

# Адаптивний алгоритм встановлення стійкої позиції гексапода

Оліфірович М.Я., Деркач А.А., Борецький В.Ф.  
Київський національний університет імені Тараса Шевченка  
м. Київ, Україна  
turdecarta@gmail.com

**Анотація**—Використовуючи симулятор *Coppeliasim*, підготовлено симуляцію для дослідження пошуку оптимальної позиції та руху для гексапода. Алгоритм включає в себе нейронну мережу, що навчалася з використанням генетичного алгоритму. Результатом експерименту стало знаходження алгоритмом стійкої позиції гексапода відповідно до заданих умов.

**Ключові слова**—рухи робота, алгоритм руху, оптимальний рух, *Coppeliasim*, нейронні мережі, генетичний алгоритм.

## Вступ

Останнім часом зростає кількість статей присвячених автоматизованим системам, особливо присвячених автоматизації процесів виробництва. Такі дослідження мають за мету підвищити ефективність та зменшити затрати часу та ресурсів. Особливий інтерес представляють дослідження підвищення ефективності рухів таких систем [4]. Окрему область досліджень представляє застосування нейронних мереж для вирішення таких задач.

Мета даної роботи – дослідити алгоритм пошуку оптимальної позиції, використовуючи еволюційний підхід до навчання нейронних мереж [1]. Для дослідження використовувався програмний продукт *Coppeliasim* з ліцензією EDU. Симулятор використовувався як фізичний рушій, а код алгоритму написаний на Python [3]. Суть експерименту полягала в тому, що гексаподу потрібно досягти максимальної висоти тіла, зберігаючи стійку позицію.

## Моделювання

Для моделювання використовувався напіввідкритий програмний продукт *Coppeliasim*, що використовується світовою спільнотою для проектування різного роду автоматизованих систем, таких як промислові АBB маніпулятори, автоматизовані автомобілі, хобійні роботи тощо. Перевагою цього симулятора є наявність ліцензії EDU та простий для вивчення інтерфейс, а також те, що в ньому представлений широкий набір моделей різних роботів, серед яких є гексапод з оберненою кінематикою [2]. Цю модель було задіяно в експерименті. *Coppeliasim* надає API для передачі керування процесом моделювання до зовнішньої програми, виступаючи в ролі симулятора фізичної взаємодії моделей на полі.

Обернена кінематика гексапода обраховувалася симулятором.

Керуюча програма написана на Python містить у собі блок з нейронною мережею, параметри якої налаштовуються за допомогою генетичного алгоритму. Відбір параметрів нейронної мережі із популяції відбувається за двома критеріями: перший – найбільша висота (позиція Z координати) другий – стійке положення протягом циклу навчання (робот не перевертається на бік або на спину).

Для навчання використовувався звичайний одношаровий перцептрон, на вхід якого подавалися значення повороту кожного серводвигуна (6 ніг по 3 серводвигуна на кожному, загальна кількість двигунів – 18). Вихідними параметрами були значення кутів повороту серводвигунів для однієї, двох, трьох або шести ніг, які копіювалися у випадку однієї, двох та трьох ніг на шість, три і дві ноги відповідно.

## РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ

На рис.1 зображено чотири криві навчання для різних конфігурацій нейронної мережі. По горизонталі – T (епохи навчання). По вертикалі – h (позиція робота по Z координаті в метрах). Максимальна висота, якщо робота розташувати повністю вертикально на всіх ногах, складає 0,197 м.

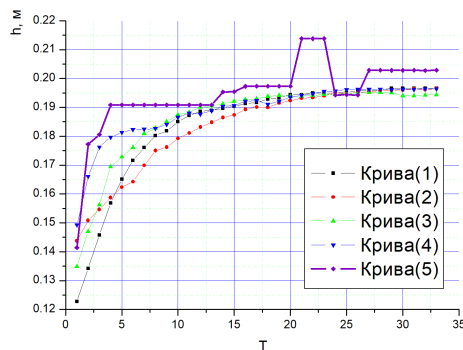


Рис. 1. Криві навчання

Під час експерименту розглядалося 4 випадки передачі даних для керування роботом. Для пер-

шого випадку (Крива 1) нейронна мережа розраховувала параметри для 1 ноги (на виході було 3 нейрони), далі дані дублювалися для кожної із ніг. Для другого випадку (Крива 2) нейронна мережа розраховує параметри для двох ніг (2 ноги \* 3 нейрона на нозі = 6 нейронів). Ці набори параметрів розподіляються навпіл для ніг (для перших трьох ніг один набір і для других трьох ніг інший набір). Для третього випадку (Крива 3) нейронна мережа генерувала дані для трьох ніг. Ці параметри передаються на 6 ніг (1 набір однаковий для двох ніг). Для випадку 4 (Крива 4 і Крива 5) нейронна мережа генерує 18 вихідних сигналів (для кожної ноги свій набір даних з трьох нейронів), усі параметри передаються для кожної ноги окремо. Крива 5 – особливий випадок, коли було знайдено такі ваги нейронної мережі, що позиція робота по Z координаті перевищила очікувану максимальну позицію гексаподу, ніж для випадку, коли б він дотикався до поверхні всіма кінцівками (рис. 2).

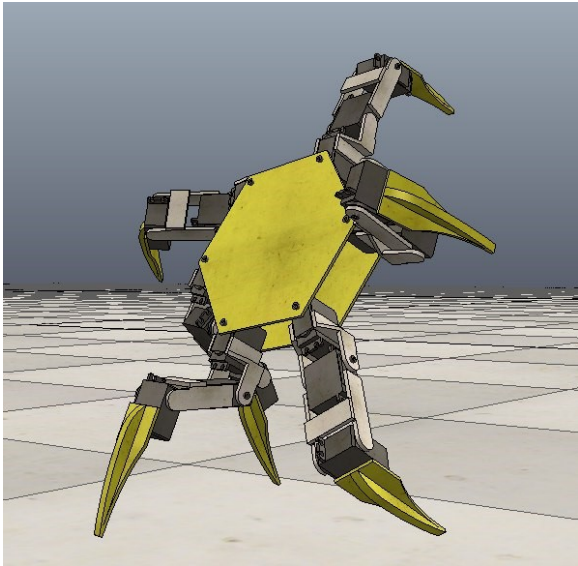


Рис. 2. Варіант положення кінцівок, де перевищено висоту найбільш очікуваної позиції

Нейронна мережа при даних конфігураціях навчається впродовж 10-15 епох.

#### Висновки

Використовуючи CoppeliaSim та Python, підготовлено програмне забезпечення для роботи з нейронною мережею. Проведено симуляцію алгоритму пошуку стійкого положення робота в просторі. Знайдено конфігурації нейронної мережі для найвищої стійкої позиції робота. Було знайдено конфігурацію, при якій позиція робота по Z координаті перевищила очікувану максимальну позицію гексаподу, коли б він дотикався до поверхні всіма кінцівками.

#### ЛІТЕРАТУРА

- [1] Саттон Р.С. Обучение с подкреплением / Р.С. Саттон, Э.Г. Барто; пер. с англ. – Москва : БИНОМ. Лаборатория знаний, 2011. – 399. с.
- [2] Климчик А.С. Разработка управляющих программ промышленных роботов / А.С. Климчик, Р.И. Голицкий, Ф.В. Фурман, К.И. Сёмкин; Курс лекций для студентов специальности «Информационные технологии и управление в технических системах», Белорусский государственный университет информатики и радиоэлектроники, Кафедра систем управления. – Минск : 2008. – 131с.
- [3] Pablo Martinez Campos Simulacion Cinematica Y Dinamica Del Robot Hexapodo Escalador Romerin – Madrid, 2019. – 118с.
- [4] <https://rosindustrial.org/news/2020/8/4/real-time-trajectory-planning-for-industrial-robots> (дата звернення 05.08.2020).