

А. КАРГІН, Р. КУЗЬМЕНКО

МОДЕЛЬ ЙМОВІРІСНО-НЕЧІТКОГО АКТУАТОРА В КОНТУРІ М'ЯКОГО УПРАВЛІННЯ АВТОНОМНОЮ БЕЗЛЮДНОЮ СИСТЕМОЮ

Предметом дослідження статті є методи й моделі штучного інтелекту, що застосовуються в управлінні автономними безлюдними системами. **Мета роботи** – створення нової моделі актуатора автономних безлюдних систем, що реалізує керівні рішення, прийняті штучним інтелектом в умовах невизначеності. У статті розв'язано такі **завдання**: запропоновано модель ймовірісно-нечіткого актуатора (ЙНА) та досліджено можливість її застосування як універсального контролера виконавчих механізмів (ВМ) в автономних системах. Модель ЙНА, що запозичена в біологічних актуаторів м'язів, формалізована у вигляді множини автоматоподобних елементів з ймовірнісним механізмом призначення їх вхідних змінних, розрахованих на підставі нечітких характеристик керівних рішень, отриманих від системи ШІ, що підтримує технологію м'якого управління. Упроваджено такі **методи**: нечітке управління, прийняття рішень в умовах невизначеності на основі фактора впевненості, теорія автоматів, теорія ймовірностей. **Досягнуті результати**: запропоновано й обґрунтовано модель ЙНА, запозичену у живих істот; розроблено алгоритм ЙНА, що реалізує керівні рішення, отримані методом м'якого управління. **Висновки**. Ймовірісно-нечіткий актуатор, на відміну від наявних методів реалізації керівних рішень у моделях м'якого управління, відкриває можливості реалізації команди, що в прийнятті рішень не мали абсолютної переваги з-поміж усіх потенційно можливих. Така здатність актуаторів, що застосовуються в автономних системах, корисна в умовах, коли система стикається з незнайомою ситуацією, оскільки всі наявні в її пам'яті прототипи реакції визначаються незначною впевненістю. У цих випадках для підтримання автономності важливо спробувати різні поведінки, а не тільки ту, що в рейтингу стоїть першою. Крім цього, метод "спроб і помилок" властивий для моделі самонавчання в автономних системах, яка спирається на нього. Комп'ютерні експерименти підтвердили можливість реалізації цього механізму за допомогою запропонованої моделі ЙНА.

Ключові слова: штучний інтелект; м'яке управління; автономна безлюдна система; нечіткі системи; ймовірнісні моделі.

Вступ

Сучасний етап визначається широким застосуванням різних автономних систем [1], зокрема роботів, автономних систем військового, промислового, побутового, аграрного [2] та освітнього призначення [3]. Автономні системи дають змогу звільнити людину від виконання складних завдань [2], замінити людину у виконанні завдань, пов'язаних зі шкідливим довкіллям [2]. Якщо ж завдання належать до військової галузі [4], то це допомагає зберегти здоров'я та життя. У сфері обслуговування автономні системи сприяють підвищенню ефективності, безпеки та якості обслуговування [4], зокрема у сфері освіти можуть покращувати навчальний процес [3, 5]. Автономні системи ефективні також у галузі залізничного транспорту [6]. Наведений перелік напрямів, де автономні системи застосовуються постійно, збільшується.

У створенні автономних безлюдних систем найбільш важливим компонентом є система прийняття рішень та управління. Нині застосовуються такі моделі систем управління: жорстке програмне

управління, гнучке програмне управління та гнучке ситуаційне управління, нечітке та нейронечітке управління. Жорстке програмне управління ефективно для типових стандартних завдань, які мають чітко визначені маршрути та сталі умови середовища [2]. Гнучке програмне управління розширює можливості автономного прийняття рішень, наприклад, робот здатний реалізувати свій маршрут з огляду на мінливі умови й перешкоди в довкіллі [7]. Гнучке ситуаційне управління розв'язує проблеми, що виникають унаслідок динамічних змін в оточенні завдяки зберіганню в пам'яті множини прототипів поведінки для різних класів ситуацій. Нечіткі та нейронечіткі моделі управління є ефективними в ситуаціях, коли автономна безлюдна система долає невизначеність і неповноту знань за допомогою узагальнення ситуацій [8]. Для перелічених моделей м'якого управління (ситуаційне, нечітке та нейронечітке) забезпечення високого рівня автономності потребує наявності попередніх прототипів усіх можливих ситуацій. Це обмежує застосування цих методів в автономних системах, що працюють в умовах високого рівня невизначеності, зумовленої динамікою навколишніх процесів [9].

Аналіз останніх досліджень і публікацій

Особливістю завдань автономних роботизованих систем є формування керівних рішень в умовах невизначеності та неповноти знань. Рішення у вигляді множини команд на ВМ можуть бути реалізованими чи асинхронно й паралельно, чи послідовно-паралельно. Для підтримки такого типу завдань застосовується декілька підходів. Розглянемо їх.

Управління на основі поведінки (*Behavior-Based Control*). У цьому підході система керування роботом підтримує декілька прототипів його поведінки, що виконуються одночасно. Особливістю цієї архітектури є те, що кожна поведінка може діяти незалежно та активуватися або ж деактивуватися залежно від поточної ситуації та контексту [9, 10].

Підхід до керування роботами на основі архітектури підпорядкування (*Subsumption Architecture*) передбачає різні рівні контролю. Вищі рівні контролю можуть підпорядковувати нижчі, і це дає змогу виконувати декілька команд одночасно. Наприклад, є поведінка нижчого рівня з функцією уникнення перешкод, і вона працює постійно. Водночас на поведінку вищого рівня покладене завдання пошуку певної мети, і за потреби цей рівень може переривати або коригувати дії нижчого рівня. Коли робот виконує конкретні завдання, цей підхід забезпечує надійність та гнучкість [11].

Архітектура управління завданнями (*Task Control Architecture*). Зазначений підхід надає засоби для опису, координації та виконання завдань на високому рівні абстракції. Основні компоненти архітектури: менеджер завдань, роботи, які виконують завдання, і механізми

для їх комунікації та синхронізації. Роботи здатні виконувати завдання асинхронно, тобто без очікування завершення попередніх подій чи команд [12].

Концепція *Behavior-Based Control* найбільше бере до уваги згадану вище особливість прийняття керівних рішень в автономних системах. Ця концепція управління в автономних системах розвивається різними напрямками штучного інтелекту (ШІ) для роботів [13]. Щоб подолати проблему налаштування роботів на певні послуги, що зазвичай досягаються за допомогою навчання, і консолідувати зусилля роботизованих лабораторій по всьому світу, було запущено проєкт RT-X [14]. Мета проєкту – створення узагальненого робота мозку (*General Robots Brain, GRB*) унаслідок збору інформації, ресурсів і коду, що стосуються навичок, яким роботи вже навчені [15]. ШІ відіграє провідну роль, оскільки збір і створення знань про семантичні зв'язки між об'єктами в оточенні роботів покладають на штучний інтелект. Одним з нових підходів для роботів є штучний інтелект, що відчуває (*Feeling AI, FAI*). Його дослідженню присвячені праці [9, 16, 17]. У статті розглядається можливість застосування моделі ЙНА для реалізації керівних рішень м'якого управління у FAI.

Мета роботи й завдання

Для пояснення проблеми розглянемо завдання з керування автономною безлюдною системою на прикладі колісного робота. Нехай місія робота полягає в перевезенні вантажу з вихідної позиції до позиції 2 за заданим маршрутом (рис. 1).

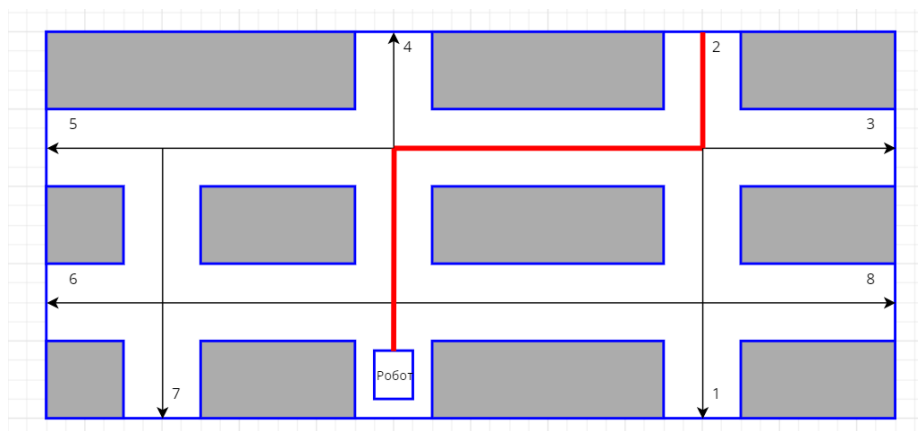


Рис. 1. Маршрут переміщення колісного робота

Колісний робот реалізує керівні рішення за допомогою типових команд, наприклад, рух

у якомусь напрямку з певною швидкістю, поворот коліс на певний кут, прискорення, гальмування тощо.

У разі жорсткого програмного управління параметри цих команд є попередньо розрахованими й не змінюються. Наприклад, для переміщення це може бути рух уперед з певною швидкістю протягом 30 с, потім виконання повороту на місці на 90° . Керівна програма такого типу має чітку послідовність команд, реалізацію яких підтримує контролер програмного управління. У разі гнучкого програмного управління команди цієї програми подібні до тих, що використовуються в жорсткому програмному управлінні, але вони містять умови завершення команди у вигляді прототипів даних від сенсорів. Наприклад, для руху прямо на 120 см контролер зчитує показники сенсора одометра, і після отримання

даних, що відповідають умовам, здійснює перехід до наступної команди, визначеної в керівній програмі.

Описані підходи мають деякі обмеження. Наприклад, робот, наближаючись до повороту, має спочатку знизити швидкість і плавно повернути колеса для виконання маневру повороту. У певні моменти часу необхідно, щоб не завершувалася поточна команда й починала виконуватися інша, потім обидві команди реалізовувалися одночасно.

Приклад розподілу в часі виконання команд за умови наближення до повороту, під час повороту та продовження руху наведені на рис. 2. На рисунку видно, що кожна наступна команда виконується без завершення попередньої.

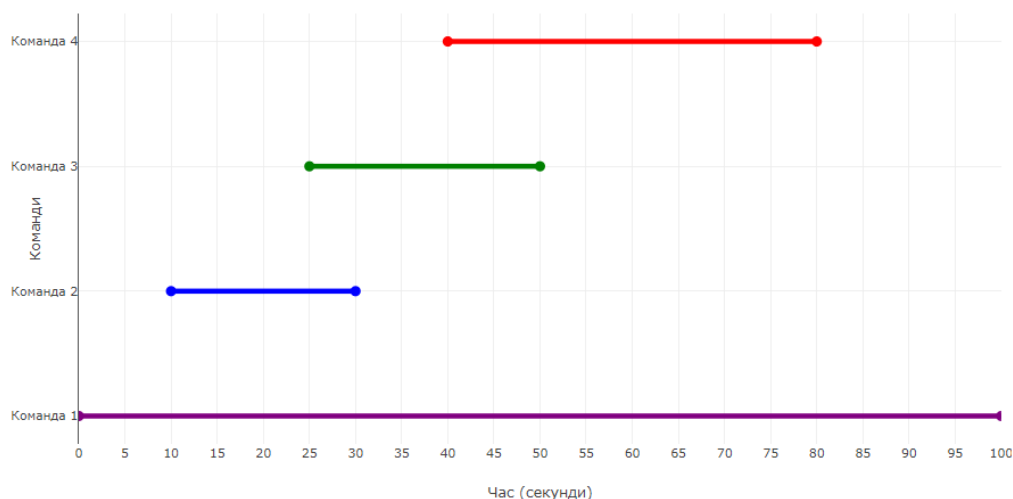


Рис. 2. Виконання команд робота з часом

На рис. 2 проілюстровано виконання таких команд: команда 1 (K1) – рух уперед, команда 2 (K2) – гальмування, команда 3 (K3) – поворот направо, команда 4 (K4) – прискорення та команда 5 (K5) – поворот наліво. Наведений комплекс команд з м'якою синхронізацією в часі є типовим для методів ситуаційного управління з нечіткими моделями прототипів ситуацій [17, 18]. Але всі переваги від використання нечітких моделей прийняття керівних рішень нівелюються останньою фазою нечіткого оброблення даних, а саме фазою дефазифікації. Усі відомі моделі нечіткого виведення виконують попереднє усереднення рішень, отриманих різними незалежними вирішувачами, а потім обчислення чисельного значення сигналу управління. Такий підхід звужує можливості автономних систем, оскільки завжди має пріоритет чи основний внесок рішення-переможця. Рішення, які мають незначну впевненість, ніколи не можуть бути реалізованими.

Водночас такі "слабкі" рішення виникають через неповноту знань, і без їх пробної реалізації неможливе самонавчання, що є необхідною функцією автономної системи.

Отже, для автономних систем, що приймають керівні рішення в умовах невизначеності, недостатньо одного використання м'якого управління, необхідна нова модель ВМ, яка давала б змогу поряд з "потужними" щодо впевненості реалізувати й "слабкі" керівні рішення. Модель такого ВМ пропонується запозичити у живих істот.

У праці [18] запропоновано модель актуатора, запозичену в біологічного актуатора, а саме м'язи живих істот, і розглянуто можливості її застосування в робототехніці. Модель повною мірою бере до уваги нечіткі характеристики керівного рішення у вигляді впевненості, динаміку реалізації рішення у вигляді постійного запізнення. Однак у моделі відсутня можливість реалізувати "слабке" рішення, отримане

нечіткою системою управління, оскільки воно не може конкурувати з "потужним" нечітким рішенням з більшим фактором упевненості.

Мета статті – запропонувати модель ЙНА та дослідити можливості її застосування для реалізації прийнятих FAI рішень в автономних системах.

Ймовірісно-нечітка модель актуатора, запозичена у живих істот

На практиці, залежно від інтерфейсу системи управління та ВМ, розрізняють кілька типів ВМ [18].

ВМ першого типу приймає на вхід цифровий код із певного діапазону $[K_{min}, K_{max}]$, а на виході створює фізичну властивість, кількісна характеристика якої лежить у діапазоні $[0, N]$ і відповідає значенню вхідного коду.

Другий тип ВМ реалізує багатоопозиційне управління загалом. Наприклад, двопозиційне управління реалізує "увімкнене" / "вимкнене", трипозиційне управління швидкістю "стоп" / "мала" / "середня". Багатоопозиційний ВМ можна також розглядати як представника першого типу: на вході цифровий код із певного діапазону, а на виході – фізичний параметр, що відповідає обраній позиції.

Для третього типу ВМ на вхід надходить сигнал "зменшити" або "збільшити" на величину вхідного коду управління, або "зберегти" попереднє (поточне) значення фізичного параметра. Наприклад, команди "трохи чи сильно загальмувати або прискоритися". ВМ декрементує або інкрементує поточне значення вихідного фізичного параметра на величину з певного діапазону $[N_{min}, N_{max}]$ відповідно до значення вхідного коду або не змінює нічого.

Отже, незалежно від типу ВМ, відповідно до інтерфейсу системи управління з ВМ, усі три типи однакові: за значенням вхідного коду $q \in [Q_{min}, Q_{max}]$ формується вихідний фізичний параметр із значенням $n \in [N_{min}, N_{max}]$.

Нехай поведінка робота, сформована FAI, визначається на множині з L можливих команд управління та реалізується L універсальними ВМ. Кожному ВМ відповідає актуатор, що виконує роль нечітко-ймовірнісного інтерфейсу між вихідними змінними системи прийняття рішень FAI та ВМ.

Вихідні змінні FAI – це нечіткі характеристики команд, наприклад "трохи прискоритися" або "встановити середню швидкість".

Система прийняття керівних рішень FAI формує як нечітку характеристику команди "нечіткий фактор впевненості" (НФВ) $-1 \leq cf \leq +1$, який є чисельною оцінкою впевненості, що актуатор має бути активований. Значення $cf = +1$ відповідає повній впевненості, а $cf = -1$ означає повну впевненість навпаки, що актуатор не має бути активований [16].

Отже, на вхід актуатора надходить НФВ cf , а на вихід – код $m \in [0, M]$. Актуатори є універсальними як за реалізованою функцією, так і за входом/виходом. Значення вихідного коду актуатора масштабується під значення вхідного коду ВМ і слугує вхідним значенням для реальних ВМ. Актуатор реалізує ймовірісно-нечітке перетворення значення НФВ cf у чітке число m . Тому будемо використовувати запропоновану аббревіатуру ЙНА, посилаючись на модель актуатора, викладену нижче.

Модель ЙНА запозичена у живих істот, а точніше симулює поведінку м'яза, який містить M незалежних м'язових волокон [19]. Нижче розглянуто модель елементарного актуатора (ЕА) – біологічного виконавчого механізму, тобто клітини м'яза, або м'язового волокна.

Динаміка реакції на команду як вплив рівня активації та ступеня актуальності на скоротливу реакцію волокна наведена в графічному вигляді на рис. 3. Показано реакцію м'язового волокна на одиночний подразник (a), два (b) і три (c) подразники, що відстають один від одного на певний часовий інтервал. З графіків випливають два важливі висновки. Перший полягає в тому, що реакція починається із запізненням на час $n\tau$ після активації волокна. Другий висновок: одиночний подразник незалежно від його амплітуди недостатній для того, щоб волокно скоротилося на максимально можливе значення.

У третьому випадку скорочення м'язового волокна досягає максимальних значень. І, нарешті, з рис. 3, c видно, що за умови 20 і більше стимулів волокно скорочується до максимального значення та зберігає цей стан деякий час, після чого розслабляється через втомлюваність. На основі цих даних додано модель ЕА, яка, власне, є чіткою.

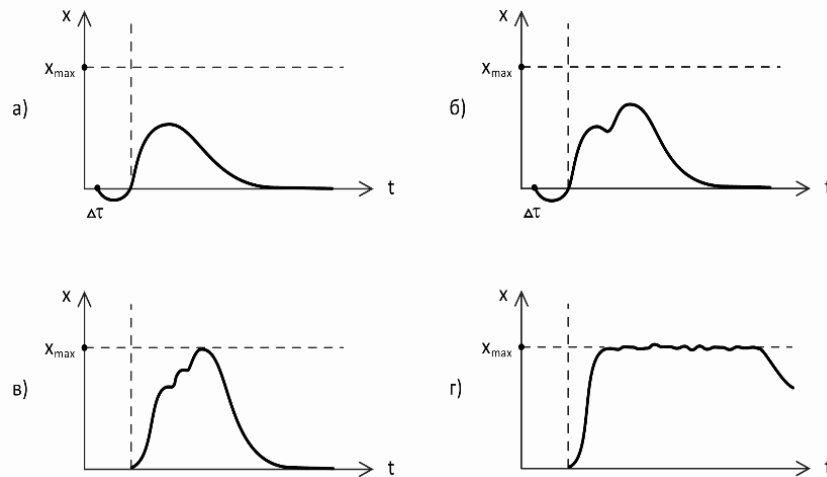


Рис. 3. Залежність реакції м'язового волокна від кількості подразників

Стан ЕА залежить від кількості дискретних інтервалів часу, коли вхід був активований. У моменти часу $k\tau$ на вхід ЕА надходить інформація про активацію, і ЕА може перейти в інший стан та змінити значення вихідної змінної з 0 на 1 або навпаки. Приймається автоматоподібна модель.

Динамічну характеристику задано двома числами – n_1 та n_2 . Коли значення вихідної змінної $x=0$, і в момент часу t на вхід ЕА надходить інформація про активацію, то за деякий час у момент $t+n_1\tau$ зміниться значення виходу ЕА, тобто $x=1$. Якщо до моменту часу $t+n_2\tau$ активації не було, то значення виходу зміниться на $x=0$. Додано три стани: спокою (П), розігріву (Р) та активності (А). У кожен такт часу $k\tau$ ЕА може перебувати в одному з перелічених станів і залежно від значення вхідної змінної здатний переходити з одного стану в інший відповідно до графа на рис. 4. На дугах графа зображені умови переходу, а після скісної риски вказані додаткові обчислювальні операції.

Наприклад, $I_p := n_1$ змінній I_p , що відповідає стану "розігрів", надається значення параметра n_1 динамічної характеристики. Аналіз рисунка показує, що в стані "спокій" у моменти часу $k\tau$ у разі значень вхідної змінної $\alpha \leq 0$ ЕА залишається в цьому самому стані до наступного моменту часу $(k+1)\tau$, а за умови $\alpha > 0$ переходить у стан "розігрів" з виконанням операції $I_p := n_1$.

Аналогічно, в стані "розігрів", якщо обидва значення вхідного сигналу $\alpha \leq 0$ і $\alpha > 0$, ЕА може як залишитися в цьому стані (на рисунку дуги не позначені умовами $\alpha \leq 0$ і $\alpha > 0$), так і перейти в стан "активність", якщо змінна $I_p = 0$. Перехід з останнього в стан "спокій" можливий за умови $I_A = 0$, у цьому разі не виконується жодна додаткова обчислювальна операція (за вертикальною рисою вказаний знак порожньої множини).

Проаналізуємо поведінку ЕА на прикладі вхідної послідовності для значень параметрів динамічної характеристики $n_1 = 3$ і $n_2 = 4$.

$$\begin{aligned} \alpha(0) &= -1, \alpha(\tau) = -1, \alpha(2\tau) = +1, \alpha(3\tau) = +1, \alpha(4\tau) = +1, \alpha(5\tau) = -1, \alpha(6\tau) = +1, \\ \alpha(7\tau) &= -1, \alpha(8\tau) = -1, \alpha(9\tau) = -1, \alpha(10\tau) = -1. \end{aligned} \quad (1)$$

Послідовності (1), згідно з графом (рис. 4), відповідають послідовності станів і значень вихідної змінної:

$$\begin{aligned} S(0) &= \text{П}, S(\tau) = \text{П}, S(2\tau) = \text{Р}, S(3\tau) = \text{Р}, S(4\tau) = \text{Р}, S(5\tau) = \text{Р}, \\ S(6\tau) &= \text{А}, S(7\tau) = \text{А}, S(8\tau) = \text{А}, S(9\tau) = \text{А}, S(10\tau) = \text{А}, S(11\tau) = \text{А}, S(12\tau) = \text{П}. \end{aligned}$$

$$\begin{aligned} x(0) &= 0, x(\tau) = 0, x(2\tau) = 0, x(3\tau) = 0, x(4\tau) = 0, x(5\tau) = 0, \\ x(6\tau) &= +1, x(7\tau) = +1, x(8\tau) = +1, x(9\tau) = +1, x(10\tau) = +1, x(11\tau) = +1, x(12\tau) = 0. \end{aligned}$$

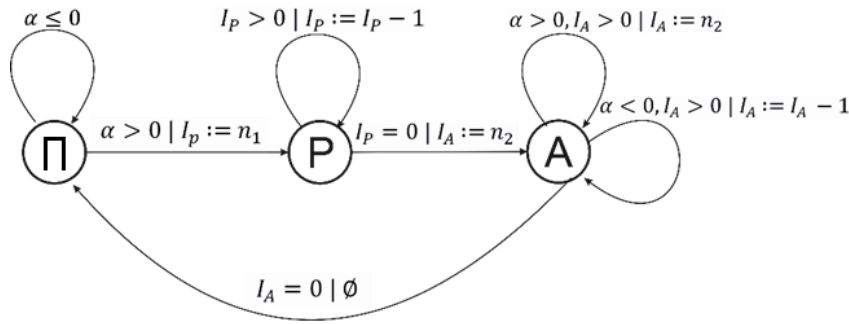


Рис. 4. Граф переходів станів ЕА

Після того, як було додано модель ЕА, опишемо модель ЙНА як систему, створену з M незалежних ЕА. Розглядаємо динамічну модель, що функціонує в дискретні моменти часу $k\tau$, $k=1,2,\dots$, де τ – інтервал дискретизації. На момент $k\tau$ на вході ЙНА є поточне значення НФВ $cf(k\tau)$. Стан ЙНА подає множину станів усіх M ЕА $\{(l_p^i(k-1)\tau), (l_r^i(k-1)\tau), (l_A^i(k-1)\tau), i=1,2,\dots,M\}$. Ймовірно-нечіткий механізм ЙНА на підставі значення НФВ, по-перше, формує значення ймовірності на вході кожного ЕА, по-друге, методом рулетки знаходить для кожного ЕА значення функції активації його входу $\alpha_i(k\tau) = \{0 \text{ чи } 1\}$, $i=1,2,\dots,M$. Значення $\alpha_i(k\tau)=0$ говорить про те, що ЕА на цей час не буде активованим, а $\alpha_i(k\tau)=1$, навпаки, означає, що на вході ЕА буде 1. На наступному етапі оброблення вхідної інформації ЙНА розраховують значення виходів для всіх ЕА $x_i(k\tau)$ $i=1,2,\dots,M$. Алгоритм обчислення $x_i(k\tau)$ спочатку знаходить стан $\{(l_p^i(k\tau), l_r^i(k\tau), l_A^i(k\tau))\}$ i -го ЕА згідно з можливими переходами (рис. 4) для заданого значення входу $\alpha_i(k\tau)$, а потім для нового вектора стану знаходиться $x_i(k\tau) = \{0 \text{ чи } 1\}$. І на останньому етапі обчислюється значення вихідної змінної моделі ЙНА

$$m(k\tau) = \sum_{i=1}^M x_i(k\tau). \quad (2)$$

У роботі розглядаємо два варіанти переходу від НФВ до ймовірностей подій.

Перший варіант ґрунтується на дослідженнях з біології [19]. Щоб активізувати біологічний

актуатор, в ділянку, де він розташований, випускаються ліганди – спеціальні молекули, які запускають хімічні реакції, що активують актуатор. Кількість (концентрація) лігандів залежить від характеристики збурення нейронів. Нейробіологи оцінюють параметри перелічених процесів ймовірностями. Отже, якщо прийняти, що значення НФВ визначає ступінь збурення нейрона, то концентрація лігандів в оточенні ЕА може бути подана кількістю лігандів, кожен з яких має вірогідність зіткнутися з ЕА й активізувати його. Для цього варіанта застосована наступна спрощена ймовірнісна модель. Множина можливих подій $\{A_j, j=0,1,2,\dots,J(k\tau)+1\}$, де $J(k\tau) = f(cf(k\tau))$, $cf \in [-1, +1]$ – це кількість елементів-лігандів, запущених для активізації i -го ЕА; A_0 – подія, що жоден з лігандів не зіткнеться з ЕА; A_j – подія що j -й ліганд зіткнеться з ЕА. Кількість лігандів $J(k\tau)$ залежить від значення НФВ. Завдання – знайти розподіл імовірностей для повної групи незалежних подій і обрати тільки одну з них методом рулетки.

У другому варіанті значення НФВ $\{cf_i, i=1,2,\dots,n\}$, $cf \in [-1, +1]$ розглядають як безпосереднє значення ймовірності, що i -й ЕА буде активізованим на цьому такті часу, тобто на вході i -го ЕА буде $\alpha_i(k\tau)=1$. Завдання ймовірнісного механізму – знайти множину ЕА, які будуть активовані в цьому такті часу. Для кожного ЕА незалежно методом рулетки вирішується, чи буде він активованим, чи ні.

Для першого варіанта спочатку знаходять ймовірності $P_0 = P(A_0)$, $P_1 = P(A_1)$, ..., $P_J = P(A_J)$ повної групи подій $P_0 + P_1 + \dots + P_J = 1$.

Ймовірності розраховують таким чином:

$$\begin{aligned} P_0 &= \frac{1.75 - cf}{2}, \quad P_j = \frac{(0.25 + cf)/J}{2}, \quad j = 1, 2, \dots, J, \quad \text{якщо } cf \geq 0; \\ P_0 &= 1, \quad P_1 = \dots = P_J = 0, \quad \text{якщо } cf_i \leq 0.75; \\ P_0 &= \frac{1.25 + |cf|}{2}, \quad P_j = \frac{(0.75 - |cf|)/J}{2}, \quad j = 1, 2, \dots, J, \quad \text{якщо } 0.75 < cf_i < 0. \end{aligned} \quad (3)$$

Алгоритм визначення значення вхідної змінної $\alpha_i(k\tau)$ ЕА подано нижче:

```

start Main_Program
Step 1. Initialization
initialize array = { -0.5, 0.5, 0.6 }
normalizedArray = NormalizeArray(array)
Step 2. Perform roulette selection
selectedIndex = RouletteSelection(normalizedArray)
print "Selected index: " + selectedIndex + " Selected
value: " + normalizedArray[selectedIndex]
end Main_Program

```

RouletteSelection(normalizedArray)

Step 1. Initialization

initialize *rnd* as Random

initialize *randomValue* as *rnd.NextDouble()*

Step 2. Iterating through normalizedArray

initialize *cumulative* as 0.0

for each element in normalizedArray **do**

cumulative += element

if *randomValue* <= *cumulative* **then**

return current index

end if

end for

return 0

end RouletteSelection

NormalizeArray(arr)

Step 1. Initialization

initialize *normalizedArr* as array of doubles with
length equal to *arr* length

Step 2. Conversion from confidence to probability

for each *element* in *arr* **do**

normalizedArr[*i*] = (*element* + 1) / 2

end for

Step 3. Calculation of sum of normalized values

initialize *sum* as 0

for each item in *normalizedArr* **do**

sum += item

end for

Step 4. Normalization to sum of 1

for each *element* in *normalizedArr* **do**

normalizedArr[*i*] /= *sum*

end for

return *normalizedArr*

end NormalizeArray

У другому варіанті також спочатку для заданого значення НФВ знаходять ймовірності активації кожного ЕА за тими самими формулами (3), що й для першого варіанта за умови $J = 1$. Потім визначають, чи буде активованим ЕА для кожного окремо ЕА, чи ні. З цією метою також використано метод рулетки для кожного з ЕА. Алгоритм цього завдання виглядає так:

start Main_Program

Step 1. Initialization

initialize array = { -0.5, 0.25, 0.5 }

probabilityArray = ConvertToProbabilityArray(array)

selectedList = RouletteSelection(array,

probabilityArray)

end Main_Program

RouletteSelection(array, probabilityArray)

Step 1. Initialization

initialize *selectionList* as empty list of integers

initialize *rnd* as Random

Step 2. Iterating through array elements

for each *element* in array **do**

randomValue = *rnd.NextDouble()*

 print "Array item: " + *element* + " Probability

value: " + *probabilityArray*[*i*]

if *randomValue* <= *probabilityArray*[*i*] **then**

 print "Random Value: " + *randomValue* + "

Result of selection: True"

 add 1 to *selectionList*

else

 print "Random Value: " + *randomValue* + "

Result of selection: False"

 add 0 to *selectionList*

end if

end for

return *selectionList*

end RouletteSelection

ConvertToProbabilityArray(arr)

Step 1. Initialization

initialize probabilityArray as array of doubles with
length equal to arr length

**Step 2. Conversion from confidence to probability
for each element in arr do**

probabilityArray[i] = (element + 1) / 2

end for

return probabilityArray

end ConvertToProbabilityArray

Результати моделювання

У процесі моделювання поведінки колісного робота з використанням ЙНА розглянуто рух робота по маршруту, що зображено червоною лінією на рис. 1.

$$q(k\tau) = Q_{\min} + \frac{m(k\tau)}{Q_{\max} - Q_{\min}} \quad \text{однопозиційний ВМ;}$$

$$q_i(k\tau) = \begin{cases} 1, & \text{if } m_i(k\tau) = m^*, \\ 0, & \text{other wise,} \end{cases} \quad \text{де } m^* = \text{MAX}(m_1(k\tau), m_2(k\tau), \dots, m_n(k\tau)) \quad (4)$$

n -позиційний ВМ.

Нижче наведено фрагменти результатів моделювання п'яти кроків руху робота по маршруту (рис. 1). Розглянуто спрощену часову модель: система управління генерувала команди з періодичністю $T=10$ одиниць часу, і протягом цього інтервалу НФВ команди на вході ЙНА не змінювалися. ЙНА використовував ці значення з періодичністю $\tau=1$, тобто 10 разів за інтервал T обчислювали вихідну

змінну ЙНА. Розрахунки виконували за моделлю другого варіанта, коли НФВ на вході ЙНА розглядали як безпосереднє значення ймовірності.

Крок 1. Початок руху колісного робота.

Робот починає рух з початкової точки. Система м'якого управління (СМУ) FAI прийняла керівні рішення щодо руху прямо переважно з малою швидкістю. Вектор управління має такий вигляд:

$$\begin{aligned} &((cf_{K1\text{стоп}}(T=0) = -0.95, cf_{K1\text{мала}}(T=0) = +0.85, cf_{K1\text{велика}}(T=0) = +0.3), \\ &(cf_{K2\text{вимкнуте}}(T=0) = +0.65, cf_{K2\text{увімкнуте}}(T=0) = -0.95), \\ &(cf_{K30}(T=0) = -0.95, cf_{K345}(T=0) = -0.95, cf_{K390}(T=0) = -0.95), \\ &(cf_{K4}(T=0) = +0.35), (cf_{K50}(T=0) = -0.95, cf_{K545}(T=0) = -0.95, cf_{K590}(T=0) = -0.95)). \end{aligned} \quad (5)$$

Вихідний вектор кодів ЙНА, що подаються на ВМ, має такі значення:

$$\begin{aligned} &((m_{K1\text{стоп}}(T+k\tau) = 0/0, m_{K1\text{мала}}(T+k\tau) = 78/1, m_{K1\text{велика}}(T+k\tau) = 17/0), \\ &(m_{K2\text{вимкнуте}}(T+k\tau) = 67/1, m_{K2\text{увімкнуте}}(T+k\tau) = 0/0), \\ &(m_{K30}(T+k\tau) = 0/0, m_{K345}(T+k\tau) = 0/0, m_{K390}(T+k\tau) = 0/0), \\ &(m_{K4\text{вимкнуте}}(T+k\tau) = 0/0, m_{K4}(T+k\tau) = 26/98), \\ &(m_{K50}(T+k\tau) = 0/0, m_{K545}(T+k\tau) = 0/0, m_{K590}(T+k\tau) = 0/0)), \quad k = 4, 5, \dots, 9. \end{aligned} \quad (6)$$

У (6) наведено по два значення кодів, наприклад, для $m_{K1\text{мала}}(T+k\tau) = 78/1$. Перша цифра 78 показує

значення вихідної змінної (1) ЙНА (кількість ЕА із 100 мають значення виходу $x=1$). Друга цифра –

це значення коду $q_i(k\tau)$ в (4), що подається на вхід ВМ. Наприклад, код входу ВМ команди K1 є (0,1,0). Ще один коментар до (6) стосується того, що вказано початкове значення часу $k = 4$. Це пов'язано з початком моделювання множини ЕА, оскільки обрано $n_1 = 3$ в моделі (рис. 4).

Крок 2. Виконання повороту направо.

Моделюється повороту направо на перехресті (рис. 1), коли СМУ FAI прийняла неоднозначне рішення: рухатися прямо на тій самій передачі швидкості "мала", пригальмовуючи, або переключити передачу швидкості на "стоп" та виконати поворот направо на 90°. Вектор управління має такий вигляд:

$$\begin{aligned} &((cf_{K1\text{стоп}}(T=10) = +0.49, cf_{K1\text{мала}}(T=10) = +0.55, cf_{K1\text{велика}}(T=10) = -0.95), \\ &(cf_{K2\text{вимкнуте}}(T=10) = -0.15, cf_{K2\text{увімкнуте}}(T=10) = +0.65), \\ &(cf_{K30}(T=10) = -0.85, cf_{K345}(T=10) = +0.45, cf_{K390}(T=10) = +0.85), \\ &(cf_{K4}(T=0) = -0.95), (cf_{K50}(T=10) = -0.95, cf_{K545}(T=10) = -0.95, cf_{K590}(T=10) = -0.95)). \end{aligned} \quad (7)$$

Отримано вихідний вектор кодів ЙНА на момент часу $T+0$

$$\begin{aligned} &((m_{K1\text{стоп}}(T+0) = 47/0, m_{K1\text{мала}}(T+0) = 53/1, m_{K1\text{велика}}(T+0) = 0/0), \\ &(m_{K2\text{вимкнуте}}(T+0) = 6/0, m_{K2\text{увімкнуте}}(T+0) = 48/1), \\ &(m_{K30}(T+0) = 0/0, m_{K345}(T+0) = 29/0, m_{K390}(T+0) = 77/1), \\ &(m_{K4}(T+0) = 0/0), (m_{K50}(T+0) = 0/0, m_{K545}(T+0) = 0/0, m_{K590}(T+0) = 0/0)), \end{aligned}$$

який принципово не змінювався протягом трьох наступних часових інтервалів $T+\tau$, $T+2\tau$, $T+3\tau$. А на момент часу $T+4\tau$ ЙНА отримав

принципово інші значення для ВМ, що реалізує команду K1, а саме замість рух на передачі "мала" реалізував "стоп".

$$((m_{K1\text{стоп}}(T+4\tau) = 51/1, m_{K1\text{мала}}(T+0) = 46/0, m_{K1\text{велика}}(T+0) = 0/0), \dots).$$

Висновки

ЙНА, на відміну від наявних методів реалізації керівних рішень у моделях м'якого управління, відкриває можливості реалізації команди, що не набула абсолютної переваги з-поміж усіх потенційно можливих команд і не була першою в пріоритетному переліку. Таку особливість потребують актуатори, що застосовуються в автономних системах. В умовах невизначеності, коли автономна система опиняється в незнайомій ситуації, для якої попередньо не було сформовано прототипу поведінки, всі наявні прототипи реакції визначаються незначною впевненістю в тому, що темплейт, який зберігається в пам'яті, відповідає фактичній ситуації. У цьому разі важливо спробувати

різні прототипи поведінки, а не тільки той, що в рейтингу відповідності стоїть першим з незначною різницею від інших. Цей висновок підтверджують дослідження з біології, що механізм самонавчання живої істоти оснований на методі "спроб і помилок". Щоб знайти й запам'ятати відповідну реакцію, потрібна диверсифікація реакцій на незнайому ситуацію. Комп'ютерні експерименти, наведені в роботі, підтвердили можливість реалізації цього механізму за допомогою запропонованої моделі ЙНА.

Надалі заплановано дослідити універсальну модель ЙНА й розробити рекомендації щодо її застосування в різних застосунках м'якого управління, зокрема управління колісним роботом на основі даних від різних сенсорів.

Список літератури

1. Litman T. A. Autonomous Vehicle Implementation Predictions: Implications for Transport Planning. Victoria Transport Policy Inst., Aug. 2022. URL: <https://www.vtpi.org/avip.pdf> (дата звернення: 27.11.2024).
2. Siciliano B., Khatib O. (Eds.). Modeling and Control of Underwater Robots. *Springer Handbook of Robotics* (2nd ed.). Berlin, Heidelberg: Springer-Verlag, 2016. 2225 p. DOI: <https://doi.org/10.1007/978-3-319-32552-1>
3. Belpaeme T., Kennedy J., Ramachandran A., Scassellati B., Tanaka F. Social robots for education: A review. *International Journal of Social Robotics*. 2018. Vol. 10. No. 3. P. 299–315. DOI: <https://doi.org/10.1126/scirobotics.aat5954>
4. Pagliarini L., Lund H. H. The future of Robotics Technology. *Journal of Robotics, Networking and Artificial Life*. Vol. 3. No. 4. March 2017. P. 270–273. DOI: 10.2991/jrnal.2017.3.4.12
5. Mubin O., Stevens C. J., Shahid S., Mahmud A. A., Dong J. J. A Review of the Applicability of Robots in Education. *Journal of Technology in Education and Learning*. 2013. Vol. 1. No. 1. P. 1–7. DOI: 10.2316/Journal.209.2013.1.209-0015
6. Singh P., Dulebenets M. A., Pasha J., Gonzalez E. D. R. S., Lau Y.-y., Kampmann R. Deployment of Autonomous Trains in Rail Transportation: Current Trends and Existing Challenges. *IEEE Access*. 2021. Vol. 9. P. 1550–1562. DOI: 10.1109/ACCESS.2021.3091550
7. Thrun S., Burgard W., Fox D. Probabilistic Robotics. MIT Press, 2005. Розділ 5: Robot Motion. P. 91–119. URL: <https://docs.ufpr.br/~danielsantos/ProbabilisticRobotics.pdf> (дата звернення: 27.11.2024).
8. Piegat A (2001) Fuzzy modelling and control. Physica-Verlag, 728 p.
9. Kargin A., Petrenko T. Feeling Artificial Intelligence for AI-Enabled Autonomous Systems. In *Proc. of 2022 IEEE Global Conf. on Artificial Intelligence and Internet of Things (GCAIoT)*. Alamein New City, Egypt, Dec. 18, 2022, P. 88–93. DOI: 10.1109/GCAIoT57150.2022.10019235
10. Beetz M., Jain D., Mösenlechner L., Tenorth M. Towards performing everyday manipulation activities. *Robotics and Autonomous Systems*. 2010. P. 1–10. DOI: 10.1016/j.robot.2010.05.007
11. Arkin R. Behavior-Based Robotics. *Automatica*. 1998. P. 69–79. DOI: 10.1016/S0005-1098(02)00169-3
12. Brooks R. A Robust Layered Control System for a Mobile Robot. *IEEE Journal on Robotics and Automation*. 1986. Vol. 2, № 1. P. 14–23. DOI: 10.1109/JRA.1986.1087032
13. Simmons R. Experience with a Task Control Architecture for Mobile Robots. 1989. P. 4–8. URL: https://www.academia.edu/51291699/Experience_with_a_Task_Control_Architecture_for_Mobile_Robots (дата звернення: 27.11.2024).
14. Guizzo E. Types of Robots. Categories frequently used to classify robots. Robotsguide.com. URL: <https://robotsguide.com/learn/types-of-robots> (дата звернення: 15.06.2024).
15. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. Open X-Embodiment Collaboration, 2024. URL: <https://robotics-transformer-x.github.io/> (дата звернення: 15.06.2024).
16. Levine S., Hausman K. The global project to make a general robotic brain. *IEEE Spectrum*. Jan 2024. URL: <https://spectrum.ieee.org/global-robotic-brain> (дата звернення: 15.06.2024).
17. Kargin A., Petrenko T. Knowledge Distillation for Autonomous Intelligent Unmanned System. In Pedrycz W., Chen S.-M. *Advancements in Knowledge Distillation: Towards New Horizons of Intelligent Systems*. Studies in Comput. Intell., vol. 1100. Springer International Publishing, 2023, P. 193–230. DOI: 10.1007/978-3-031-32095-8
18. Kargin A., Petrenko T. Spatio-Temporal Data Interpretation Based on Perceptual Model. In Mashtalir V., Ruban I., Levashenko V. *Advances in Spatio-Temporal Segmentation of Visual Data. Studies in Computational Intelligence*, Vol. 876. Springer, Cham, 2020, P. 101-159. DOI: 10.1007/978-3-030-35480-0_3
19. Hengji Wang, Joshua Swore, Shashank Sharma, and Adrienne L. Fairhall. A complete biomechanical model of Hydra contractile behaviors, from neural drive to muscle to movement. *PNAS*, March 10, 2023, 120 (11). DOI: 10.1073/pnas.2210439120

References

1. Litman, T. A. (2022), "Autonomous Vehicle Implementation Predictions: Implications for Transport Planning, Victoria Transport Policy Institute", available at: <https://www.vtpi.org/avip.pdf> (last accessed 27.11.2024).

2. Siciliano, B., Khatib, O. (2016), "Modeling and Control of Underwater Robots". *Springer Handbook of Robotics* (2nd ed.), Springer-Verlag, Berlin, Heidelberg, 2225 p. DOI: <https://doi.org/10.1007/978-3-319-32552-1>
3. Belpaeme, T., Kennedy, J., Ramachandran, A., Scassellati, B., Tanaka, F. (2018), "Social robots for education: A review", *International Journal of Social Robotics*, Vol. 10, No. 3, P. 299–315. DOI: <https://doi.org/10.1126/scirobotics.aat5954>
4. Pagliarini, L., Lund, H. H. (2017), "The future of Robotics Technology", *Journal of Robotics, Networking and Artificial Life*, Vol. 3, No. 4, P. 270–273. DOI: 10.2991/jrnal.2017.3.4.12
5. Mubin, O., Stevens, C. J., Shahid, S., Mahmud, A. A., Dong, J. J. (2013), "A Review of the Applicability of Robots in Education", *Journal of Technology in Education and Learning*, Vol. 1, No. 1, P. 1–7. DOI: 10.2316/Journal.209.2013.1.209-0015
6. Singh, P., Dulebenets, M. A., Pasha, J., Gonzalez, E. D. R. S., Lau, Y.-y., Kampmann, R. (2021), "Deployment of Autonomous Trains in Rail Transportation: Current Trends and Existing Challenges", *IEEE Access*, Vol. 9, P. 1550–1562. DOI: 10.1109/ACCESS.2021.3091550
7. Thrun, S., Burgard, W., Fox, D. (2005), "Probabilistic Robotics, MIT Press, Chapter 5: Robot Motion", P. 91–119, available at: <https://docs.ufpr.br/~danielsantos/ProbabilisticRobotics.pdf> (last accessed 27.11.2024).
8. Piegat A (2001), Fuzzy modelling and control. Physica-Verlag, 728 p.
9. Kargin, A., Petrenko, T. (2022), "Feeling Artificial Intelligence for AI-Enabled Autonomous Systems. In Proc". *IEEE Global Conf. on Artificial Intelligence and Internet of Things (GCAIoT)*. Alamein New City, Egypt, Dec. 18, 2022, P. 88–93. DOI: 10.1109/GCAIoT57150.2022.10019235
10. Beetz, M., Jain, D., Mösenlechner, L., Tenorth, M. (2010), "Towards performing everyday manipulation activities", *Robotics and Autonomous Systems*. P. 1–10. DOI: 10.1016/j.robot.2010.05.007
11. Arkin, R. (1998), "Behavior-Based Robotics", *Automatica*. P. 69–79. DOI: 10.1016/S0005-1098(02)00169-3
12. Brooks, R. (1986), "A Robust Layered Control System for a Mobile Robot", *IEEE Journal on Robotics and Automation*, Vol. 2, No. 1, P. 14–23. DOI: 10.1109/JRA.1986.1087032
13. Simmons, R. (1989), "Experience with a Task Control Architecture for Mobile Robots", P. 4–8, available at: https://www.academia.edu/51291699/Experience_with_a_Task_Control_Architecture_for_Mobile_Robots (last accessed 27.11.2024).
14. Guizzo, E. (2024), "Types of Robots. Categories frequently used to classify robots", *Robotsguide.com*, available at: <https://robotsguide.com/learn/types-of-robots> (last accessed 15.06.2024).
15. Open X-Embodiment Collaboration (2024), "Open X-Embodiment: Robotic Learning Datasets and RT-X Models", available at: <https://robotics-transformer-x.github.io/> (last accessed 15.06.2024).
16. Levine, S., Hausman, K. (2024), "The global project to make a general robotic brain", *IEEE Spectrum*, Jan. 2024, available at: <https://spectrum.ieee.org/global-robotic-brain> (last accessed 15.06.2024).
17. Kargin, A., Petrenko, T. (2023), "Knowledge Distillation for Autonomous Intelligent Unmanned System", in Pedrycz, W., Chen, S.-M. (Eds.), *Advancements in Knowledge Distillation: Towards New Horizons of Intelligent Systems*, Studies in Computational Intelligence, Vol. 1100, Springer International Publishing, P. 193–230. DOI: 10.1007/978-3-031-32095-8
18. Kargin, A., Petrenko, T. (2020), "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, P. 101-159. DOI: 10.1007/978-3-030-35480-0_3
19. Hengji, Wang, Joshua, Swore, Shashank, Sharma, and Adrienne, L. Fairhall (2023), "A complete biomechanical model of Hydra contractile behaviors, from neural drive to muscle to movement". *PNAS*, March 10, 2023, 120 (11). DOI: 10.1073/pnas.2210439120

Надійшла (Received) 01.02.2025

Відомості про авторів / About the Authors

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

Кузьменко Роман Сергійович – Український державний університет залізничного транспорту, аспірант кафедри інформаційних технологій, Харків, Україна; e-mail: kuzmenko88@kart.edu.ua; ORCID ID: <https://orcid.org/0009-0000-1675-8658>

Kargin Anatolii – Doctor of Sciences (Engineering), Professor, Ukrainian State University of Railway Transport, Department of Information Technology, Kharkiv, Ukraine.

Kuzmenko Roman – Ukrainian State University of Railway Transport, PhD Student at the Department of Information Technology, Kharkiv, Ukraine.

PROBABILISTIC-FUZZY ACTUATOR MODEL IN THE SOFT CONTROL CIRCUIT OF AN AUTONOMOUS UNMANNED SYSTEM

The **subject matter** of the article is artificial intelligence methods and models used in autonomous unmanned system control. The **goal** of the work is to create a new actuator model for autonomous unmanned systems that implements control decisions made by Artificial Intelligence under conditions of uncertainty. The following **tasks** were solved in the article: a model of a Probabilistic Fuzzy Actuator (PFA) is proposed and the possibility of its application as a universal controller of the actuators in autonomous systems is investigated. The PFA model, borrowed from biological muscle actuators, is formalized as a set of automata-like elements with a probabilistic mechanism for assigning their input variables calculated on the basis of fuzzy characteristics of control decisions obtained from an AI system that supports soft control technology. The following **methods** are used – fuzzy control, decision-making under uncertainty based on the confidence factor, automata theory, probability theory. The following **results** were obtained – a model of PFA borrowed from living beings has been proposed and substantiated; a PFA algorithm has been developed that implements control decisions obtained by the soft control method. **Conclusions:** Probabilistic Fuzzy Actuator, unlike existing methods of implementing decisions in soft control models, opens up the possibility of implementing commands that do not have an absolute advantage among all potentially possible ones when making decisions. This capability of autonomous system actuators is useful in conditions when the system encounters an unfamiliar situation since all reaction prototypes existing in its memory are characterized by low confidence. In these cases, to maintain autonomy, it is important to try different behaviors, not just the one that ranks first. Besides this, the "trial and error" method is still required by the self-learning model in autonomous systems that rely on it. Computer experiments confirmed the possibility of implementing this mechanism using the proposed PFA model.

Keywords: artificial intelligence; soft-computing-based control; autonomous unmanned system; fuzzy systems; probabilistic models.

Бібліографічні описи / Bibliographic descriptions

Каргін А. О., Кузьменко Р. С. Модель ймовірнісно-нечіткого актуатора в контурі м'якого управління автономною безлюдною системою. *Сучасний стан наукових досліджень та технологій в промисловості*. 2025. № 1 (31). С. 61–72. DOI: <https://doi.org/10.30837/2522-9818.2025.1.061>

Kargin, A., Kuzmenko, R. (2025), "Probabilistic-fuzzy actuator model in the soft control circuit of an autonomous unmanned system", *Innovative Technologies and Scientific Solutions for Industries*, No. 1 (31), P. 61–72. DOI: <https://doi.org/10.30837/2522-9818.2025.1.061>
