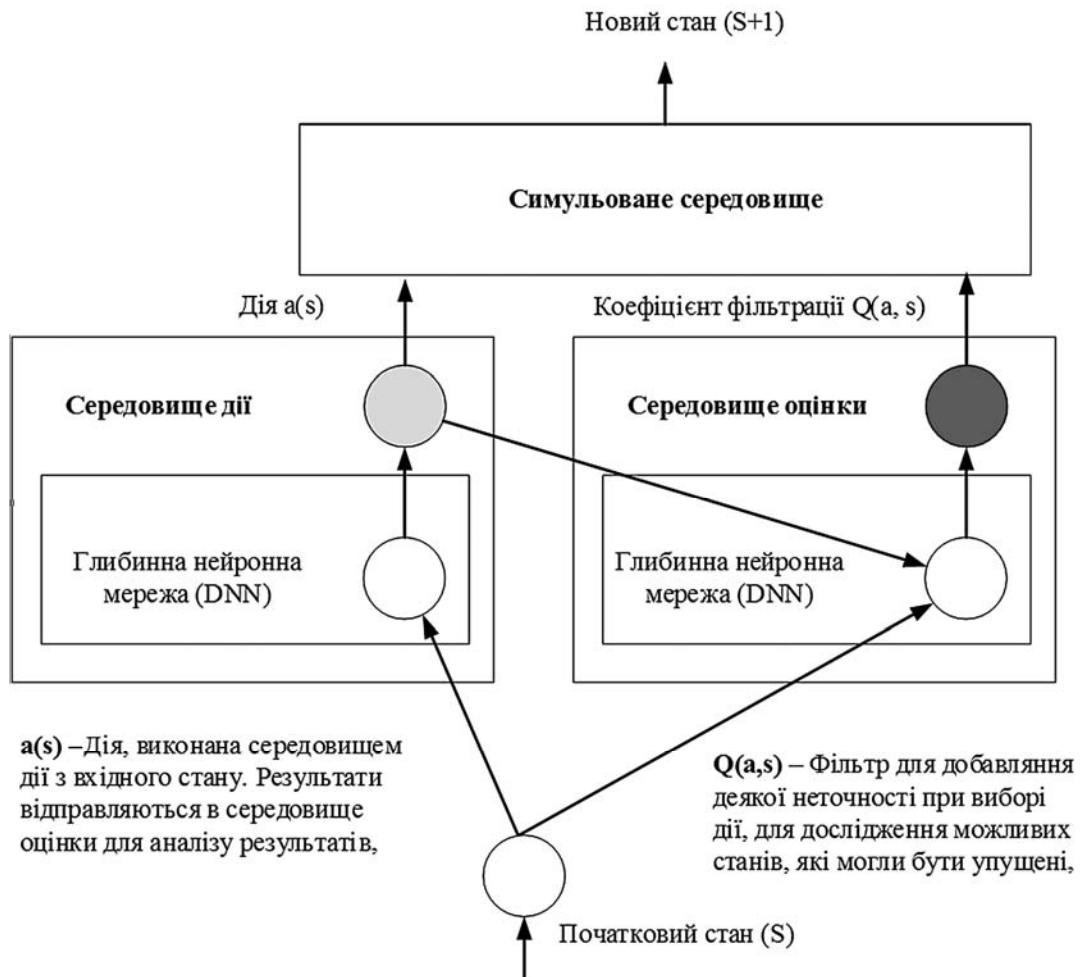


Рис. 3. Структурна схема запропонованого алгоритму навчання з підкріпленням

Щоб навчити DDPG агента, нам потрібно визначити функцію оцінки (винагороди), яка дозволить моделі зрозуміти, чи рухається вона в правильному напрямку. Як зазначалось вище, функція оцінки матиме на вході координати  $x$  та  $y$  центра розпізнаного об'єкта.

### Попереднє опрацювання даних

Впродовж тренування будуть генеруватись та подаватись на вхід випадкові координати розпізнаного об'єкта в діапазоні  $0 \leq x \leq 960$ ;  $0 \leq y \leq 720$ .

Для агента, який буде тренуватися, створено середовище на базі "Continuous Mountain Car". Це доречно, оскільки нас цікавить час, напрямок та швидкість пересування дрона.

Тренування агента ділять на незалежні епізоди, кожен з яких виконують покроково. Алгоритм кожного кроку (step function) наступний:

- 1) оновлення позицій враховуючі вхідні дії для генерування наступного стану;
- 2) обчислення нагороди;
- 3) перевірка чи стан кінцевий (done);
- 4) повернення наступного стану, нагороди та статусу завершення роботи.

Для визначення, чи стан кінцевий, необхідно визначити, чи виконується одна з наступних умов:

- 1) кількість виконаних кроків  $> 10$ ;

2)  $x$  чи  $y$  набуває значення, більшого за максимально встановлені в середовищі спостереження (у цьому випадку це  $960 \times 720$ );

3) координата центра екрана збігається із координатою центра блоку розпізнаного об'єкта.

Якщо стан кінцевий, епізод завершується і виконується функція скидування. Алгоритм функції скидування такий:

скидування координат, задання нових випадковим чином;

ініціалізація внутрішніх та кінцевого (done) станів.

Для визначення коефіцієнта функції оцінювання (dist) було використано таку формулу:

$$d_x = A_x - A_y$$

$$d_y = B_x - B_y$$

(2)

$$dist = \sqrt{d_x^2 + d_y^2},$$

де  $A_{x,y}$  – координати центра екрана;  $B_{x,y}$  – координати центра прямокутника з об'єктом розпізнавання;  $d_{x,y}$  – координати для обчислення відстані між двома точками;  $dist$  – результуючий вектор. Коефіцієнт функції оцінки.

Кінцеву формулу функції оцінки було оптимізовано методом спроб та помилок до вигляду:

$$reward = reward - (dist * 0.25), \text{ npu } (dist > 100)$$

(3)

$$reward = reward + (100 + dist), \text{ npu } (dist < 100),$$

де:  $reward$  – функція оцінки;  $dist$  – результуючий вектор. Коефіцієнт функції оцінювання.

За кожною ітерацією функція оцінювання буде збільшуватись або зменшуватись відповідно до того, чи в необхідному напрямку йде тренування.

Результати тренування для різних параметрів наведено на рис. 4.

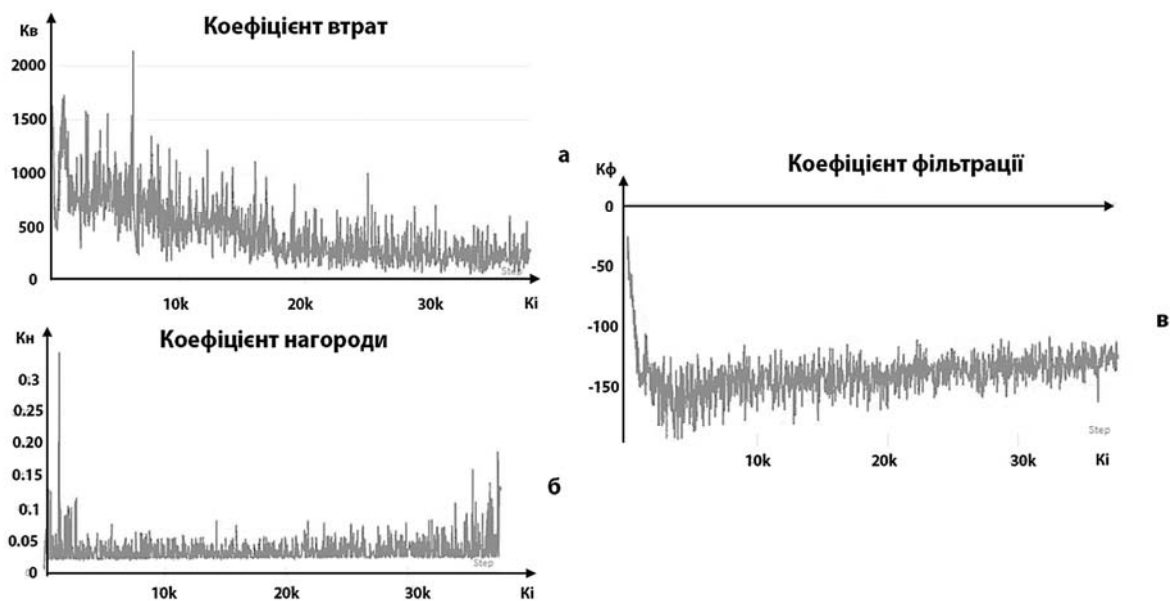


Рис. 4. Результати різних параметрів при тренуванні моделі навчання з підкріпленням відносно ітерації: а – коефіцієнт втрат; б – коефіцієнт нагороди (дія); в – коефіцієнт фільтрації

Коефіцієнт втрат вказує, наскільки система помилялася під час тренування. Коефіцієнт нагороди (дія) визначає, які значення нагороди подавалися моделі під час тренування. Коефіцієнт фільтрації використовується для додавання деякої неточності у результати залежно від ходу тренування.

У результаті тренування з вхідною функцією оцінювання (формула 3) отримано модель, яка натренована виконувати дії (рухи в різні боки) залежно від вхідних даних.

### Результати вимірювань

Отримані результати відсилаються у вигляді команди за протоколом UDP до дрону, який виконує відповідний маневр. Знаходячи декілька об'єктів на екрані, дрон вибирає найвірогідніший і слідує йому. Дослідження було проведено одночасно для таких параметрів:

1)  $x$ ,  $y$  – центральні координати розпізнаного об'єкта. Задача системи наблизитись до координат **center** – центра середовища розпізнавання. Ці координати передаються агенту навчання з підкріпленням. Результати вимірювань продемонстровано на рис. 5. Лінії штрихова та пунктирна прямують до координати 490 – центра екрана.

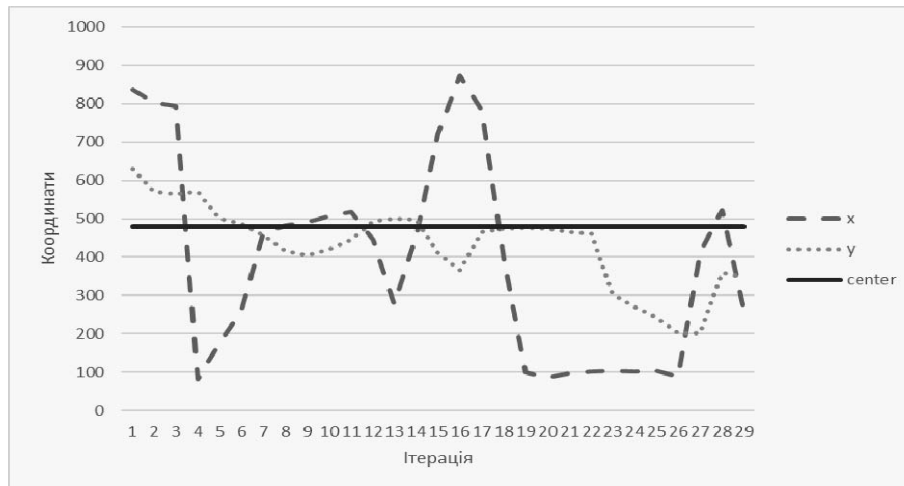


Рис. 5. Графік змін координат  $x$ ,  $y$  при експериментальних дослідженнях

2) **area\_p** – коефіцієнт відстані між об'єктом розпізнавання та середовищем розпізнавання. Обчислюється за формулою (1). Результати вимірювань продемонстровано на рис. 6. Безперервно намагається опинитись між двома лініями ( $\min_p$ ,  $\max_p$ ), якщо об'єкт змінює відстань до дрона.

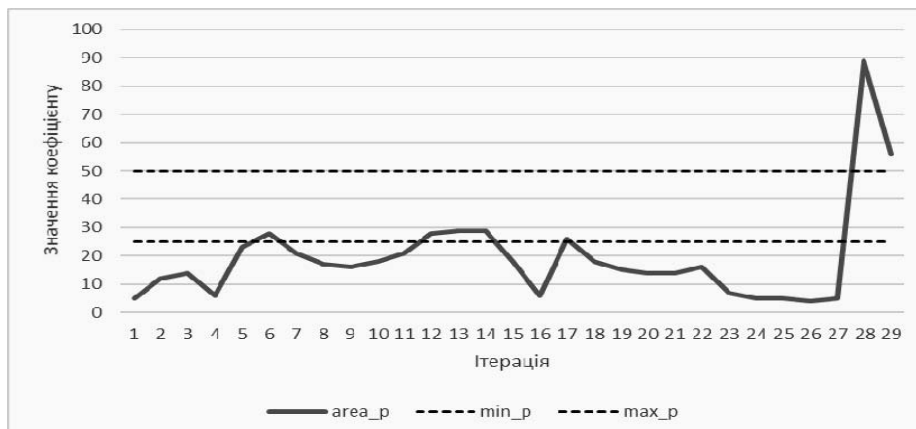


Рис. 6. Графік змін координат коефіцієнта відстані  $area_p$ .

3)  $A_x$ ,  $A_y$  – результуючі коефіцієнти руху, які відправляється як команди до дрона після обробки агентом DDPG. Результати вимірювань продемонстровано на рис. 6.  $A_x$  – зміна напрямку руху дрона за/проти годинникової стрілки.  $A_y$  – зміна напрямку руху дрона догори та донизу.

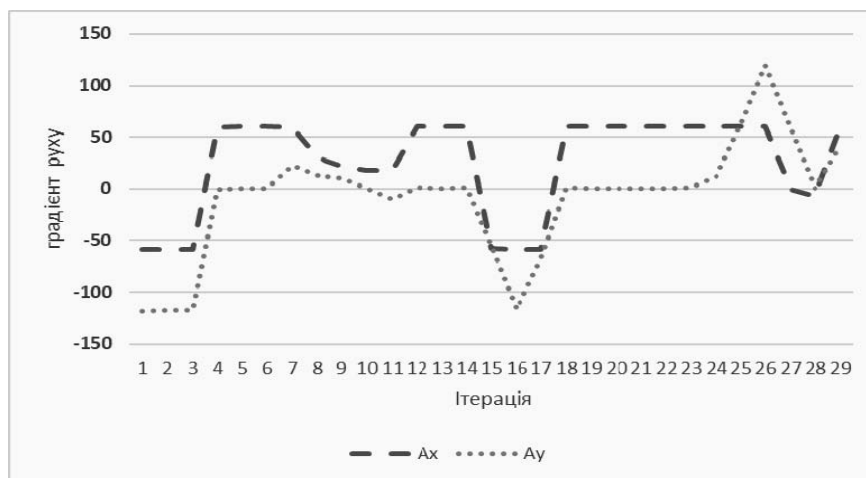


Рис. 7. Графік градієнтів руху  $A_x$ ,  $A_y$

Результати експериментальних досліджень показують, що система правильно реагує на зміну розташування об'єкта, відсилаючи команди руху вперед/назад (рис. 6), догори/донизу та відносно руху годинникової стрілки (рис. 7). Додатні градієнти вказують на рух за годинниковою стрілкою та догори, від'ємні – проти годинникової стрілки та вниз. Існує деяка затримка (від пів секунди до секунди), пов'язана з швидкістю передавання команд до дрона, проте це може бути вирішено використанням високошвидкісного WIFI адаптера або застосуванням більш швидкодіючих засобів обміну інформацією.

### Висновки

У роботі запропоновано алгоритм оперативного наведення засобів спостереження на рухомий об'єкт. Використано базову модель DDPG. Проведено тренування запропонованої моделі навчання з підкріпленням для оперативного наведення засобів спостереження на об'єкт у відкритому середовищі. Запропоновані засоби досліджено експериментально за допомогою відомої моделі YOLOV3. У роботі представлено реалізацію алгоритму оперативного наведення на рухомий об'єкт як частину модуля вимірювально-керувального вузла КФС. Проведено тестування розробленої моделі на реальному середовищі, використовуючи дрон TELLO. Отримано позитивні результати роботи моделі. Дрон реагує швидко (в середньому 0,3 секунди) на зміну розташування об'єкта, проте іноді існує невелика затримка в 0,5–1 секунди, пов'язана з передаванням команд через мережу WIFI. Загалом, запропонований алгоритм оперативного наведення засобів вимірювально-керувального вузла КФС на рухомий об'єкт показав свою здатність щодо застосування в реальному середовищі.

### Список літератури

1. Melnyk A., (2016, November). *Cyber-physical systems multilayer platform and research framework. Advances in Cyber-Physical Systems [Online]. Available: <http://science.lpnu.ua/acps/all-volumes-and-issues/volume-1-number-1-2016/cyber-physical-systems-multilayer-platform-and>*
2. Botchkaryov O., Golembo V., Paramud Y., Yazuk V., *Cyber-physical systems: technologies of data collection [Text]: monography – O. Botchkaryov, V. Golembo, Y. Paramud, V. Yazuk. Editorial chiev: prof. A. Melnyk, Lviv: Magnolia 2006, 2019. 176 p. pp. 10–12.*



3. Koubaa A., Qureshi B., (2018, March). DroneTrack: Cloud-Based Real-Time Object Tracking using Unmanned Aerial Vehicles, *IEEE Access [Online]*. Available: <https://doi.org/10.1109/ACCESS.2018.2811762>
4. Ding G., Zhang L., Lin Y., Tsiftsis T., Yao Y. (2018, January). An Amateur Drone Surveillance System Based on the Cognitive Internet of Things, *IEEE Communications Magazine [Online]*. Available: <https://doi.org/10.1109/MCOM.2017.1700452>
5. Pons P., Jaen J., Catala A. (2015, November). Developing a depth-based tracking system for interactive playful environments with animals, *ACE '15: Proceedings of the 12th International Conference on Advances in Computer Entertainment Technology [Online]*. Available: <https://doi.org/10.1145/2832932.2837007>
6. Kushnir D., Paramud Y. (2020, June). The algorithm of Cyber-Physical system targeting on a movable object using the smart sensor unit, *Scientific-Technical Journal "Advances in Cyber-Physical Systems"*. Vol. 5, No. 1, 2020.
7. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, Demis Hassabis (2015, February). Human-level control through deep reinforcement learning [Online]. Available: <https://doi.org/10.1038/nature14236>
8. Josh Achiam (2020 January). Deep Deterministic Policy Gradient [Online]. Available: <https://spinningup.openai.com/en/latest/algorithms/ddpg.html>
9. John Schulman, Oleg Klimov, Filip Wolski, Prafulla Dhariwal, Alec Radford (2017 July). Proximal Policy Optimization [Online]. Available: <https://openai.com/blog/openai-baselines-ppo/>
10. Josh Achiam (2020 January). Soft Actor-Critic [Online]. Available: <https://spinningup.openai.com/en/latest/algorithms/sac.html>
11. Eren Unlu, Emmanuel Zenou, Nicolas Riviere, Paul-Edouard Dupouy (July 2019). Deep learning-based strategies for the detection and tracking of drones using several cameras [Online] Available: <https://doi.org/10.1186/s41074-019-0059-x>
12. D. Kushnir, Y. Paramud, (2019, November). Methods for real-time object searching and recognizing in video images on ios mobile platform. *Computer Systems and Networks. Vol. 1, Number 1. [Online]*. 1. (1). pp. 24-34. Available: <https://doi.org/10.23939/csn2019.01.024>

## THE INTELIGENE ALGORITHM OF CYBER-PHYSICAL SYSTEM TARGETING ON A MOVABLE OBJECT USING THE SMART SENSOR UNIT

D. Kushnir, Y. Paramud

Lviv Polytechnic National University,  
Computer Engineering Department

© Kushnir D., Paramud Y., 2020

As a result of the analytical review, it was established that smart sensor units are one of the main components of the cyber-physical system. One of the tasks, which have been entrusted to such units, are targeting and tracking of movable objects. The algorithm of targeting on such objects using observation equipment has been considered. This algorithm is able to continuously monitor observation results, predict the direction with the highest probability of movement and form a set of commands to maximize the approximation of a moving object to the center of an information frame. The algorithm, is based on DDPG reinforcement learning algorithm. The algorithm has been verified on an experimental physical model using a drone. The object recognition module has been developed using YOLOv3 architecture. iOS application has been developed in order to communicate with the drone through WIFI hotspot using UDP commands. Advanced filters have been added to increase the quality of recognition results. The results of experimental research on the mobile platform confirmed the functioning of the targeting algorithm in real-time.

**Key words:** Cyber-physical system, smart sensor unit, reinforcement learning, targeting algorithm, drones.