

УДК 519.7:004.8

Я.В.Лотиш, Л.О.Гуменюк, В.В.Лотиш  
Politechnika Lubelska (Польща)  
Луцький національний технічний університет

## КОМП'ЮТЕРНА ІМІТАЦІЙНА МОДЕЛЬ СОРТУВАЛЬНОГО ПРИСТРОЮ З ЕЛЕМЕНТОМ НЕЙРОУПРАВЛІННЯ

**Я.В. Лотиш, Л.О. Гуменюк, В.В. Лотиш.** Комп'ютерна імітаційна модель сортувального пристрою з елементом нейроуправління. В роботі отримана імітаційна модель сортування зі звичайним режимом роботи та з режимом з нейроуправлінням. Вона дозволяє імітувати роботу сортувального механізму. Розробка даної моделі дозволила відпрацювати режими нейронавчання та нейророботи механізму сортування.

**Ключові слова:** імітаційна модель, нейронні системи, нейроуправління, сортування об'єктів.

**Ya.V.Lotysh, L.A.Gumenyuk, V.V.Lotysh.** Computer simulation model of the sorting devices with an element of neurocontrol. We obtain the simulation model of sorting with normal operation and with the regime of neurocontrol. It allows you to simulate the work of sorting mechanism. Development of this model allowed to work out a regimes of neuro control and neuro work of sorting mechanism.

**Keywords:** simulation model, neural system, neurocontrol, sorting objects.

**Я.В.Лотыш, Л.А.Гуменюк, В.В.Лотыш.** Компьютерная имитационная модель сортировочного устройства с элементом нейроуправления. В работе получена имитационная модель сортировки с обычным режимом работы и с режимом с нейроуправлением. Она позволяет имитировать работу сортировочного механизма. Разработка данной модели позволила отработать режимы нейрообучения и нейророботы механизма сортировки.

**Ключевые слова:** имитационная модель, нейронные системы, нейроуправление, сортировка объектов.

### Постановка проблеми

#### Аналіз останніх досліджень та публікацій

На сьогодні штучні нейронні системи отримали широке поширення для вирішення різноманітних задач інтелектуального управління, прогнозування в умовах невизначеності, нелінійності, стохастичності завдяки своїм можливостям навчання за даними, що характеризують функціонування досліджуваного об'єкта.

Такі системи представляють собою новий міждисциплінарний науковий напрям, близький до інтелектуальних систем, тому, що, на відміну від керованих систем, які розвиваються на стику кібернетики, математики, електроніки та інших наук, нейронні системи синтезуються з нейроінформатикою, зокрема з нейрокеруванням.

Аналіз відомих теоретичних та інженерних підходів до управління складними об'єктами показав, що найбільш доцільною на сьогодні є технологія „інтелектуального” управління, де АСУ наслідує інтелектуальні функції людини при управлінні певним об'єктом, тобто має здатність до імітації мислення та навчання по відношенню до об'єкту управління, впливу навколишнього середовища та умов свого функціонування. З інженерної точки зору інтелектуальне управління повинно мати наступні властивості: здатність до навчання та адаптивність, живучість, простий алгоритм керування, дружній по відношенню до користувача людино - машинний інтерфейс, здатність до включення нових компонентів, що забезпечують кращі рішення в умовах обмежень, що накладаються технічними засобами. Найкраще цим умовам відповідають нейромережеві методи управління завдяки властивостям адаптації та узагальнення.

**Метою роботи** є розробка імітаційної комп'ютерної моделі сортування зі звичайним режимом роботи та з режимом з нейроуправлінням (відлагодження режиму нейроуправління).

#### Основні результати дослідження.

Для відображення роботи системи управління сортування об'єктів розроблена імітаційна модель.

Імітаційна модель працює в двох режимах роботи :

- перший - реакція на спрацювання однієї з кнопок;
- другий - використовуючи опереджувальну можливість сортування на базі нейро підходу.

Вхідними даними алгоритму є початкове положення гнучкого механізму, стан кнопок на екранній формі, а також режим роботи моделі (Без нейро, Нейро).

Якщо імітаційна модель знаходиться в режимі «Без нейро», то реакцією на вхідні дані буде нахил гнучкого механізму в одну із трьох сторін, після чого він повертається в початкове положення. Напрямок нахилу буде залежати від того, яка саме з кнопок була натиснута.

Якщо модель перебуває в режимі «Нейро», то вона здійснює навчання завдяки використанню навчальної вибірки. При натисканні на кнопку здійснюється поворот механізму та повернення його в початкове положення. Далі проходить обробка даних, на основі якої визначається положення виконавчого механізму. Обробка даних проходить з використанням навченої нейронної мережі, завдяки якій визначається реакція моделі, що матиме вигляд пошуку найбільшої ймовірності натискання однієї з кнопок та повернення гнучкого механізму в певне положення.

Для програмної реалізації імітаційної моделі застосовується мова програмування Delphi з використанням компонента NeuralBase [1] для роботи з нейромережею.

Оскільки програма має два режими роботи – вибирається потрібний (Без нейро, Нейро) (рис.1.1).

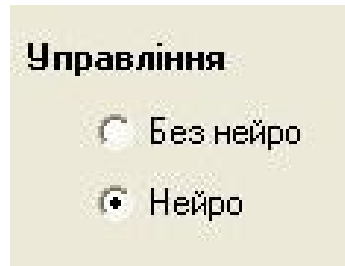


Рис.1.1. Режими роботи імітаційної моделі.

Якщо вибраний режим Без нейро, то при натисканні на одну з трьох кнопок механізм здійснить рух у відповідну сторону (рис.1.2), після чого повернеться в початкове положення.

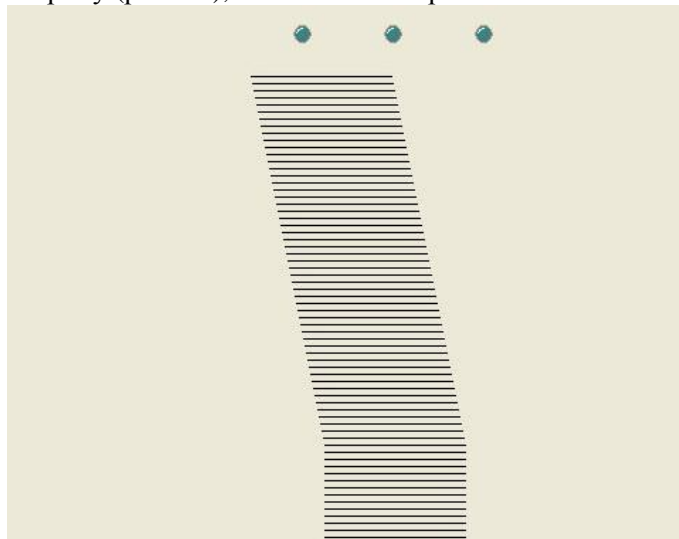


Рис.1.2.Імітаційна модель гнучкого механізму.

На прикладі натискання на ліву кнопку програмна реалізація має вигляд :

```
AbLED1.Checked:=True;  
Vlivo();  
sleep(5);  
AbLED1.Checked:=False;  
NazadVpravo();
```

Процедури Vlivo() та NazadVpravo() створюють імітацію руху гнучкого механізму, при якому Vlivo() повертає механізм вліво, а NazadVpravo() – в початкове положення.

Якщо вибраний режим Нейро, тоді використовується компонент NeuralBase для роботи з нейромережами (рис.1.3).



Рис.1.3.Компонент NeuralBase.

На початку програми формується навчальна вибірка – видаляються всі попередні приклади з неї. Це виконується завдяки процедурі `NeuralNetBP.ResetPatterns`. Далі створюються вхідні та вихідні вектори нейромережі :

```
SetLength(xInputVector, 3);  
SetLength(xOutputVector, 1);
```

Потім у вектори записується інформація, на основі якої нейронна мережа проходитиме навчання. Ці дані стануть прикладами для нейронної мережі, по яких буде прийматися рішення.

```
xInputVector[0] := 0;  
xInputVector[1] := 0;  
xInputVector[2] := 0;  
xOutputVector[0] := 0.4;
```

Приклади будуть записані в навчальну вибірку за допомогою функції `NeuralNetBP.AddPattern()`.

Далі необхідно натиснути на кнопку «Навчити» (рис.1.4). В процедурі `TMainForm.Button3Click` змінній `prbEpoch.Max` присвоюється кількість епох навчання. Під час запуску цієї процедури активізується процедура `TMainForm.NeuralNetBPEpochPassed` для відображення процесу навчання. Процес навчання проходить завдяки використанню функції `NeuralNetBP.TeachOffLine` [2-5].

```
NeuralNetBP.EpochCount := strtoint(Edit2.text);  
prbEpoch.Max := NeuralNetBP.EpochCount;  
prbEpoch.Position := 0;  
NeuralNetBP.TeachOffLine;
```

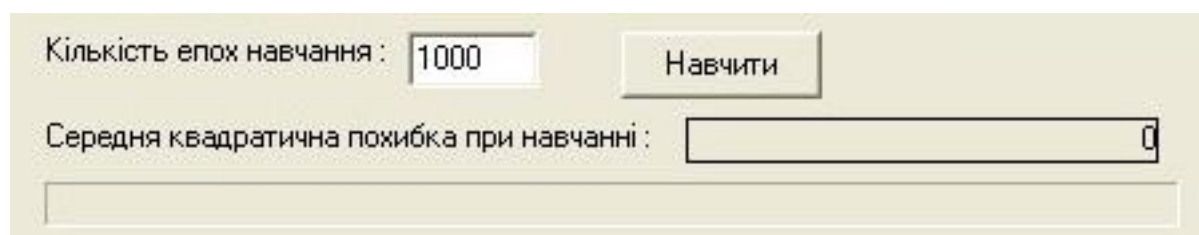


Рис.1.4.Навчання нейронної мережі

Процедура `TMainForm.NeuralNetBPEpochPassed`, яка використовується для відображення швидкості навчання та середньої квадратичної помилки при навчанні (рис.1.5), має вигляд:

```
prbEpoch.Position := prbEpoch.Position + 1;  
sttError.Caption := FloatToStr(NeuralNetBP.TeachError);
```

Тут `prbEpoch.Position` – це поточний стан перебігу навчання, який відображається на панелі, а `sttError.Caption` – середня квадратична похибка при навчанні.



Рис.1.5. Відображення процесу навчання.

При натисканні на одну з трьох кнопок, імітаційна модель здійснить нахил виконавчого механізму у відповідне положення і поверне його в початкове.

Потім програма присвоїть змінній *S* значення кнопки, яка була натиснута, та відправить його в процедуру *Port1RxChar*, де проходить основна обробка даних.

В процедурі *Port1RxChar* створюється вхідний вектор *SetLength(xInputVector, 3)*.

Залежно від того, яке саме значення *S* було відправлене, налаштовується вхідний вектор .

Вхідний вектор ініціалізується значеннями, відповідними до кількості натискань:

```
InputVector[0]:=Natusk1;  
xInputVector[1]:=0;  
xInputVector[2]:=0;
```

За допомогою функції *NeuralNetBP.Compute(xInputVector)* вхідний вектор подається на вхід нейронної мережі. Отримані значення з вихідного вектора записуються у вихідний масив *Mas\_imov*.

```
Mas_imov[1]:= NeuralNetBP.Output[0]
```

Позиція гнучкого механізму визначається завдяки функції *pozition()*, в яку відправляються дані з масиву *Mas\_imov*.

Отримані результати записуються у змінну *R1*.

```
R1:=pozition(Mas_imov);
```

Змінні *R1* та *R0* відображають значення, де найбільша ймовірність натискання кнопки (*R1*) та попереднє положення сервоприводу (*R0*). Процедура *Mechanism* відображає положення, в якому буде перебувати виконавчий механізм.

```
Mechanism(R0,R1);
```

### Висновки

Таким чином, отримана імітаційна модель, яка дозволяє імітувати роботу сортувального механізму. Робота даної моделі дозволила відпрацювати режими нейронавчання та нейророботи механізму сортування. В подальшому даний підхід буде використано при розробці програмного апаратного комплексу сортування на базі нейропідходу.

### Список використаних джерел

1. <http://www.basegroup.ru/>
2. <http://iasa.org.ua/lectons/tpr/neuro/hopfield.htm>
3. [https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F\\_%D1%81%D0%B5%D1%82%D1%8C\\_%D0%A5%D0%BE%D0%BF%D1%84%D0%B8%D0%BB%D0%B4%D0%B0](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C_%D0%A5%D0%BE%D0%BF%D1%84%D0%B8%D0%BB%D0%B4%D0%B0)
4. J. Hopfield, «Neural networks and physical systems with emergent collective computational abilities», *Proceedings of National Academy of Sciences*, vol. 79 no. 8 pp. 2554—2558, April 1982.
5. [http://sernam.ru/book\\_gen.php?id=69](http://sernam.ru/book_gen.php?id=69)
6. Суровцев И.С., Ключин В.И., Пивоварова Р.П. Нейронные сети. — Воронеж: ВГУ, 1994. — 224 с.
7. Уоссермен Ф. Нейрокомпьютерная техника: теория и практика. — М.: Мир, 1992.
8. Горбань А.Н. и др. Нейроинформатика. — Электронная публикация.
9. Интернет: Sarle, W.S., ed. (1997), Neural Network FAQ, part 1-7: Introduction, periodic posting to the Usenet newsgroup comp.ai.neural-nets, URL ftp://ftp.sas.com/pub/neural/FAQ.html.
10. Мкртчян С.О. Нейроны и нейронные сети. (Введение в теорию формальных нейронов) — М.: Энергия, 1971. — 232 с.
11. Гилл Ф., Мюррей У., Райт М. Практическая оптимизация. - М.: Мир, 1985. - 509 с.
12. Лоскутов А.Ю., Михайлов А.С. Введение в синергетику. — М.: Наука, Гл. ред. физ.-мат. лит., 1990. - 272 с.
13. Muller B., Reinhardt J. Neural Networks. An introduction. — Berlin: Springer-Verlag, 1991. — 266p.
14. Волькенштейн М.В. Биофизика: Учеб. руководство. — М.: Наука, Гл. ред. физ.-мат.лит., 1988. - 592 с.
15. Басканова Т.Ф., Ланкин Ю.П. Нейросетевые алгоритмы самостоятельной адаптации.// Научная сессия МИФИ-99. Всероссийская научно-техническая конференция "Нейроинформатика-99". Сборник научных трудов. В 3 частях. Ч.1.- М.: МИФИ, 1999.- С.17-24.
16. Дорогов А.Ю., Алексеев А.А. Математические модели быстрых нейронных сетей. В сб. научн. тр. СПбГЭТУ "Системы управления и обработки информации". Вып.490, 1996.
17. Ф.Уоссермен, Нейрокомпьютерная техника, М., Мир, 1992.
18. Омату С., Халид М., Юсоф Р. Нейроуправление и его приложения. Пер. с англ. Н.В. Батина; Под. Ред. А.И. Галушкина, В.А. Птичкина. – М.: ИПРЖР, 2000. – 272 с.
19. Artificial Neural Networks: Concepts and Theory, IEEE Computer Society Press, 1992.