

А. О. Каргін^{1,2}, С. В. Жуков¹, Д. А. Сергєєв¹, Є. Л. Сілін²

¹Харківський національний технічний університет радіоелектроніки, Харків, Україна

²Український державний університет залізничного транспорту, Харків, Україна

МОДЕЛЬ БАЗОВОГО РІВНЯ ШТУЧНОГО ОТОЧЕННЯ АВТОНОМНИХ ІНТЕЛЕКТУАЛЬНИХ БЕЗЛЮДНИХ СИСТЕМ НА ПРИКЛАДІ МОБІЛЬНОГО РОБОТУ ЩО ОБСЛУГОВУЄ

Анотація. Мотивація. Незважаючи на значний прогрес у галузі створення безлюдних систем, забезпечення необхідного рівня їх автономності залишається актуальним завданням. Важливу роль у його вирішенні відіграє штучний інтелект. Особливості безлюдних систем породили нову модель «Штучний інтелект що відчуває» (ШІВ), що підтримує автономність. **Метою даної роботи** є створення штучного оточення для експериментів з безлюдними системами, підтримуваними ШІВ, на прикладі складу контейнерів, що обслуговується колесним роботом. **Предметом дослідження** є методи та моделі управління автономними мобільними роботами на підставі даних від різноманітних сенсорів. **Результати.** Описується архітектура штучного оточення, базові компоненти системи що мають багатошарову організацію. На прикладі штучного оточення «Склад контейнерів», як прототипу автономної безлюдної системи що обслуговується колесним роботом, показано набір контролерів, приклад схем підключення сенсорів та актуаторів, методи та алгоритми управління, які потрібні на базовому рівні. Наведено обґрунтування методу ідентифікації номерів контейнерів що задовольняє вимогам, що накладаються автономними системами. **Висновок.** Доведено що архітектура базового рівня штучного оточення має бути універсальною щодо підтримки різних методів й алгоритмів управління на базі різноманітних даних від сенсорів. Універсальність досягається завдяки багатошаровій організації контролерів за для підтримки функцій ШІВ. Експерименти зі контролерами базового рівня штучного оточення «Склад контейнерів» продемонстрували можливість підвищення рівня автономності безлюдної системи за рахунок розширених можливостей використання в управляючій програмі робота різноманітних методів управління на різних етапах плану із урахуванням поточної ситуації, що оцінюється на множині даних від різноманітних сенсорів. **Напрямок подальших досліджень.** Майбутні кроки можуть бути присвячені розвитку моделі та інструментарію для інших класів безлюдних систем.

Ключові слова: автономні інтелектуальні безлюдні системи, штучний інтелект що відчуває, штучне оточення, програмне управління, робот, сенсор, розпізнавання номерів контейнерів.

Вступ

В сучасному світі транспортування товарів є важливою складовою бізнесу в багатьох галузях промисловості. Товари що транспортуються упаковуються у контейнери. Технологія транспортування передбачає проміжні етапи тимчасового зберігання контейнерів на складах, обслуговування яких автоматизовано. Більшість сучасних складів відносяться до безлюдних систем (БС) [1, 2]. Автоматичні склади, як БС, передбачають автономне функціонування у тому числі й в непередбачених ситуаціях, коли системи керування різноманітним обладнанням складу повинні приймати рішення без втручання людини. Такі склади в залежності від спеціалізації обладнанні мобільними роботами, які виконують, як завдання по транспортуванню контейнерів, так й допоміжні функції, наприклад, додатковий облік й контроль наявності певних контейнерів на певних місцях й таке інше [2]. Мобільні роботи (МР) мають переміщуватися до вказаної позиції складу по певному маршруту й використовуючи зорову камеру ідентифікувати місце, наявність контейнеру певного типу й його номер для подальшої обробки системою обліку. За для того, щоб не порушувати цикли основних технологічних процесів МР повинні орієнтуватися в поточній ситуації, уникати аврійних ситуацій й виконувати своє завдання в автономному режимі.

Незважаючи на гігантський прогрес у галузі створення БС, таких як безпілотні автомобілі [3], апарати військового призначення [4] та інші системи, забезпечення необхідного рівня автономності БС

залишається актуальним завданням [5]. Важливу роль у вирішенні цього завдання відіграє штучний інтелект (ШІ). Автономні системи з підтримкою штучного інтелекту пройшли у своєму розвитку етапи програмування автономних систем та інтелектуальних БС. Сьогодні є актуальними автономні інтелектуальні БС (АІБС) [5, 6]. На ШІ в АІБС покладають рішення багатьох завдань з обробки інформації: отримання та первинна обробка даних від сенсорів, формування моделі оточення, планування дій у цьому оточенні та прийняття рішень, що призводять до досягнення мети, реалізація реакцій, що управляють, навчання та адаптація до змін в оточенні [7, 8]. Ця специфіка знайшла відображення у класифікації ШІ, який пропонується для АІБС. В роботах [5, 9] вводиться три типи штучного інтелекту – механічний, мислячий і той що відчуває. Механічний ШІ використовується для простих, стандартизованих, повторюваних і рутинних завдань. Мислячий ШІ використовується для складних, систематичних, заснованих на правилах і чітко визначених завдань. ШІ що відчуває, зокрема у військовій сфері, обговорюється у зв'язку з необхідністю АІБС вирішувати соціальні, емоційні, комунікативні та інтерактивні завдання. В роботі [10] пропонується модель і проект архітектури ШІВ не з точки зору прийняття рішень, які відповідають соціальним людським цінностям, а з технічної точки зору для розширення можливостей ШІВ щодо забезпечення необхідного рівня автономії АІБС.

Для тестування моделей ШІВ потрібні різного роду полігони [11], де створюється штучне оточення АІБС що моделює різноманітне застосування БС.

Метою цієї роботи є розробка штучного оточення АІБС на прикладі автоматизованого складу контейнерів. МР, як складова компонента автоматизованого складу контейнерів, розглядається у штучному оточенні як прототип АІБС. На базі колесного робота потрібно створити фізичну й програмну модель штучного оточення «Склад контейнерів» для подальших експериментів з АІБС що вирішує завдання аналізу ситуацій, ситуаційного планування маршруту й управління переміщеннями в умовах перешкод, розпізнаванню номерів контейнерів у різних умовах експлуатації.

Основна частина

Архітектура штучного оточення «Склад контейнерів». Створюється штучне оточення на прикладі автоматизованого сховища з різними секціями для вантажних контейнерів. Загальні вимоги до штучного оточення наступні.

У секціях сховища встановлені маркери відповідних номерів вантажних місць де зберігаються контейнери (ідентифікатори місця секції) та системи освітлення для забезпечення якісного зображення номеру контейнера у будь який час доби [1]. Для забезпечення безперебійної роботи МР, необхідно мати доступ до електричної мережі (позиції зарядки акумуляторних батарей, зарядної станції), а також бездротову мережу зв'язку, щоб передавати отримані дані для подальшої обробки та аналізу. Важливо також щоб МР мав права доступу до системи управління сховищем, щоб забезпечити швидкий й безпечний доступ к хмарним сховищам даних.

Для виконання основних функцій МР створюється штучна секція сховища для вантажних контейнерів. Приміщення секції поділяється на 3 основні зони з маршрутами «А-а», «А-б» та «А-с» та нумеровані вантажні місця від нуля до шести. Робот може курсувати між рядами, до будь-якого з контейнерів за маршрутом зображеним на рис. 1.

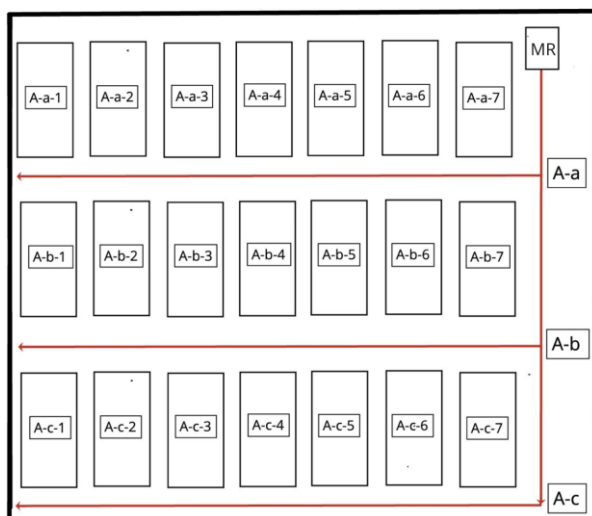


Рис. 1. Схема секції складу вантажних контейнерів

Підлога має розмітку для орієнтування МР. Робот, позначений на рисунку літерою «MR», має доступ до кожного вантажного місця. Вони позначені пунк-

тирною лінією – це є місце де, як вважається, контейнери мають стояти. Надписом «СН» позначена зарядна станція робота, яка винесена у окрему зону у приміщенні. Секція має одну зарядну станцію. МР може обслуговувати декілька секцій упродовж дня, періодично заряджаючись між патрулюванням та перевірки.

Для експериментів обрано два типу МР з різним апаратним забезпеченням. Перший тип МР на базі чотирьох колісного повнопривідного шасі Multi Chassis-4WD Robot Kit ATV, мікроконтролеру Arduino Mega з набором Motor shield, модулем wi-fi esp8266, матрицею із 10-ти інфрачервоних датчиків відображення ku-033 для управління переміщеннями вздовж розмітки на підлозі, ультразвуковим датчиком HC-SR04, встановленим на поворотній платформі з сервопривідом SG90, та датчиком одометром H206.

Другий тип МР реалізований на основі Raspberry Pi навчальний набір (PiCar-V для побудови роботів з відкритим вихідним кодом, оснащений ширококутною USB веб-камерою та укомплектований трьома новими платами мікроконтролеру Raspberry Pi, яка має вбудований порт для камери з високою роздільною здатністю що дозволяє працювати з відео та фото без додаткових інструментів або пакетного програмного забезпечення). Плата має достатньо велику оперативну пам'ять для складних обчислень, такі як зчитування інформації з зображення у режимі реального часу. Операційна система Linux прискорює процес розробки окремих компонентів системи на більш релевантних для типу задачі мовах.

Компоненти системи управління переміщеннями мобільного робота. ШІВ має шарову архітектуру [11, 12]. До нижнього першого шару належать програмні інтерфейси з апаратним забезпеченням АІБС, тобто драйвери сенсорів й актуаторів. Для штучного оточення що розглядається в данній роботі це драйвери датчику одометра, ультразвукового датчику відстані та матриці датчиків відображення що влаштовані на днищі шасі робота, а також драйвери сервопривіду поворотної платформи та приводів двигунів Motor Shield. До другого шару належать контролери руху. Це контролер руху вздовж розмітки на підлозі (K1), контролер руху вздовж штучної огорожі (K2), контролер руху до візуального маркера по вільній траєкторії (K3) та контролери відстані (S) й швидкості (V) руху. До третього шару належить контролер програмного управління рухом що реалізує план (програму в термінах управління роботами) досягнення кінця цільової точки. І на верхньому четвертому шарі знаходяться інтелектуальні компоненти прийняття рішень в умовах неповної інформації [12] що застосовують когнітивні моделі сприйняття даних від сенсорів та емоційної оцінки ситуації для прийняття рішень [10].

Перелічені вище контролери розроблені на мові C++ в середовищі Arduino IDE. Роль драйверів відіграють вбудовані функції та бібліотеки Arduino IDE. Розглянемо організацію драйверів й контролерів на прикладі одометра. При переміщеннях колесного робота одометр збирає дані про обертання колес. Дані обробляються мікроконтролером для визначення пройденої відстані та швидкості робота.

Серед трьох типів одометрів (механічні, електро-механічні та електронні) обрано електронний що представляє собою інфрачервоний датчик H206 та диск з 20 отворами, який кріпиться на вісь двигуна (рис. 2).

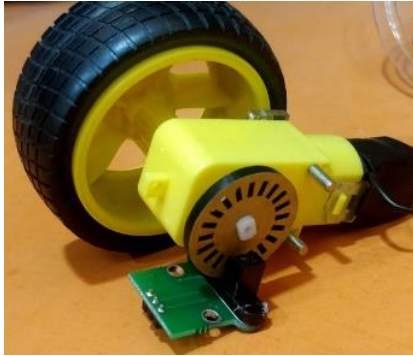


Рис. 2. Апаратна реалізація датчика одометра

Диск розміщується між інфрачервоним випромінювачем та приймачем (датчик H206) і при кожному проходженні отворів диска датчик реагує на зміну його сигналу. Оскільки отворів у диска 20, а датчик реагує на оби дві зміни сигналу, то за час повного обороту колеса МР мікроконтролер нарахує 40 переривань. Відстань, пройдена роботом за один оберт колеса, дорівнюватиме довжині його кола. Для радіуса колеса r пройдена відстань обчислюється за формулою:

$$S = 2\pi r * (n/40), \quad (1)$$

де n – кількість переривань.

Датчик H206 підключається до піну переривання (цифрові піни 2 та 3 на рис. 3) і його показання фіксує вбудована функція *attachInterrupt()*.

Таким чином роль драйвера датчика одометр виконує ця функція.

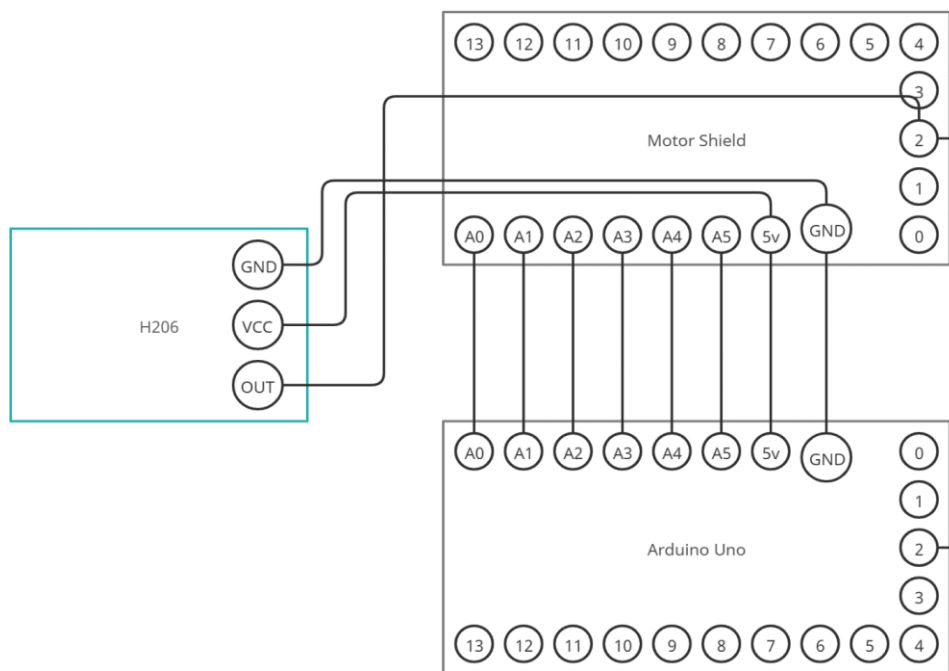


Рис. 3. Схема з'єднання компонентів на базі датчика H206

Контролер відстані видає по запиті поточне значення пройденої відстані (1) в метрах, реалізує функції обнулення $n=0$ в (1) та переривання при досягненні значення відстані S^* , яке задається.

Контролер швидкості підтримує поточну лінійну швидкість руху роботу рівною заданому значенню v^* . Використовується функція *millis()*, значення якої є кількість мілісекунду починаючи з часу подачі живлення на плату Arduino. Підраховується кількість мілісекунд t на кожний повний оберт колеса й обчислюється миттєва швидкість v в одиницях м/сек:

$$v = 2\pi r * 1000/t. \quad (2)$$

Знаходиться похибка управління

$$e = v * -v \quad (3)$$

й застосовуючи один з алгоритмів управління зі зворотними зв'язками корегуються оберти колес.

Контролери руху реалізують управління переміщеннями роботу окремих етапів плану переміщення. Контролери руху вздовж розмітки на підлозі використовує дані від драйверу матриці датчиків відображення на їх підставі обчислюється відхилення робота від лінії розмітки, тобто похибка управління, й на її основі знаходить значення управляючого сигналу. Аналогічно два інших контролерів руху обчислюються похибки управління на основі даних від ультразвукового датчику й відео камери, відповідно. Контролери реалізують управління зі зворотними зв'язками застосовуючи налаштовані ПІД алгоритм чи нечіткі алгоритми, такі як нечіткий лінійний статичний із зоною нечутливості й насиченням, нечіткий трьохпозиційний, нечіткий нелінійний із гістерезисом й насиченням, нечіткий ПІД.

Контролер програмного управління виконує функцію менеджера окремих етапів плану досягнення

цільової позиції. Маршрут від поточної точки місцезнаходження робота до цільової розбивається на окремі етапи, послідовність яких уявляє управляючу програму роботи. Контролер програмного управління активізує етапи управляючої програми шляхом активізації команди що вказана, контролює завершення виконання етапу (успішне чи аврійне), та приймає рішення щодо активізації наступного етапу. В управляючій програмі (УП) можуть бути етапи з різними методами управління, на відміну від класичного програмного управління. Етапи що реалізують жорстке програмне управління, коли умовою завершення етапу є час (в опису етапу УП задається інтервал часу) можуть перемішуватися з етапами що реалізують гнучке програмне управління, коли умовою завершення етапу є пройдена відстань, чи відстань до перешкоди, чи наявність певного маркера що ідентифікується камерою зору. Контролер цього рівня виконує моніторинг даних від відповідних датчиків (одометр, ультразвуковий та відео) й порівнює поточні дані з заданими в УП. При виконанні умов, вказаних в описі етапу, ініціюється завершення етапу УП. Жорстке програмне управління порівняно з гнучким менш ресурсоємне але може застосовуватися в умовах відсутності перешкод за підтримки контролерів нижчого шару що мають високі значення критерію якості управління.

Обов'язковою компонентою що належить до четвертого шару компонент зі штучним інтелектом є планувальник завдань, який в залежності від стану (ситуації) штучного оточення генерує УП, як послідовність етапів досягнення цілі. Планувальник завдань реалізує метод безперервного планування за прецедентами [10, 12]. Безперервне планування передбачає що після завершення кожного етапу УП (успішно чи аварійно) планувальник виходячи з поточної ситуації по новому генерує план досягнення цілі. Такий підхід забезпечує високий рівень автономності прийняття рішень в умовах збурень та неповної інформації.

Взаємозв'язки різного рівня контролерів при реалізації МР функції контролю в штучному оточенні «Склад контейнерів» розглянемо на наступному

прикладі. Нехай при виконанні УП після завершення поточного етапу МР знаходиться у позиції A-a-3 (рис.4).

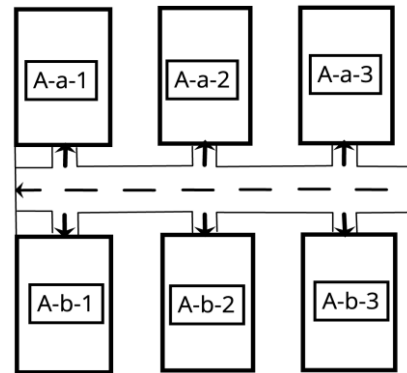


Рис. 4. Приклад маршруту переміщення

Планувальник сгенерував УП переміщення МР до вантажної позиції A-b-1. Приклад трьох типів УП наведено в табл. 1. Методи програмного управління роботом демонструються на прикладі обмеженого набору команд управління: вперед, праворуч, ліворуч. В методі жорсткого програмного управління видача команди викликає початок відповідного руху. Цей рух триває до тих пір, поки не буде видана наступна команда вимкнути попередньо увімкнутий рух. Для методу жорсткого програмного управління вважається, що переміщення здійснюються з постійними швидкостями й тривають константний час. Наприклад, t_{A-a-3} секунд потрібно часу для переміщення МР від місця моніторингу номеру контейнера на позиції A-a-3 (рис.4) до лінії розмітки A-a (рис. 1). До складу команд методу жорсткого програмного управління (інтерпретація програми управління роботом - жорстка послідовність виконання команд: i-a по порядку команда в програмі не може почати виконуватися, доки не буде завершено виконання i-1-ї команди) обов'язково входить затримки, наприклад, *затримка* t_{A-a-3} з параметром часу. Інтерпретатор команд контролеру при обробці цієї команди затримує на t_{A-a-3} одиниць часу початок виконання наступної команди.

Таблиця 1 – Приклади управляючих програм

Жорстке програмне управління		Гнучке програмне управління	Гнучке ситуаційне програмне управління
<i>вперед</i>	<i>затримка</i> t_{1-3}	<i>вперед</i> K1 $S=120$	<i>вперед</i> K1 $R=f, S=50, V=1$
<i>затримка</i> t_{A-a-3}	<i>вимк_вперед</i>	<i>праворуч</i> $\gamma=90^\circ$	<i>праворуч</i> $\gamma=90^\circ$
<i>вимк_вперед</i>	<i>ліворуч</i>	<i>вперед</i> K1 $S=500$	<i>вперед</i> K2 $R=h, S=50, V=2$
<i>праворуч</i>	<i>затримка</i> t_{90°	<i>ліворуч</i> $\gamma=90^\circ$	<i>ліворуч</i> $\gamma=90^\circ$
<i>затримка</i> t_{90°	<i>вперед</i>	<i>вперед</i> K1 $S=120$	<i>вперед</i> K1 $R=k, S=50, V=1$
<i>вимк_праворуч</i>	<i>затримка</i> t_{A-b-1}	<i>стоп</i>	<i>стоп</i>
<i>вперед</i>	<i>вимк_вперед</i>		
	<i>стоп</i>		

Метод жорсткого програмного управління вимогливий до попереднього впорядкування середовища: обов'язкові жорсткі вимоги до швидкості переміщення і миттєві режими розгону/гальмування робота. Ці вимоги для колесних роботів практично ніколи не можливо виконати. У методі програмного управління із зворотними зв'язками (гнучкого

програмного управління) для ідентифікації події виходу робота на певні позиції використані дані від датчиків, а не інтервали часу. В табл. 1 наведено приклад УП, в якій використовуються дані від одометрів (колес роботу ($S=120\text{см}$) і сервоприводу розвороту ($\gamma=90^\circ$). В УП указано що використовується контролер управління K1 (рух вздовж розмітки на підлозі на

підставі даних від датчиків відображення). Умова вимикання команди задається конкретним значенням пройдені МР відстані, наприклад, $S=120\text{см}$ у першій команді *вперед* в Таблиці 1. Контролер К1, отримуючи у реальному часі дані від драйверу одометра, розраховує поточну відстань (1) й видає сигнал вимкнути_вперед, коли значення відстані дорівнює чи більш вказаного, тобто, $S \geq 120$.

В методі гнучкого ситуаційного програмного управління є можливість задавати будь яку ситуацію й будь який контролер. В прикладі (Таблиця 1) на першому етапі УП використовується контролер К1, третій – контролер К2, п'ятий - знову К1. При реалізації першого етапу, крім контролера К1, паралельно працюють контролери швидкості ($V=I$) й відстані ($S=50$) та контролер руху вздовж смуги, який додатково відстежує наявність маркеру f на підлозі, що позначає перехрестя лінії розмітки від контейнера $A-a$



Рис. 5. Приклади контейнерів й область з їх номером

Перший метод на основі бібліотек OpenALPR (Automatic License Plate Recognition), алгоритму Haar Cascade та бібліотеки OpenCV. Бібліотека OpenALPR використовується в системі розпізнавання різного типу номерів, тому числі контейнерів. OpenALPR використовує навчальні дані, що містять інформацію про різні шрифти та символи знаків з різних країн. Бібліотека використовує алгоритми комп'ютерного зору для локалізації номерних знаків на зображенні та витягування з них текстової інформації. Вона налаштована для використання у режимі реального часу, що дозволяє розпізнавати номерні знаки на відео з використанням веб-камер. OpenCV - бібліотека відкритого коду для комп'ютерного зору та обробки зображень. Використовується для розв'язання різноманітних задач, пов'язаних з обробкою зображень, таких як детекція облич, розпізнавання тексту, відслідковування об'єктів та інші [13].

Запропоновано на отриманому зображенні з камери, використовуючи OpenCV та Haar Cascade Classifier, знаходити зображення номеру контейнера на відео та вирізати цей фрагмент, зберігаючи як окремий файл зображення. Якщо область успішно визначена, фрагмент передається до OpenALPR для розпізнавання. OpenALPR використовує неймерережеві алгоритми, щоб розпізнати цифри номеру контейнера та перетворити фрагмент у текстовий формат. Після цього результат розпізнавання номеру контейнера передається до програмного забезпечення автоматизованого складу.

Другий метод на основі Tesseract (двигун розроблений Google, що може бути використаний на

3 (рис. 4) та лінії розмітки $A-a$ (рис.1). Ця умова укавана в команді у вигляді $R=f$.

Методи ідентифікації номеру контейнера.

Позиціонування МР на вантажному місті секції виконується за розміткою на підлозі. Номер місця вказан кодом розмітки пілоги, який зчитується матрицею датчиків відображення. Ідентифікація номеру контейнера розбита на дві фази. Перша - позиціонування поворотної платформи на якій встановлена веб камера на контейнер та зчитування зображення. Друга фаза вже є безпосередньо фаза обробки зображення. Це локалізація контейнера на зображенні, локалізація області на контейнері та розпізнавання номеру в локалізованій області.

Були проаналізовані три підходи обробки даних від камери, під'єднаної до мікрокомп'ютера Raspberry Pi 3B+, для вирішення завдання розпізнавання номеру контейнеру, що зображено на рис. 5.

Raspberry Pi для розпізнавання тексту, включаючи номери контейнерів). Можна використовувати різні алгоритми машинного навчання, такі як згорткові або рекурентні нейронні мережі для навчання моделей Raspberry Pi розпізнаванню номерів контейнерів на зображеннях.

Розглянута наступна схема обробки даних. Препроцесинг зображень (операції попередньої обробки зображень, такі як бінаризація, видалення шуму, розпізнавання ліній тощо). Детекція тексту (виявлення регіону з текстом на зображенні, визначення координат розташування текстових блоків, рядків та окремих символів). Розпізнавання тексту (рекурентні нейронні мережі для розпізнавання тексту шляхом аналізу текстових блоків, рядків та символів, що дозволяє розпізнавати текст з варіаціями шрифту, розміру, кутів та інших атрибутів).

Третій метод на основі Tesseract TensorFlow та Keras. Ці популярні бібліотеки машинного навчання можуть бути використані для розпізнавання номерів контейнерів на зображеннях. Вони забезпечують широкий спектр можливостей для розробки та тренування власних моделей розпізнавання номерів контейнерів, включаючи використання камери Raspberry Pi як джерела зображень [14].

Висновки

Для дослідження можливостей нового напрямку штучного інтелекту, а саме штучний інтелект що відчуває, призначеного для підтримки автономності безлюдних систем, створюється штучне оточення. АІБС взаємодіє зі своїм оточенням на базовому рівні

шляхом отримання даних від сенсорів про стан оточення й впливу на стан оточення через свої актуатори. Доведено що архітектура базового рівня штучного оточення має бути універсальною щодо підтримки різних методів й алгоритмів управління на базі різноманітних даних від сенсорів. Універсальність досягається завдяки багатопаровій організації контролерів. На прикладі штучного оточення «Склад контейнерів», як прототипу АІБС показно набір контролерів, приклад схем підключення сенсорів та

актуаторів, методи та алгоритми управління, які потрібні на базовому рівні щоб підтримувати функції ШІВ. Експерименти зі контролерами базового рівня штучного оточення «Склад контейнерів» продемонстрували можливість підвищення рівня автономності безлюдної системи за рахунок розширених можливостей використання в управляючій програмі робота різноманітних методів управління на різних етапах плану із урахуванням поточної ситуації, що оцінюється на множині даних від різноманітних сенсорів.

СПИСОК ЛІТЕРАТУРИ

1. Функціональне призначення та класифікація складів в логістиці // WareTeKa News, 13.11.2020, <https://wareteka.com.ua/uk/blog/klassifikaciya-skladiv-v-logistyki/>
2. Rasmussen S, Kingston D, Humphrey L (2018) Brief Introduction to Unmanned Systems Autonomy Services (UxAS). Int. Conf. on Unmanned Aircraft Sys. (ICUAS). <https://doi.org/10.1109/ICUAS.2018.8453287>
3. Dimitrakos TD, Kyriakidis EG (2008), "A semi-Markov decision algorithm for the maintenance of a production system with buffer capacity and continuous repair times", International Journal of Production Economics, Vol. 111(2), pp. 752-762, doi: <https://doi.org/10.1016/j.ijpe.2007.03.010>
4. T. A. Litman, "Autonomous Vehicle Implementation Predictions: Implications for Transport Planning," Victoria Transport Policy Inst., Rep., Aug. 2022.
5. H. Shakhathreh et al., "Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges," J. IEEE Access, vol. 7, pp. 48572–48634, Apr. 2019, doi: 10.1109/ACCESS.2019.2909530.
6. T. Zhang et al., "Current trends in the development of intelligent unmanned autonomous systems," Frontiers Inf. Technol. Electron. Eng., vol. 18, Feb. 2017, pp. 68–85, doi: 10.1631/FITEE.1601650.
7. J. Reis, Y. Cohen, N. Melao, J. Costa, and D. Jorge, "High-Tech Defense Industries: Developing Autonomous Intelligent Systems," Appl. Sci., vol. 11, 4920, 2021, doi: 10.3390/app11114920.
8. M.Huang and R.Rust, "Artificial Intelligence in Service," J. of Service Res., vol. 21(2), Feb. 2018, pp.155-172, doi: 10.1177/1094670517752459.
9. M. Czerwinski, J. Hernandez, D. McDuff, "Building an AI that feels," Appl. Sci., vol.11, 4920, Apr. 2021, doi: 10.3390/app11114920.
10. A. Kargin, T. Petrenko, "Feeling Artificial Intelligence for AI-Enabled Autonomous Systems" in Conf. Proc. of 2022 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT) Alamein New City, Egypt, 18-21 December 2022, P.88-93.
11. A. Kargin and T. Petrenko, "Spatio-Temporal Data Interpretation Based on Perceptual Model," in Advances in Spatio-Temporal Segmentation of Visual Data. Studies in Computational Intelligence, V. Mashtalir, I. Ruban, V. Levashenko, Eds., vol. 876, Springer, Cham, 2020, pp. 101-159.
12. A. Kargin and T. Petrenko, "Multi-level Computing With Words Model to Autonomous Systems Control," in Proc. 9th Int. Conf. Inf. Control Sys.&Tech (ICST-2020), A. Pakštas, T. Hovorushchenko, H. Yin, N. Rudnichenko. Eds. Odessa, Ukraine, Sep. 24–26, 2020, CEUR Workshop Proceedings, vol. 2711, pp. 16-30. [Online]. Available: <http://ceur-ws.org/Vol-2711/>
13. Келер А., Брэдски Г. Изучаем OpenCV 3: навч. посіб. ДМК Пресс, 2017. 826 с.
14. Keras. Simple. Flexible. Powerful. [Online]. Available: <https://keras.io/>

Received (Надійшла) 15.03.2023

Accepted for publication (Прийнята до друку) 17.05.2023

A basic-level model of the artificial environment of autonomous intelligent unmanned systems as an example of a mobile robot that serves

Anatolii Kargin, Stanislav Zhukov, Danilo Sergeev, Yevhenii Silin

Abstract. Motivation. Despite significant progress in the field of creating unmanned systems, ensuring the necessary level of their autonomy remains an topicality task. Artificial intelligence plays an important role in its solution. Features of unmanned systems gave rise to a new model of Feeling Artificial Intelligence (FAI) that supports autonomy. **The goal of this work** is to create an artificial environment model for experiments with unmanned systems supported by FAI using the example of a container warehouse serviced by a wheeled robot. **The object of research** is the methods and models of controlling autonomous mobile robots based on data from various sensors. **Results.** The architecture of the artificial environment, the basic components of the system with a multi-layered organization are described. Using the example of the artificial environment "Container Warehouse" as a prototype of an autonomous unmanned system serviced by a wheeled robot, a set of controllers, an example of sensor and actuator connection schemes, control methods and algorithms that are required at the basic level are shown. The justification of the method of identification of container numbers, which satisfies the requirements imposed by autonomous systems, is presented. **Conclusions.** It is proven that the architecture of the basic level of the artificial environment should be universal in terms of supporting various control methods and algorithms based on various data from sensors. Versatility is achieved due to the multi-layered organization of controllers to support the functions of FAI. Experiments with controllers of the basic level of the artificial environment "Container Warehouse" demonstrated the possibility of increasing the level of autonomy of the unmanned system due to the expanded possibilities of using various control methods in the robot's control program at different stages of the plan, taking into account the current situation, which is evaluated on the basis of a set of data from various sensors.

Keywords: autonomous intelligent unmanned systems, artificial environment, programming control, robot, sensor, recognition of container numbers.