

УДК 004.81:004.383.8

КАРГІН А. О., д-р техн. наук, професор (Український державний університет залізничного транспорту),

ПЕТРЕНКО Т. Г., канд. техн. наук, доцент (Український державний університет залізничного транспорту),

ГІЄВСЬКИЙ Д. О., аспірант (Український державний університет залізничного транспорту)

Автономні інтелектуальні безлюдні системи: методи та моделі управління, що задовольняють критерій автономності

При створенні нових поколінь безлюдних систем (БС) основна увага приділяється проблемі підвищення їхнього рівня автономності. Застосування штучного інтелекту (ШІ) для завдань управління БС у невпорядкованому оточенні є одним із напрямів підвищення їхнього рівня автономності. Сьогодні є актуальним створення третього покоління ШІ для врахування особливостей завдань управління в автономних інтелектуальних БС (АІБС). Управління в АІБС є процесом підтримки реалізації плану дій у реальному часі, на кожному етапі плану приймається рішення з урахуванням поточного стану виконання плану, поточної ситуації та можливості виконати частину плану дій, що залишилася, для досягнення мети. У статті розглянуто методи програмного управління та методи управління на основі знань у вигляді правил. Наведено структуру багатошарової розподіленої нечіткої логічної системи (НЛС), що включає множини незалежних локальних НЛС з традиційними лінгвістичними змінними (ЛЗ) і факти з контекстної пам'яті. Розглянуто модифікований механізм нечіткого виведення, який завдяки введенню нечіткого фактора впевненості отримав можливість уніфіковано обробляти як ЛЗ, так і факти. Наведено приклад застосування НЛС в АІБС і результати комп'ютерних експериментів з колісним роботом.

Ключові слова: штучний інтелект, автономна інтелектуальна безлюдна система, нечітка логічна система, управління планом реалізації дій.

Вступ.

При створенні нових поколінь безпілотних автомобілів [1], автономних апаратів військового призначення [2] та інших автономних систем основна увага приділяється проблемі підвищення їхнього рівня автономності [3]. Індустріальні БС компонується з автоматичних ліній і верстатів з числовим програмним управлінням, транспортних систем і роботів [4]. У залізничній галузі створюються БС на базі розумних поїздів [5]. Сьогодні успіхи ШІ зміцнюють впевненість розробників та інвесторів БС вирішити зазначене завдання в найкоротші терміни. Розрізняють три класи БС залежно від рівня автономності: від запрограмованих автоматичних БС та інтелектуальних БС до автономних інтелектуальних БС (АІБС) [4, 6, 7]. Перший тип має обмеження. БС можуть працювати лише як попередньо запрограмовані і не можуть адаптуватися до будь-яких змін навколишнього середовища.

Другий тип має певні можливості сприйняття, прийняття рішень і контролю, а також може налаштовуватися відповідно до змін навколишнього середовища. Останній тип БС має високий рівень

автономності та інтелекту, не поступаючись або навіть перевершуючи людей у багатьох завданнях [8].

Отже, ШІ є важливою складовою БС, що забезпечує автономність. Рівень автономності, забезпечений апаратними ресурсами БС, задає рамки можливостей ШІ для підтримки автономності, і навпаки, від рівня інтелектуальності, забезпеченого технологіями ШІ, залежить рівень автономності прийняття рішень БС. При розробленні інтелектуальної компоненти традиційних інтернет-речей не акцентується увага на наявності цієї залежності. Однак при створенні АІБС це потрібно робити. З урахуванням зазначеної залежності в роботах [7, 9] вводяться три типи моделей штучного інтелекту - механічний, мислячий і ШІ, 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. Прикладом цивільного застосування механічного інтелекту є використання сервісних роботів присутності для отримання в режимі «Face to face» консультацій, придбання білетів чи заселення в готелі.

Вони замінюють людину в рутинних і стандартизованих завданнях [10].

Прикладом мислячого штучного інтелекту є чотириногі роботи Boston Dynamics, які добре адаптуються, універсальні та здатні вловлювати особливості живих істот. Щодо ШІ, що відчуває, то зараз у науковому середовищі обговорюються різні аспекти від соціальних наслідків до проектів архітектури [8, 11].

Критерії автономності мають різне тлумачення залежно від сфери застосування АІБС і часто даються на якісному рівні. Для автономних автомобілів розроблено стандарт, за яким існує шість рівнів автономності [12]. У роботах з автоматизації виробництва пропонується інший погляд на рівні автономності, як і у сфері створення систем інтернету речей обговорюються свої визначення [13, 14]. У цій роботі межі автономності АІБС розглядаються в загальному сенсі, як ступені неупорядкованості предметної сфери, при яких АІБС здатна виконувати свою місію без будь-якого втручання людини.

Постановка проблеми.

Спочатку розглянемо причини, які порушують автономність управління в АІБС. Виділяють три основні причини [15]: 1) розмірність задачі прийняття рішень. АІБС обробляє мультимодальні дані від великої кількості сенсорів та інших джерел; 2) чіткі межі в просторі прийняття керуючих дій. При чітких межах через неминучі шуми у вхідних даних у прикордонних областях зберігається можливість помилкових дій; 3) особливості процесу управління в БС. Управління в АІБС є процесом підтримки реалізації плану дій, тобто це багатоетапний процес у реальному часі. На кожному етапі приймається рішення з урахуванням поточного стану виконання плану, поточної ситуації та можливості виконати частину плану дій, що залишилася, для досягнення мети.

Усунення вказаних причин дасть змогу поліпшити показник автономності АІБС, тому дослідження моделей, які можуть бути використані для управління реалізаціями плану дій АІБС і задовольняють вимоги щодо рівня автономності прийняття управляючих рішень у неупорядкованому середовищі, є актуальним.

Традиційно при автоматизації управління послідовністю дій обладнання, у тому числі в робототехніці, використовуються моделі програмного управління [16, 17]. Програмне управління з прямими та зворотними зв'язками передбачає жорстке упорядкування середовища, і будь-які непередбачувані зміни викликають аварійні зупинки обладнання. Це є суттєвим обмеженням використання моделей у БС, з точки зору забезпечення автономності управління.

Одним із напрямів подолання зазначеної проблеми є використання ШІ, що породило новий клас БС, відомий як АІБС, який розглянуто в цій роботі. Сьогодні в наукових публікаціях обговорюється широкий спектр моделей ШІ і підходів до інтелектуалізації БС. Актуальний на сьогодні арсенал підходів і моделей ШІ, які можуть бути адаптовані для вирішення завдання, що розглядається, поділений на два напрями: 1) орієнтований на обробку даних від датчиків; 2) орієнтований на обробку знань. Перший включає підходи злиття даних і показань сенсорів: інтелектуальний аналіз даних, вилучення знань із потоків даних, агрегація різнорідних даних [18, 19] і дистиляція знань [15, 20]. Ці моделі можуть бути використані на етапах первинної обробки даних, але для вирішення завдання безпосереднього прийняття рішень та управління вони не призначені. З цієї ж причини важко використовувати «чисті» штучні нейронні мережі для додатків, до яких належить АІБС [21]. Другий підхід, відомий у ШІ як знання-орієнтовані системи, засновані на правилах [21], дає змогу реалізовувати завдання прийняття рішень, урахувавши більшість вищевказаних особливостей АІБС. Прийняття рішень у галузях робототехніки, інтернету речей, розумних машин виконує Rules Engine [22]. Він широко використовується у вбудованих системах реального часу, однак проблема узагальнення даних, що підтримується нечіткими логічними системами (НЛС), залишається актуальною. Використання НЛС у «чистому» вигляді є проблематичним через обмежені можливості налаштування нечіткої системи (кількість вхідних змінних не перевищує п'ять-сім) і масштабованість (додавання нових вхідних числових змінних або зміна їхньої кількості чи визначення термінів лінгвістичних змінних потребує зміни існуючої бази знань [23]). Підхід «розділяй і володарюй», що добре зарекомендував себе в перших експертних системах на базі правила продукції [24] щодо зменшення розмірності простору рішень, застосовується і в НЛС [23] для завдань управління. Однак для вищезгаданої моделі управління, що підтримує реалізацію плану дій, застосування підходу «розділяй і володарюй» не відомо.

Мета цієї статті полягає в дослідженні можливостей НЛС для управління планом реалізації дій АІБС завдяки суттєвому зменшенню розмірності завдань прийняття рішень при застосуванні підходу «розділяй і володарюй» і демонстрації можливості такої моделі управління в підтримці автономності АІБС.

Задача управління АІБС. Нехай місія АІБС (наприклад колісного робота) полягає в безперервному переміщенні вантажу з позиції A на позицію B (рис. 1). План реалізації місії поданий як послідовність станів системи «Оточення-АІБС», які АІБС завдяки своїм діям (актуаторам) має послідовно проходити. Для прикладу план реалізації місії можна подати у вигляді замкнутої послідовності станів «Завантажена АІБС на

позиції В», «Переміщення АІБС на позицію 4», «Переміщення АІБС на позицію А», «Розвантажена АІБС на позиції А», «Переміщення АІБС на позицію 3», «Переміщення АІБС на позицію В» і знову на початок послідовності, тобто стан «Завантажена АІБС на позиції В» (рис. 1).

$$\rightarrow B^{Load} \rightarrow 4^{Move} \rightarrow 1^{Move} \rightarrow A^{Move} \rightarrow A^{UnLoad} \rightarrow 3^{Move} \rightarrow 5^{Move} \rightarrow B^{Move} \rightarrow . \quad (1)$$

Окремі етапи плану подані як локальні цілі АІБС. Досягнення кожної локальної цілі у свою чергу

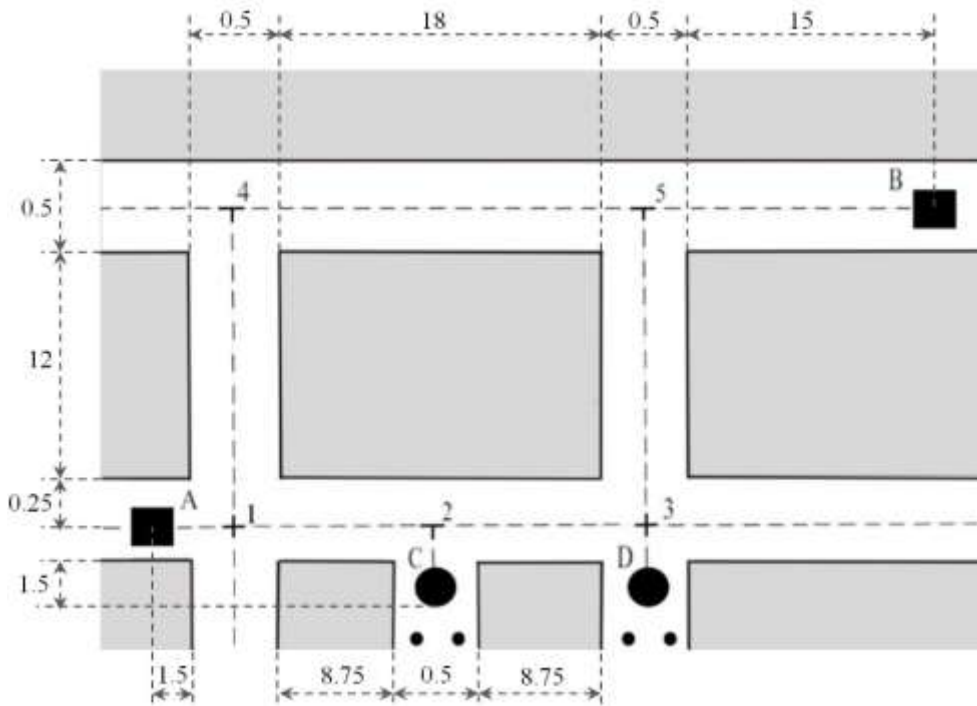


Рис. 1. Модель оточення АІБС

Головна особливість розглянутого в роботі завдання управління – це управління реалізацією плану дій за виразом (1), управління виконанням послідовності дій у певних умовах, що призводять до досягнення цілі. Як згадувалося вище, відомі такі методи управління реалізацією плану дій:

1. Методи програмного управління, використовувані для управління БС перших двох поколінь, – це програмне управління без зворотних зв'язків із ними. До цієї групи методів можна віднести і більшість інтелектуальних алгоритмів управління, розроблених спеціально для реалізації жорсткої послідовності дій у вигляді спеціалізованої програми управління з використанням моделей штучного інтелекту.

2. Методи управління на основі знань (системи, засновані на правилах).

3. Методи і моделі цілеспрямованого управління.

Архітектура пропонованої системи управління являє собою багатопланову структуру. До першого, нижнього, шару належать традиційні

можливо за більш детальним планом дій. Місія реалізується за умов подолання перешкод. Розглядається кілька типів перешкод, наприклад низький заряд акумулятора, що потребує заїзду на станцію заряджання, об'єкт-перешкода на шляху руху АІБС чи руйнування оточення – розмітка на підлозі витерлася або відсутність бічної огорожі. Перешкоди потребують переривання плану реалізації місії, переходу до нового плану усунення перешкоди і продовження реалізації місії.

регулятори, що підтримують задані значення параметрів [17, 21], наприклад такі, як відхилення від лінії при русі робота вздовж розмітки на підлозі та інші. До другого шару належать контролери, що реалізують фрагменти програмного управління окремими послідовностями дій, які не схильні до різного роду збурень, наприклад поворот робота на заданий кут ліворуч чи праворуч, поворот поворотної платформи, завантаження чи розвантаження контейнера, рух вперед до заданого маркера чи відстань до перешкоди. На третьому шарі знаходиться множина локальних НЛС, які містять безпосередньо знання щодо реалізації управління АІБС шляхом активізації певних контролерів першого чи другого рівнів з наданням їм певних значень параметрів. Ці локальні бази знань включають уже невелику кількість традиційних нечітких правил з лінгвістичними змінними, що згруповані окремо для кожного етапу плану й використовуються традиційним механізмом нечіткого виведення. На четвертому шарі знаходиться множина глобальних знань (для кожного плану дій є своя база нечітких правил), які теж обробляються традиційним механізмом нечіткого виведення. І,

нарешті, на п'ятому шарі знаходиться нечітка база знань у вигляді нечітких структурованих правил, що реалізує цілеспрямоване управління АІБС, включаючи перемикання з одного плану дій до іншого залежно від ситуації й повернення до перерваного плану дій, коли збурення усунуто. Відповідно до методу, який реалізує управління, кожний шар отримав свою назву: шар регуляторів, шар програмного управління, шар нечіткого управління етапами плану, шар управління реалізацією плану та шар цілеспрямованого управління. У роботі розглянуто моделі, застосовувані на нижніх чотирьох шарах.

Контролери шару регуляторів. До першого, нижнього, шару належать традиційні регулятори, що підтримують задані значення параметрів. Система, що розглядається, включає такі традиційні для класичного управління регулятори: руху робота вздовж розмітки на підлозі, що корегує відхилення робота від лінії розмітки на підлозі (*moveR*) для визначеного параметра похибки управління; руху робота вздовж штучної огорожі на заданій відстані від неї (*moveL*); швидкості руху (*moveV*); позиціонування робота за маркером (*moveM*) (керований параметр - це відстань між заданими точками на зображенні маркера і карті оточення АІБС, що побудована на основі різного типу сенсорів, таких як відео, ультразвукового лідача та інших); зупинки за маркером (*stopM*); зупинки за відстанню (*stopS*); безумовної зупинки (*stop*); повороту на заданий кут (*leftΨ*, *rightΨ*). У табл. 1 наведено приклади застосування ідентифікаторів цих контролерів у командах, що вказані в правилах верхніх шарів, на прикладі оточення АІБС (рис. 1).

Т

Таблиця 1

Приклади вказівки на команди контролерів

Шар регулятора	Шар програмного управління. Жорстке програмне управління	Шар програмного управління. Гнучке програмне управління
1	2	3
<i>moveM</i> $M=M^+$	<i>moveM</i>	<i>moveM</i>
<i>moveV</i> $V=12$	$delay\ t_{B-4} = 10.25$	<i>stopS</i> 33.75
<i>leftΨ</i> $\gamma=90^\circ$	<i>move_off_M</i>	<i>leftΨ</i> $\gamma=90^\circ$
<i>moveM</i> $M=M^+$	$delay\ t_{90^\circ} = 0.5\ c$	<i>moveM</i>
<i>rightΨ</i>	<i>leftΨ_off</i>	<i>stopS</i> 12.5
		<i>rightΨ</i> $\gamma=90^\circ$
		<i>moveM</i>

$\gamma=90^\circ$	<i>moveM</i>	<i>stopS 1.5</i>
<i>stopS</i> $S=12.5$	<i>delay t_{4→1} = 3.75 c</i>	
	<i>move_off_M</i>	
	<i>rightΨ</i>	
	<i>delay t_{90°} = 0.5 c</i>	
	<i>rightΨ_off</i>	
	<i>moveM</i>	
	<i>delay t_{1→A} = 0.45 c</i>	
	<i>move_off_M</i>	

Моделі й методи шару програмного управління. У методах програмного управління план від вихідної позиції, наприклад з місцезнаходження робота до цільової, поділяється на окремі етапи плану (кадри), послідовність яких є управляючою програмою (УП) АІБС. Контролер програмного управління активізує кадри управляючої програми шляхом активізації вказаної команди, контролює завершення виконання цього етапу (успішне чи аварійне) і приймає рішення про активізацію наступного етапу. В УП, що реалізує жорстке програмне управління, умовою завершення етапу є час (у кадрі УП задається інтервал часу). Далі методи управління демонструються на прикладі АІБС з обмеженим набором команд управління, а саме мобільним роботом з набором команд *вперед, праворуч, ліворуч*. У методі жорсткого програмного управління видача команди викликає початок відповідної дії (руху робота). Ця команда не змінюється, і відповідно реалізація команди триває до тих пір, поки не буде видана наступна команда, що вимикає (деактивізує) попередньо увімкнуту дію. Метод жорсткого програмного управління передбачає, що реалізація команди, наприклад переміщення, здійснюється з постійними характеристиками (швидкість руху) і триває певний час.

Розглянемо на прикладі оточення колісного робота, що рухається маршрутом $B^{Load} \rightarrow 4^{Move} \rightarrow 1^{Move} \rightarrow A^{Move} \rightarrow A^{Unload}$ (рис. 1) зі швидкістю $V = 3 \text{ м/с}$. При точно заданих відстанях між ключовими точками маршруту розраховується потрібний час для досягнення цих точок, наприклад $S_{B \rightarrow 4} = 33.75 \text{ м}$, $t_{B \rightarrow 4} = 10.25 \text{ с}$. Це значення вказується в команді затримки, що впливає з команди *moveM*. У табл. 1 у стовпці 2 (жорстке програмне управління) наведено УП, що реалізує фрагмент плану за виразом (1). З прикладу видно, що до складу команд методу жорсткого програмного управління обов'язково входить команда затримки (*delay*) з параметром часу. Контролер, що інтерпретує УП, при обробці цієї

команди затримує на t одиниць часу початок виконання наступної команди. Інтерпретація УП є жорсткою послідовністю виконання команд: i -та за порядком команда в УП не може почати виконуватися, доки не буде завершено виконання $i-1$ -ї команди.

Метод жорсткого програмного управління вимогливий до попереднього впорядкування середовища: обов'язкові вимоги щодо жорсткого дотримання розрахованої швидкості переміщення й миттєві режими розгону/гальмування робота. Ці вимоги для колісних роботів практично ніколи неможливо виконати. Навіть при точних розрахунках часу для точно вимірянних відстаней і заздалегідь встановлених у реальному світі робот дещо відхиляється від розрахованого часу, зокрема через часові відхилення на розгін і гальмування. Ці дрібні відхилення накопичуються, що врешті-решт призводить до аварії. Це є головним обмеженням застосування методу жорсткого програмного управління в АІБС, де необхідно підтримувати автономність в умовах різноманітних збурень.

У методі програмного управління зі зворотними зв'язками (гнучке програмне управління) для ідентифікації подій, наприклад виходу робота на певні позиції, використовуються дані від датчиків, а не інтервали часу. У табл. 1 у третьому стовпці наведено УП, у якій у командах використовуються дані від двох датчиків. Реалізацію команди *moveM* в УП підтримує контролер управління рухом вздовж розмітки на підлозі, що використовує дані від датчиків відображення. Умова вимикання команди переміщення під управлінням контролера задається конкретним значенням пройденої відстані, наприклад $S_{B \rightarrow 4} = 33.75$ у команді *stopS 33.75* (табл. 1). Контролер програмного управління звертається до контролера першого шару *stopS*, який, отримуючи в реальному часі дані від датчика одометра, розраховує поточну відстань і видає сигнал вимкнути контролер

руху $moveM$, коли значення відстані дорівнює зазначеному чи більше нього.

Методи гнучкого програмного управління мають переваги порівняно з методами жорсткого програмного управління. По-перше, завдяки використанню даних від датчиків розширюються можливості подолання перешкод, оскільки етапи плану завершуються на підставі фактичних даних про стан оточення, а не на очікуваних попередньо розрахованих даних про час виконання команди. По-друге, на підставі методу програмного управління зі зворотними зв'язками створюються різні варіанти управління реалізацією планів, наприклад ситуаційне управління, де є можливість задавати і контролювати будь-якої складності ситуацію завершення виконання етапу. Крім того, в УП можуть бути етапи з різними контролерами управління, тобто за етапом жорсткого програмного управління може бути етап, що реалізує гнучке програмне управління, коли умовою завершення етапу є пройдена відстань, відстань до перешкоди чи наявність певного маркера, що ідентифікується камерою зору. У роботі розглянуто ще приклад використання матриці датчиків відображення, що дає змогу розпізнавати маркери на підлозі. Це контролер $moveR$, який для ідентифікації маркерів на підлозі має такі еталони: M^- – маркер звичайної розмітки, M^+ – маркер перехрестя, M^A – маркери позицій, M^0 – відсутність будь-якого маркера.

Незважаючи на перераховані переваги, методи гнучкого програмного управління, у тому числі ситуаційного, мають недоліки, що обмежують їхні можливості підтримки автономності АІБС при вирішенні завдань, що розглядаються у статті, а саме збурення спричиняють ситуації (стан навколишнього середовища), коли реалізація дії відповідно до етапу унеможливується. Дія є, а очікуваного результату немає. Наприклад, для досягнення мети потрібно, щоб АІБС перемістилася з позиції, де вона знаходиться, до позиції 5. У кадрі УП задана мета як умова завершення дії, а не вказані умови, потрібні, щоб активізація команди призвела до успішної реалізації переміщення. Тому успішна реалізація УП гнучкого й ситуаційного програмного управління залежить від упорядкування середовища, як і раніше для методу жорсткого програмного управління, тобто під конкретне упорядковане середовище потрібно створювати спеціальну УП. Порушення будь-якого впорядкування такого середовища призводить до неможливості закінчення виконання УП. Це суперечить принципам автономності й є обмеженнями для використання в «чистому» вигляді всіх розглянутих вище методів програмного управління в АІБС.

Шар нечіткого управління окремими етапами плану. Методи управління на основі систем, що ґрунтуються на правилах.

У робототехніці, інтернеті речей і розумних машинах для прийняття управляючих рішень у

реальному часі на основі різноманітних даних, у тому числі і від датчиків, використовують такі моделі й методи штучного інтелекту, засновані на правилах [25, 26]:

- 1) системи, засновані на правилах з символічним поданням знань і символічним виведенням (продукційні моделі);
- 2) системи, засновані на правилах з символічним поданням знань та імовірнісним виведенням (баєсові моделі);
- 3) системи, засновані на правилах з символічним поданням знань і виведенням на основі фактора впевненості;
- 4) системи, засновані на правилах з лінгвістичними змінними подання знань і нечітким виведенням.

Усі чотири перелічені вище системи, засновані на правилах, задовольняють вимоги щодо формування управляючого рішення з урахуванням певних умов, заданих у правилах.

ЯКЩО умова ТОДІ дія (p). (2)

Якщо стан системи «Оточення-АІБС» співпадає з умовою якогось правила, то реалізується дія, вказана в правилі. При обговоренні питання щодо можливості застосування тієї чи іншої моделі для управління в АІБС треба брати до уваги тип системи «Оточення-АІБС», оскільки механізми виведення цих моделей мають особливості. Так, перший тип є класичною системою з продукційними правилами і припускає замкнутість, повноту знань і відсутність будь-якої невизначеності. Тому ці моделі заздалегідь не розглядаються як ті, що можуть бути застосовані для забезпечення автономності АІБС. Дві наступні моделі – імовірнісний і виведення на основі фактора впевненості – ураховують неповноту знань, пов'язану з причинно-наслідковими зв'язками, і рішення формується тільки в умовах такої невизначеності. Однак їм притаманні два недоліки порівняно з нечіткими моделями. По-перше, вони не враховують нечіткість локалізації самих об'єктів і явищ, між якими існують вказані вище причинно-наслідкові зв'язки. У той же час НЛС ураховують обидва типи невизначеностей. Експертні оцінки у вигляді функцій належностей відображують нечіткість визначення умови та дії, а вага p правила (2) ураховує неповноту знань про причинно-наслідкові зв'язки між умовою та дією. Друга перевага НЛС – вони обробляють усі правила нечіткої бази знань паралельно й за один крок нечіткого виведення отримують рішення. У той же час імовірнісні та моделі на основі фактора впевненості потребують багатокрокового послідовного виведення проміжних фактів, щоб отримати рішення. Для систем реального часу це є суттєвим недоліком. Отже, НЛС повною мірою враховують різні причини невизначеності та є такими, що здатні в реальному часі виконувати нечітке логічне виведення з урахуванням динаміки змін вхідних даних.

Як було зазначено раніше, у «чистому» вигляді нечіткі системи не можуть бути застосовані в додатках, до яких належить АІБС. Вказані проблеми можуть бути подоланими завдяки особливостям процесу управління реалізацією плану в АІБС. Перша особливість полягає в тому, що не всі правила чи групи правил одночасно беруть участь у визначенні управління, а тільки ті, які пов'язані з реалізацією актуального етапу плану. Отже, проблему великої розмірності задачі прийняття рішень в АІБС можна подолати шляхом «розділяй» на окремі незалежні завдання управління окремими етапами плану і «володарюй» шляхом застосування глобальних знань. Для цього база знань структурується. Множина правил розбита на окремі локальні НЛС відповідно до етапів плану і шар нечіткого управління етапами плану містить множину цих НЛС. З кожним етапом плану може бути пов'язано кілька локальних НЛС, які відповідають за управління реалізацію плану при різних збуреннях. Ці локальні НЛС є незалежними одна від одної. Обробка правил з локальних НЛС потребує модифікації механізму нечіткого виведення, а саме на кожен момент часу механізм має «знати», чи є актуальною НЛС, яку він обробляє. Друга особливість пов'язана з тим, що реалізація дії є процесом, що триває певний час, поки не буде досягнуто локальної мети, а дія «запускається» активізованим правилом, коли виникає певна подія. У всіх інших випадках активізація цього правила неприпустима. Це ще зменшує кількість правил, які одночасно формують управління в певній ситуації, що поліпшує налаштування бази знань при її створенні. А для врахування цієї особливості при обробці правил потрібно, щоб була можливість тимчасово «приховувати» певні правила НЛС, а механізм мав можливість їх «бачити» тільки тоді, коли трапляються певні події, вказані в правилі. Третя особливість, яка допомагає подолати вказані проблеми, – це можливість відслідковувати контекст у процесі реалізації плану. Якщо ввести контекстні факти і для їхньої підтримки застосувати контекстну пам'ять, факти з якої є вхідними чи вихідними змінними локальних НЛС, то це суттєво скорочує кількість входів і виходів системи та захищає від використання при збуреннях помилкових вхідних даних НЛС порівняно з традиційним підходом, коли застосовуються входи з часовими затримками.

Усі три згадані особливості враховуються в моделі з правилами продукцій. Однак, якщо побудувати нечітку систему на концепції продукційних систем, у цьому випадку будуть втрачені переваги нечіткого логічного виведення традиційних систем, заснованих на лінгвістичних змінних. Тому в роботі розглядається варіант, коли нечітка система на базі лінгвістичних змінних модифікується так, щоб вона підтримувала функції згаданих продукційних систем і зберегла механізм нечіткого виведення на правилах з лінгвістичними змінними.

Структура складається з таких компонентів: системи сприйняття (СС), яка містить факти, що описують стан «Оточення-АІБС» на підставі даних від сенсорів, контекстної пам'яті (КП), яка містить факти, що характеризують стан виконання плану і локальних НЛС, частина яких належить до шару нечіткого управління окремими етапами плану, а інша частина до шару управління реалізацією плану. Кожна з НЛС зберігає модель традиційної НЛС: входами й виходами системи є чисельні змінні, на універсумі яких побудовані ЛЗ, які так само входять до нечітких правил. Частина цих входів і виходів є чисельними характеристиками фактів з СС, а частина з КП. У нечітких правилах передбачена можливість виділити ЛЗ, що належать КП (наявністю зірочки в імені ЛЗ), і лінгвістичні вирази, що описують події. Лінгвістичні вирази в правилах НЛС мають традиційну форму «*Ім'я_ЛЗ є терм_ЛЗ*», тільки як *Ім'я_ЛЗ* застосовуються нечіткі характеристики стану фактів. Факт може перебувати у збудженому, загальмованому чи невизначеному стані. Чисельною оцінкою стану факту є нечіткий фактор впевненості, введений у роботу [25] як нечітке LR число X з гаусовою L-R функцією належностей

$$X : \{x \mid m_X(x), \forall x \in [-q, +q], q \geq +1\}$$

$$m_X^L(x) = \exp(-(x-\alpha)^2 / 2 \cdot (v_L \cdot t_L)^2), \forall x \in [-1, \alpha]$$

$$m_X^R(x) = \exp(-(x-\alpha)^2 / 2 \cdot (v_R \cdot t_R)^2), \forall x \in (\alpha, +1]$$

з трьома параметрами: $(-1.0 \leq \alpha \leq +1.0)$ – впевненість; t_L – інтервал часу з моменту отримання даних; t_R – інтервал часу з моменту зміни даних. Передбачувана впевненість (ПВ) є чисельною оцінкою нечіткого фактора впевненості, яка враховує старіння даних, розраховується на підставі виразів (3) і (4):

$$cf = \alpha \cdot k_t,$$

де v_L, v_R – коефіцієнти швидкості старіння даних;

$$k_t = 1 - \frac{\sum_{\forall x \in [-1, \alpha]} m_X^L(x) + \sum_{\forall x \in (\alpha, +1]} m_X^R(x)}{\text{Card}([-1, +1]) - 1}$$

Старіння даних з часом призводить до того, що впевненість у тому, що факт знаходиться у збудженому $cf \approx +1$ чи загальмованому $cf \approx -1$ станах, прагне до нуля, що характеризує невизначеність стану (повна відсутність впевненості), тобто $cf \approx 0$. Для випадків, коли інтервали часу невеликі (чи з моменту отримання інформації, чи в спеціальних випадках з моменту змінення даних), упевненість не дуже змінюється порівняно з α , тобто $cf \approx \alpha$.

Нижче наведено приклади правил цього шару, у яких можна побачити всі перелічені вище особливості.

П₁: **ЯКЩО** CF_I^{Move} is high **ТА** $Event(CF_M^0$ is high) **ТА** $CF_D^R \& Close$ is high **ТА** CF_P is high
ТОДІ CF_MoveL is high, CF_MoveM is low, V is meddle,

П₂: **ЯКЩО** CF_I^{Move} is high **ТА** $Event(CF_M^-$ is high)
ТОДІ CF_MoveM is high, CF_MoveL is low, V is high,

П₃: **ЯКЩО** CF_I^{Move} is high **ТА** $Event(CF_M^+$ is high) **ТА** CF_M^l is high
ТОДІ CF_Stop is high, V is zero.

Кожне правило у виразах (6) належить до локальної НЛС, що управляє реалізацією етапу I^{Move} . В імені ЛЗ CF_I^{Move} , а саме частині, що вказує на ідентифікатор етапу I^{Move} , вказана зірочка, що є ознакою контекстного факту, відповідного сенсорному факту I^{Move} . Факт активізується при актуалізації відповідного етапу плану правилами верхнього шару. Правило П₁ буде спрацьовувати тільки один раз, коли станеться подія «порушено розмітку на підлозі», вздовж якої рухається робот до позиції l . На це вказує запис $Event(CF_M^0$ is high). Для всіх інших моментів часу це правило приховано від механізму виведення й не обробляється. Далі в правилі перевіряється наявність огорожі за показниками сенсора відстані $D^R \& Close$ is high і достатнього рівня заряду акумуляторної батареї CF_P is high. Коли зазначена ситуація має місце, правило активізує такі дії: по-перше, вимикає контролер управління рухом вздовж розмітки на підлозі CF_MoveM is low; по-друге, вмикає контролер управління рухом вздовж огорожі CF_MoveL is high; по-третє, перемикає швидкість руху на середню V is meddle. Це правило демонструє переваги гібридного застосування в полі ТОДІ двох типів нечітких компонент: ЛЗ, визначені на універсумі домену, – це швидкість руху, задана на відповідній шкалі, і факти, задані на універсумі ПВ. Правило П₂, як і П₁, буде спрацьовувати тільки один раз, коли станеться подія «з'явилася розмітка на підлозі» $Event(CF_M^-$ is high). Це правило повертає управління контролеру руху вздовж розмітки. І правило П₃, коли робот наїде на маркер перехрестя на підлозі (подія $Event(CF_M^+$ is high)) і локалізує маркер позиції l CF_M^l is high, зупиняє рух на перехресті для отримання інформації про безпеку перетину перехрестя.

Шар управління реалізацією плану.

Реалізація плану дій (1) потребує реалізації не будь-якої траєкторії досягнення кінцевої мети, а певної траєкторії, що проходить через задану послідовність станів системи «Оточення-АІБС». Нечіткі логічні системи належать до систем без пам'яті, тому на будь-якому кроці нечіткого виведення «не знають», які стани системи вже були пройдені і тим більше не знають, якого стану потрібно досягти на поточному кроці, щоб досягнути кінцевої мети. Для подолання цієї проблеми множину незалежних одне від одного нечітких правил БЗ структурують шляхом введення контекстної залежності [25]. Контекстна залежність, на відміну від жорсткої програмної послідовності етапів плану, встановлює послідовність цілей, указаних в етапах плану. При цьому зберігається незалежність одне від одного правил. Такий підхід потребує певної структуризації правил.

Розглянемо приклад структуризації нечітких правил для вирішення завдання, що розглядається в статті. Для цього скористуємося методикою [25]. Як вже було зазначено при обговоренні локальних НЛС, правила шару управління реалізацією плану використовують як сенсорні факти з СС, які подають поточну ситуацію, так і контекстні факти з КП. Нехай СС містить множину \mathbf{PS} фактів, поділену на дві підмножини \mathbf{PS}^{plan} і \mathbf{PS}^{sit} . Перша містить факти, що використовуються для визначення плану, а друга – факти, що описують ситуацію оточення АІБС (7).

$$\mathbf{PS} = \{ \mathbf{PS}^{plan} = \{ f_i, i=1,2,\dots,n \} = \{ A^{UnLoad}, B^{Load}, I^{Move}, 2^{Move}, 3^{Move}, 4^{Move}, 5^{Move}, A^{Move}, B^{Move} \}, \mathbf{PS}^{sit} = \{ g_j, j=1,2,\dots,m \} \}. \quad (7)$$

Для кожного факту у виразі (7) із \mathbf{PS}^{plan} для відслідковування стадії виконання плану й відображення її в поточний контекст вводиться ще один факт, а саме контекстний факт, який знаходиться в КП. Зазначимо контекстні факти, як і сенсорні факти f_i у виразі (7), тільки з верхнім індексом зірочка $*f_i$. Нижче наведено фрагмент КП, що містить множину \mathbf{CF} відповідних до виразу (7) контекстних фактів.

$$\mathbf{CF} = \{ *A^{UnLoad}, *B^{Load}, *I^{Move}, *2^{Move}, *3^{Move}, *4^{Move}, *5^{Move}, *A^{Move}, *B^{Move} \}.$$

Отже, на довільний момент часу КП є вектор \mathbf{CF} значень ПВ

$$\mathbf{CF} = (cf(f_i), cf(*f_i), i=1,2,\dots,n, cf(g_j), j=1,2,\dots,m). \quad (9)$$

Значення ПВ фактів, що належать фрагментам \mathbf{PS}^{plan} і \mathbf{PS}^{sit} , формуються на підставі даних від сенсорів АІБС, а ПВ контекстних фактів, що належать \mathbf{CF} , змінюється правилами НЛС шару управління реалізацією плану, що відслідковують контекст [27, 28].

Далі наведено приклад правила, що вказує, коли з'являться дані від сенсорів, які підтверджують, що

ціль, наприклад A^{UnLoad} , досягнуто (завершено виконання цього етапу плану), то змінюється актуальний контекст. Поточна ціль $*A^{UnLoad}$ деактуалізується (фактор впевненості цього факту змінюється до значення $CF_{*A^{UnLoad}} = -1.0$, а наступна за планом (вираз (1)) локальна ціль актуалізується $CF_{*I^{Move}} = +1.0$:

ЯКЩО $event(CF_{*A^{UnLoad}})$ is high **ТА** $CF_{*A^{UnLoad}}$ is high
ТОДІ $CF_{*A^{UnLoad}}$ is low, $CF_{**3^{Move}}$ is high. (10)

НЛС цього шару включає для кожного етапу плану правило типу виразу (10). Так, для плану виразу (1) потрібно вісім правил типу виразу (10). Для реальних завдань АІБС плани організовані ієрархічно: етапи плану верхнього рівня ієрархії складаються з підетапів більш детального плану дій. Таких рівнів плану дій може бути більше двох. Наприклад, етап 3^{Move} плану виразу (1) для реалізації потребує більш детального плану, який включає підетапи $1^{Move} \rightarrow 2^{Move} \rightarrow 3^{Move}$ (рис. 1). Для подання етапу плану, який складається з підетапів, потрібно вказати, що цей етап не є безпосередньо реалізацією дії АІБС, і його контекстний факт не вказує безпосередньо на факт сенсорних даних. Цей факт вказує на множину впорядкованих контекстних фактів. По суті це є узагальненим контекстом контекстних фактів. У виразі (10) такий факт-контекст-контекстів позначений двома зірочками $**3^{Move}$. Три правила для кожного підетапу етапу 3^{Move} , що реалізують план виразу (1):

П₁: **ЯКЩО** $event(CF_{**3^{Move}})$ is high
ТОДІ $CF_{*1^{Move}}$ is high,
П₂: **ЯКЩО** $CF_{**3^{Move}}$ is high **ТА** $CF_{*1^{Move}}$ is high **ТА** $event(CF_{1^{Move}})$ is high
ТОДІ $CF_{*1^{Move}}$ is low, $CF_{*2^{Move}}$ is high,
П₃: **ЯКЩО** $CF_{**3^{Move}}$ is high **ТА** $CF_{*2^{Move}}$ is high **ТА** $event(CF_{2^{Move}})$ is high
ТОДІ $CF_{*2^{Move}}$ is low, $CF_{**3^{Move}}$ is low,
 $CF_{*3^{Move}}$ is high. (11)

Правило (10) після активізації актуалізує всі три правила виразів (11). Це забезпечує присутність активного контекстного факту $**3^{Move}$ у всіх правилах і разом із цим активізує факт $*1^{Move}$ за допомогою правила П₁. Актуалізовані правила П₂ і П₃ активізуються, коли з'являється відповідна подія, а саме сенсорні дані активізують відповідні сенсорні факти 1^{Move} і 2^{Move} . Правило П₃ при активізації деактуалізує факт-контекст-контекстів $CF_{**3^{Move}}$ is low і активізує наступний етап нижнього рівня ієрархії, а саме факт-контекст нижнього рівня ієрархії $*3^{Move}$.

З кожним із етапів нижнього рівня плану, а точніше з локальною метою етапу, наприклад $*1^{Move}$, пов'язано декілька локальних НЛС, що належать до шару нечіткого управління окремими етапами плану. У цих НЛС наведено знання, які описують безпосередньо умови застосування дій, що ведуть до локальної мети, плани уникнення небезпечних ситуацій чи плани виходу з небажаних ситуацій, які непередбачено з'явилися.

Для проведення комп'ютерних експериментів була створена НЛС із чотирьох шарів, як зазначалося раніше. Перелік ЛЗ наведено в табл. 2. Комп'ютерні експерименти проводилися для трьох типів перешкод: пошкодження розмітки, низький заряд акумуляторної батареї, об'єкт перешкода на шляху робота. Кількість вхідних ЛЗ складає 33, з них 11 (12-22 у табл. 2) - це факти з КП (9) і шість вихідних ЛЗ.

Лінгвістичні змінні Фечіткої логічної системи
а управління АІБС
б

Номер з/п	Ім'я ЛЗ/сенс	Універсум	Термножина	Функція належностей
1-11	CF_{f_i} / факти плану	$CF: [-1, 1]$	<i>low, high</i>	$lz(-0.5, 0.25), ls(-0.25, 0.5)$
12-22	CF_{*f_i} / факти-супутники	$CF: [-1, 1]$	<i>low, high</i>	$lz(-0.5, 0.25), ls(-0.25, 0.5)$
23	CF_{M^+} / маркер перехрестя	$CF: [-1, 1]$	<i>low, high</i>	$lz(-0.5, 0.25), ls(-0.25, 0.5)$
24	CF_{M^C} / маркер зарядної позиції	$CF: [-1, 1]$	<i>low, high</i>	$lz(-0.5, 0.25), ls(-0.25, 0.5)$
25	CF_{M^A} / маркер позиції А	$CF: [-1, 1]$	<i>low, high</i>	$lz(-0.5, 0.25), ls(-0.25, 0.5)$
26	D^L / відстань до перешкоди ліворуч	$[0, \infty)$	<i>close, far</i>	$lz(0.3, 1), ls(0.7, 2)$
27	D^R / відстань до перешкоди праворуч	$[0, \infty)$	<i>close, far</i>	$lz(0.3, 1), ls(0.7, 2)$
28	D^A / відстань до перешкоди попереду	$[0, \infty)$	<i>close, far</i>	$lz(0.3, 1), ls(0.7, 2)$
29	P рівень заряду батареї	$[0, 100]$	<i>low, medium, high</i>	$lz(0.3, 1), trap(15, 45, 55, 70), ls(0.7, 2)$
30	$Charge$ / заряджання батареї здійснене	$[0, 100]$	<i>low, high</i>	$lz(50, 75), ls(75, 90)$
31	$Rotate_{Left}$ / поворот ліворуч здійснений	$[-180, 180]$	<i>low, high</i>	$lz(-90, -75), ls(-25, 0)$
32	$Rotate_{Right}$ / поворот праворуч здійснений	$[-180, 180]$	<i>low, high</i>	$lz(60, 75), ls(75, 90)$
33	$Rotate_{Around}$ / поворот назад здійснений	$[-180, 180]$	<i>low, high</i>	$lz(150, 170), ls(170, 180)$

Правила наведені в табл. 3, результати моделювання – табл. 4.

Таблиця 3

№	event	Inputs			Outputs			Move	Charge	Rotate
		f_i	\bar{f}_i	Marker	Detector	P	f_i			
1	$Rotate_{Left}$									Left: -1
2	$Rotate_{Right}$									Right: -1
3	$Rotate_{Around}$									Around: -1
4	B_{Load}		$\bar{B}_{Load}; high$							$B_{Load}; -1$ $5_{Move}; 1$
5			$5_{Move}; high$	$M^+; high$		high		$5_{Move}; 1$		$5_{Move}; -1$ $4_{Move}; 1$
6			$5_{Move}; high$	$M^+; low$	$D^+; far$	high		$5_{Move}; 1$		$5_{Move}; -1$ $4_{Move}; 1$
7			$5_{Move}; high$	$M^+; high$		medium OR low		$5_{Move}; 1$		$5_{Move}; -1$ $3_{Move}; 1$
8			$5_{Move}; high$	$M^+; low$	$D^+; far$	medium OR low		$5_{Move}; 1$		$5_{Move}; -1$ $4_{Move}; 1$
9			$4_{Move}; high$	$M^+; high$				$4_{Move}; 1$		$4_{Move}; -1$ $1_{Move}; 1$
10			$4_{Move}; high$	$M^+; low$	$D^+; far$			$4_{Move}; 1$		$4_{Move}; -1$ $1_{Move}; 1$
11		$4_{Move}; high$ $2_{Move}; low$	$1_{Move}; high$	$M^+; high$		medium OR high		$1_{Move}; 1$		$1_{Move}; -1$ $A_{Move}; 1$
12		$4_{Move}; high$ $2_{Move}; low$	$1_{Move}; high$	$M^+; low$	$D^+; far$ $D^+; far$	medium OR high		$1_{Move}; 1$		$1_{Move}; -1$ $A_{Move}; 1$
13		$4_{Move}; high$ $2_{Move}; low$	$1_{Move}; high$	$M^+; high$		low		$1_{Move}; 1$		$1_{Move}; -1$ $2_{Move}; 1$
14		$4_{Move}; high$ $2_{Move}; low$	$1_{Move}; high$	$M^+; low$	$D^+; far$ $D^+; far$	low		$1_{Move}; 1$		$1_{Move}; -1$ $2_{Move}; 1$
15		$2_{Move}; high$	$1_{Move}; high$	$M^+; high$				$1_{Move}; 1$		$1_{Move}; -1$ $A_{Move}; 1$
16		$2_{Move}; high$	$1_{Move}; high$	$M^+; low$	$D^+; far$ $D^+; far$			$1_{Move}; 1$		$1_{Move}; -1$ $A_{Move}; 1$
17		$5_{Move}; high$ $D_{Move}; low$	$3_{Move}; high$	$M^+; high$				$3_{Move}; 1$		$3_{Move}; -1$ $D_{Move}; 1$
18		$5_{Move}; high$ $D_{Move}; low$	$3_{Move}; high$	$M^+; low$	$D^+; far$ $D^+; far$			$3_{Move}; 1$		$3_{Move}; -1$ $D_{Move}; 1$
19		$D_{Move}; high$	$3_{Move}; high$	$M^+; high$				$3_{Move}; 1$ $D_{Move}; -1$		$3_{Move}; -1$ $2_{Move}; 1$

Продовження табл. 3

№	event	Inputs				Outputs				Rotate	
		f_i	\bar{f}_i	Marker	Detector	P	f_i	\bar{f}_i	Move		Charge
20		$D^{Move}; high$	$3^{Move}; high$	$M^+; low$	$D^R; far$ $D^L; far$		$3^{Move}; 1$ $D^{Move}; -1$	$3^{Move}; -1$ $2^{Move}; 1$	-1		<i>Left: 1</i>
21				$M^C; high$			$D^{Move}; 1$	$2^{Move}; -1$	-1	1	
22	Charge	$D^{Move}; high$				high		$3^{Move}; 1$		-1	<i>Around: -1</i>
23		$3^{Move}; high$	$2^{Move}; high$	$M^+; high$				$2^{Move}; -1$ $1^{Move}; 1$	1		
24		$3^{Move}; high$	$2^{Move}; high$	$M^+; low$	$D^L; far$			$2^{Move}; -1$ $1^{Move}; 1$	1		
25		$1^{Move}; high$	$2^{Move}; high$	$M^+; high$				$2^{Move}; -1$ $2^{Move}; 1$	-1		<i>Right: 1</i>
26		$1^{Move}; high$	$2^{Move}; high$	$M^+; low$	$D^R; far$			$2^{Move}; -1$ $2^{Move}; 1$	-1		<i>Right: 1</i>
27		$C^{Move}; high$	$2^{Move}; high$	$M^+; high$				$2^{Move}; -1$ $1^{Move}; 1$	-1		<i>Left: 1</i>
28		$C^{Move}; high$	$2^{Move}; high$	$M^+; low$	$D^L; far$			$2^{Move}; -1$ $1^{Move}; 1$	-1		<i>Left: 1</i>
29				$M^C; high$			$C^{Move}; 1$	$2^{Move}; -1$	-1	1	
30	Charge	$C^{Move}; high$				high		$2^{Move}; 1$		-1	<i>Around: -1</i>
31				$M^L; high$			$A^{Move}; 1$	$A^{Move}; -1$ $A^{Unload}; 1$	-1		
32	<i>Around</i>		$A^{Unload}; high$				$A^{Unload}; 1$	$A^{Unload}; -1$			

Таблиця 4

Результати моделювання

Ітерація	Точка	Тип даних	Дані
1	B	Вхідні дані	[0,0,-1,-1,-1,-1,-1,-1,1,-1,-1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.15,0.15,3,60]
		Подія	event(<i>B_{Load}</i>)
		Вихідні дані	[0,1,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,1,0,0,0,0,1,0,0,0,0]
2	5	Вхідні дані	[0,1,-1,-1,-1,-1,-1,-1,1,-1,-1,-1,1,-1,-1,-1,-1,-1,-1,-1,-1,3,0.15,3,60]
		Вихідні дані	[0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0.75,0.25,-1,0,0,0,0,-0.5,0,0,0,0]
3	5	Вхідні дані	[0,1,-1,-1,-1,-1,1,-1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,3,3,3,60]
		Подія	event(<i>Rotate_{Left}</i>)
		Вихідні дані	[0,1,0,-1,0,0]
4	3	Вхідні дані	[0,1,-1,-1,-1,-1,1,-1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,3,3,3,60]
		Вихідні дані	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,1,1,0,0,0,0]
5	D	Вхідні дані	[0,1,-1,-1,-1,-1,1,-1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0.15,0.15,3,60]
		Вихідні дані	[0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,-1,-1,1,0,0,0]
6	D	Вхідні дані	[0,1,-1,-1,-1,-1,1,-1,1,-1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,1,0.15,0.15,3,100]
		Подія	event(<i>Charge</i>)
		Вихідні дані	[0,-1,0,0,1]
7	D	Вхідні дані	[0,1,-1,-1,-1,-1,1,-1,1,-1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,1,0.15,0.15,3,100]
		Подія	event(<i>Rotate_{Around}</i>)
		Вихідні дані	[0,1,0,0,-1]
8	3	Вхідні дані	[0,1,-1,-1,-1,-1,1,-1,1,-1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,3,3,3,100]
		Вихідні дані	[0,0,0,0,1,0,0,0,0,0,-1,0,0,0,1,-1,0,0,0,0,0,0,-1,0,1,0,0]
9	3	Вхідні дані	[0,1,-1,-1,1,-1,1,-1,1,-1,-1,1,-1,1,-1,-1,-1,-1,-1,-1,-1,3,3,3,100]
		Подія	event(<i>Rotate_{Left}</i>)
		Вихідні дані	[0,1,0,-1,0,0]

10	2	Вхідні дані	[0,1,-1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,3,0.15,3,100]
		Вихідні дані	[0,0,0,1,0,0,0,0,0,0,0,0,0,1,-1,0,0,0,0,0,0,0,1,0,0,0,0]
11	1	Вхідні дані	[0,1,-1,1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,3,3,3,100]
		Вихідні дані	[0,0,1,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,1,0,0,0,1,0,0,0,0]
12	A	Вхідні дані	[0,1,1,1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,0.15,0.15,3,100]
		Вихідні дані	[0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,-1,0,0,0,-1,0,0,0,0]
13	A	Вхідні дані	[0,1,1,1,1,-1,1,1,1,-1,-1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,1,0.15,0.15,3,100]
		Подія	event(<i>A_{unload}</i>)
		Вихідні дані	[1,0,0,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

Висновки.

Підхід «розділяй і володарюй» до проєктування системи управління реалізацією плану дій АІБС в умовах невизначеності здатний подолати проблеми, що стоять на шляху застосування традиційних НЛС. Показано, що багатопшарова організація структури системи управління, яка містить традиційні регулятори

зі зворотними зв'язками на першому шарі, фрагменти управляючих програм методу програмного управління на другому шарі і розподілену нечітку систему на третьому і четвертому шарах, відкриває можливості підтримувати автономність управління в межах спроможності апаратного забезпечення АІБС. Модифікація традиційного механізму логічного виведення в НЛС, по-перше, шляхом введення контекстної пам'яті, що містить контекстні факти, по-друге, подання стану фактів, як і значень вхідних змінних НЛС у вигляді нечітких факторів впевненості, дає змогу застосувати НЛС для управління АІБС та використовувати всі переваги нечіткого управління щодо обробки невизначеності. Проєктування традиційним методом НЛС для наведеного в статті прикладу щодо управління планом реалізації автономного вантажного робота в умовах трьох типів збурень, що характеризується більш ніж 35 вхідними чисельними змінними від датчиків, є невіршеним завданням. У той же час, як показали комп'ютерні експерименти, це завдання стає вирішуваним при застосуванні запропонованого підходу з модернізованим механізмом нечіткого логічного виведення.

Надалі планується розроблення прототипу системи, що має багатопшарову структуру на базі мікроконтролерів із набором сенсорів.

Список використаних джерел

1. Joseph L., Mondal A. K. (eds). Autonomous Driving and Advanced Driver-Assistance Systems (ADAS). Applications, Development, Legal Issues, and Testing, 1st edn. CRC Press, Boca Raton, 2021. URL: <https://doi.org/10.1201/9781003048381>.
2. Lockheed M. The Future of Autonomy. Isn't Human-Less. It's Human More, 2022. Accessed 25 Sept 2022. URL: <https://www.lockheedmartin.com/en-us/capabilities/autonomous-unmanned-systems.html>.
3. The Future of Autonomy. Isn't Human-Less. It's Human More. Lockheed Martin, 2022. Accessed: August 25, 2022. URL: <https://www.lockheedmartin.com/en-us/capabilities/autonomous-unmanned-systems.html>.
4. Chen H. et al. From Automation System to Autonomous System: An Architecture Perspective. J. of Marine Sci. and Eng. 9(6), 2021. URL: <https://doi.org/10.3390/jmse9060645>.
5. Rail Technical Strategy Capability Delivery Plan, 2017. URL: <https://www.rssb.co.uk/rts/Documents/2017-01-27-rail-technical-strategy-capability-delivery-plan-brochure.pdf>.
6. Zhang T. et al. Current trends in the development of intelligent unmanned autonomous systems. Frontiers Inf. Technol. Electron. Eng. Vol. 18, Feb. 2017. P. 68–85. doi: 10.1631/FITEE.1601650.
7. Reis J., Cohen Y., Melao N., Costa J. and Jorge D. High-Tech Defense Industries: Developing Autonomous Intelligent Systems. Appl. Sci. Vol. 11, 4920, 2021. doi: 10.3390/app11114920.
8. Chen J., Sun J., Wang G. From Unmanned Systems to Autonomous Intelligent Systems.

- Engineering 12: 2022. P. 16-19. URL: <https://doi.org/10.1016/j.eng.2021.10.007>.
9. Czerwinski M., Hernandez J., McDuff D. Building an AI that feels. Appl. Sci. Vol.11, 4920, Apr. 2021. doi: 10.3390/app11114920.
 10. Kargin A., Petrenko T. Feeling Artificial Intelligence for AI-Enabled Autonomous Systems. Conference Proceedings 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. Huang M., Rust R. Artificial Intelligence in Service. J. of Service Res. 21(2), 2018. P. 155-172. URL: <https://doi.org/10.1177/1094670517752459>.
 12. Bachute M. R. and Subhedar J. M. Autonomous Driving Architectures: Insights of Machine Learning and Deep Learning Algorithms. Mach. Learn. with Appl. Vol. 6, Sep. 2021. doi: 10.1016/j.mlwa.2021.100164.
 13. Winfield A. et al. IEEE P7001: A Proposed Standard on Transparency. Front. Robot. AI. Vol. 8, Jul. 2021. doi: 10.3389/frobt.2021.665729.
 14. Huang H. The Autonomy Levels for Unmanned Systems (ALFUS) Framework: Interim Results, Proceedings of the Performance Metrics for Intelligent Systems (PerMIS). Workshop 2006, Gaithersburg, MD, Accessed: January 29, 2023. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=823586.
 15. Kargin A., Petrenko T. Knowledge Distillation for Autonomous Intelligent Unmanned System. In: Witold Pedrycz, Shyi-Ming Chen. Advancements in Knowledge Distillation: Towards New Horizons of Intelligent Systems. Studies in Computational Intelligence. Vol. 1100. Springer International Publishing, 2023. P. 193-231.
 16. Каргін А. О., Жуков С. В., Сергєєв Д. А., Сілін Є. Л. Модель базового рівня штучного оточення автономних інтелектуальних безлюдних систем на прикладі мобільного роботу що обслуговує. *Системи управління, навігації та зв'язку*. 2023. Вип. 2(72). С. 107-113.
 17. Michels K., Klawonn F., Kruse R., Nürnberger A. Fundamentals of Control Theory. In: Fuzzy Control. Studies in Fuzziness and Soft Computing. Vol. 200. Springer, Heidelberg, 2006. P. 57-234. URL: https://doi.org/10.1007/3-540-31766-x_2.
 18. Garcia G., Luengo J., Herrera F. Data preprocessing in data mining. Intelligent Systems Reference Library. Springer Cham, 2015. URL: <https://doi.org/10.1007/978-3-319-10247-4>.
 19. Klein L. A. Sensor and data fusion: A tool for information assessment and decision making, 2nd ed. Bellingham, Washington, USA: SPIE Press, 2012.
 20. West P. et al. Symbolic Knowledge Distillation: from General Language Models to Commonsense Models. 2022. URL: <https://doi.org/10.48550/arXiv.2110.07178>.
 21. Каргін А. А. Введение в интеллектуальные машины. Кн. 1. Интеллектуальные регуляторы. Донецк: Норд-Пресс, ДонНУ, 2010. 526 с.
 22. How To Choose a Rules Engine, Waylay Whitepaper, Accessed: Mar. 15, 2019. URL: https://static.waylay.io/white-papers/010_How-to-Choose-a-Rules-Engine.pdf?_s=7t06pcdhyc3621dgs391.
 23. Piegat A. Fuzzy modelling and control. Heidelberg: Physica-Verlag Heidelberg New York, 2001.
 24. Jackson P. Introduction to Expert Systems, 3rd edn. Addison-Wesley, Boston, 1998.
 25. Kargin A., Petrenko T. Spatio-Temporal Data Interpretation Based on Perceptual Model. In: Mashtalir V, Ruban I, Levashenko V (eds) Advances in Spatio-Temporal Segmentation of Visual Data. Studies in Computational Intelligence. Vol. 876. Springer, Cham. 2020. P. 101-159. URL: <https://doi.org/10.1007/978-3-030-35480-0>.
 26. Negnevitsky M. Artificial Intelligence: A Guide to Intelligent Systems. 2nd ed. Addison-Wesley, 2005. 415 p.
 27. Kargin A., Petrenko T. Method of Using Data from Intelligent Machine Short-Term Memory in Fuzzy Logic System. 2021 IEEE 7th World Forum on Internet of Things (WF-IoT). 2021. P. 842-847. URL: <https://doi.org/10.1109/wf-iot51360.2021.9594918>.
 28. Каргін А. А., Іванюк О. І. Модель ситуаційного планування й керування переміщеннями автономного роботу. *Сучасні інформаційні системи*. 2020. Т. 4, № 3. С. 41-51. doi: 10.20998/2522-9052.2020.3.05. URL: <http://ais.khpi.edu.ua/article/view/2522-9052.2020.3.05>.

Anatolii Kargin, Tetyana Petrenko, Dmytro Hiiievskiy. Autonomous intelligent unmanned systems: control methods and models satisfying the criterion of autonomy.

When creating new generations of unmanned systems (US), the main attention is paid to the problem of increasing their level of autonomy. The use of artificial intelligence (AI) for control tasks of US in a disordered environment is one of the ways to increase their level of autonomy. Today, the creation of the third generation of feeling AI is relevant to take into account the specifics of control tasks in autonomous intelligent US (AIUS). Control in AIUS is a process of supporting the implementation of the action plan in real time, at each stage of the plan, a decision is made taking into

account the current state of the plan's implementation, the current situation and the ability to implement the remaining part of the action plan to achieve the goal. The article examines programming control methods and knowledge-based control methods in the form of rules. The structure of a multi-layer distributed fuzzy logic system (FLS) is given, which includes a set of independent local FLSs with traditional linguistic variables (LVs) and facts from context memory. A modified fuzzy inference mechanism was considered, which, thanks to the introduction of a fuzzy certainty factor, was able to uniformly process both LV and facts. An example of the use of FLS in AIUS and the results of computer experiments with a wheeled robot are given.

Keywords. feeling artificial intelligence, autonomous intelligent unmanned system, fuzzy logic system, action plan management.

Каргін Анатолій Олексійович, доктор технічних наук, професор, завідувач кафедри інформаційних технологій, Український державний університет залізничного транспорту, Харків, Україна. E-mail: kargin@kart.edu.ua, ORCID ID <http://orcid.org/0000-0003-2885-9071>.

Петренко Тетяна Григорівна, кандидат технічних наук, доцент кафедри інформаційних технологій, Український державний університет залізничного транспорту, Харків, Україна. E-mail: petrenko_tg@kart.edu.ua, ORCID ID <http://orcid.org/0000-0001-6305-7918>.

Гієвський Дмитро Олегович, аспірант кафедри інформаційних технологій, Український державний університет залізничного транспорту, Харків, Україна. E-mail: gievskiy@kart.edu.ua, ORCID ID <https://orcid.org/0009-0006-0151-2317>.

Anatolii Kargin, Doctor of technical science, professor, department of information technology, Ukrainian State University of Railway Transport, Kharkiv, Ukraine. E-mail: kargin@kart.edu.ua, ORCID ID <http://orcid.org/0000-0003-2885-9071>.

Tetyana Petrenko, Ph.D., Associate Professor, department of information technology, Ukrainian State University of Railway Transport, Kharkiv, Ukraine. E-mail: petrenko_tg@kart.edu.ua, ORCID ID <http://orcid.org/0000-0001-6305-7918>.

Dmytro Hievskiy, PhD student, department of information technology, Ukrainian State University of Railway Transport, Kharkiv, Ukraine. E-mail: gievskiy@kart.edu.ua, ORCID ID <https://orcid.org/0009-0006-0151-2317>.