

КРАВЕЦЬ П.І.,
ЛУКІНА Т.Й.,
ШИМКОВИЧ В.М.,
ТКАЧ І.І.

РОЗРОБКА ТА ДОСЛІДЖЕННЯ ТЕХНОЛОГІЇ ОЦІНЮВАННЯ ПОКАЗНИКІВ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ МІМО-ОБ'ЄКТІВ УПРАВЛІННЯ

У даній роботі розглядається комплексний формалізований підхід до реалізації багатоетапної процедури ідентифікації складних динамічних об'єктів з використанням нейромережових модельних структур, в роботі розроблено програмний пакет «МІМО-Plant» в середовищі MatLab, для оцінювання показників нейромережових моделей МІМО-об'єктів управління, представлено його детальний опис та приклади дослідження.

In this paper describes an integrated formal approach to implement multistage procedure for identification of complex dynamic objects using neural network model structures in the developed software package «MIMO-Plant» in the environment MatLab, to assess the performance of neural network models MIMO-object of control, presented details and examples of research.

Реальні промислові об'єкти управління зазвичай є багатовимірними, тобто мають кілька входів і кілька виходів МІМО (multi input, multi output) [1, 2]. У ряді випадків такі об'єкти можна промодельовувати, нехтуючи другорядними впливами, а також другорядними керованими величинами. В результаті такого обґрунтованого спрощення можна отримати просту модель з двома (керуючим і збурюючим), а в деяких випадках і з одним впливом, і з однією керованою величиною. Це істотно спрощує як аналіз об'єкта, так і розробку системи автоматичного управління ним. Однак у багатьох випадках необхідно забезпечити управління декількома керованими величинами об'єкта, його вихідними величинами, шляхом впливу на декілька його входів, тобто зміни керуючих величин, плюс до того слід врахувати ще й кілька збурень. Особливість управління та моделювання такого, досить складного в описі об'єкта, полягає в тому, що може статися, що вплив по одному входу призводить до зміни не однієї, а відразу декількох керованих величин. Наприклад, вертоліт, якщо пілот вертольота бажає прискоритися, то він, природно збільшує подачу газу. Однак тільки це не призводить до збільшення швидкості: несучий гвинт починає обертатися швидше, і вертоліт прагне піднятися. Більш того, прискорення обертання несучого гвинта змушує вертоліт ще й повертатися навколо вертикальної осі, змінюючи напрямку руху. Отже, щоб прискорити політ вертольота

на тій же висоті і по тому ж напрямку, пілот повинен збільшити подачу палива і одночасно збільшити тягу заднього гвинта, змінюючи кут атаки його лопатей, а також нахилити вертоліт вперед (точніше, перекосити площина обертання лопатей головного гвинта), щоб додана збільшенням газу потужність витрачалася на політ вперед.

Нейромережові системи управління являють собою новий високотехнологічний напрямок в теорії управління та відносяться до класу нелінійних динамічних систем [3]. Однією з головних рис нейронних мереж є універсальні апроксимаційні властивості [4,5]. Нейронні мережі дозволяють з будь-якою точністю обраховувати довільну неперервну функцію багатьох змінних $f(s_1, \dots, s_n)$. Отже, з їх допомогою можливо як завгодно точно апроксимувати функцію, породжену будь-якою неперервною системою в тому числі і МІМО-об'єктами.

Метою даної роботи є розробка технології дослідження та оцінювання нейромережових моделей МІМО-об'єктів управління для вибору оптимальної структури нейромережі.

Для вирішення завдань ідентифікації і управління найбільш адекватними є багатошарові нейронні мережі прямого поширення (БНМПР) [6-9]. При використанні БНМПР для реалізації процедури ідентифікації динамічних систем необхідно визначити «зовнішню» і «внутрішню» структури нейронної мережі. «Зовнішня» структура нейромережової моделі

повністю визначається вектором входу і вектором виходу. При виборі «внутрішньої» структури нейромережевої моделі необхідно визначити:

- кількість прихованих шарів;
- кількість нейронів у кожному схованому шарі;
- вид активаційної функції для кожного шару.

Збільшення числа нейронів у схованому шарі і збільшення числа прихованих шарів підвищують репрезентативні можливості нейронної мережі, тобто дають можливість моделювати більш складні взаємозв'язки, але призводять до значних труднощів при практичній реалізації [10], збільшення часових витрат як на навчання, так і на роботу в режимі прогнозування. Це і пояснює факт прагнення використовувати мінімальну реалізацію БНМПП в більшості технічних додатків.

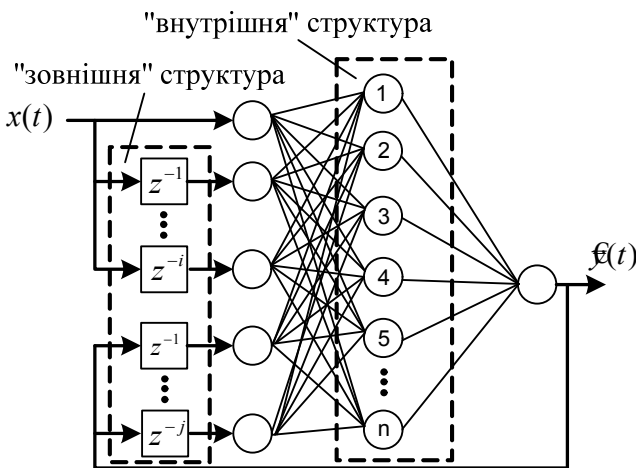


Рис. 1. Структурна схема багатшарової нейронної мережі прямого розповсюдження

Існує ряд теорем [4,6,11,12], які математично обґрунтовують апроксимуючу властивість нейронних мереж, з яких випливає що, багатшарові нейронні мережі з одним прихованим шаром, сигмоїдальними (сигмоїдальна, гіперболічний тангенс) функціями активації та лінійною функцією активації вихідного шару, може апроксимувати функції з заданою точністю в разі відсутності обмеження на число нейронів у схованому шарі. На рисунку 1 представлена структура БНМПП. Виходячи з вищевикладеного, завдання побудови нейромережевої моделі зводиться до визначення «зовнішньої» і «внутрішньої» структури, а саме визначення кількості затриманих входів i , затриманих сигналів

виходу мережі j , і кількості нейронів прихованого шару n .

Основна ідея роботи полягає в побудові оптимальної, в силу деякого критерію, моделі за результатами спостережень над вхідними та вихідними змінними системи. На практиці реалізація процедури ідентифікації (рисунок 2) вимагає вирішення цілого ряду допоміжних завдань, основними з яких є:

- планування/проведення експерименту і попередня обробка експериментальних даних;
- вибір модельної структури;
- оцінка (оптимізація параметрів) моделі;
- прийняття рішення про адекватність моделі.

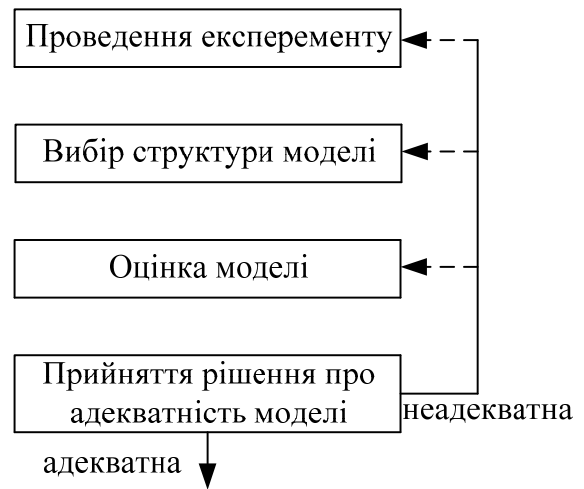


Рис. 2. Загальна схема реалізації процедури ідентифікації

Для вирішення даної задачі було розроблено програмний пакет «MIMO-Plant» в середовищі MatLab.

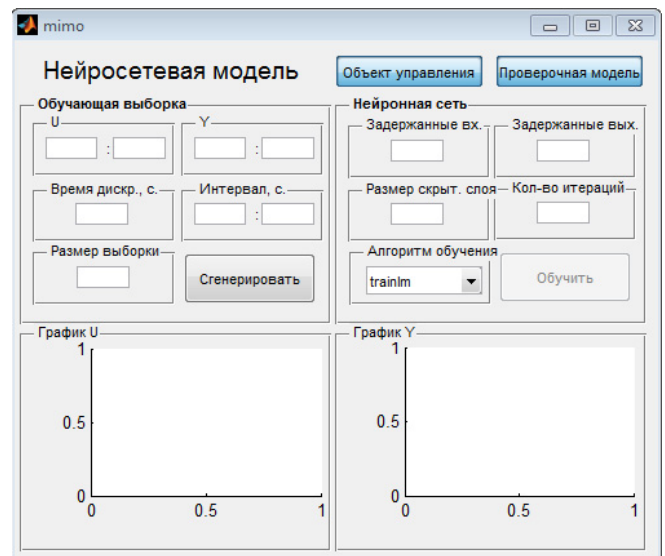


Рис. 3. Головне вікно програмного пакету «MIMO-Plant»

При натисканні на кнопку «Объект управления» відкриється модель об'єкта управління.

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

Рис. 4. Математична модель об'єкта управління

Завдання об'єкта відбувається за допомогою блоку State-Space. Цей блок реалізує систему, поведінку якої можна визначити як:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du,$$

де x – вектор станів, u – вхідний вектор, а y є вихідним вектором.

Матриця коефіцієнтів повинна мати такі характеристики:

- матриця **A** повинна бути n на n матриця, де n -число станів;
- матриця **B** повинна бути n на m матриця, де m -число входів;
- матриця **C** повинна бути r на n матриця, де r -число виходів;
- матриця **D** повинна бути r на m матрицю.

Ширина вхідного вектора залежить від кількості стовпців в матрицях **B** і **D**. Ширина вихідного вектора залежить від кількості рядків у матрицях **C** і **D**.

Підготовчим етапом створення нейромережі є генерація навчальної вибірки, яка відбувається при натисканні кнопки «Сгенерировать».

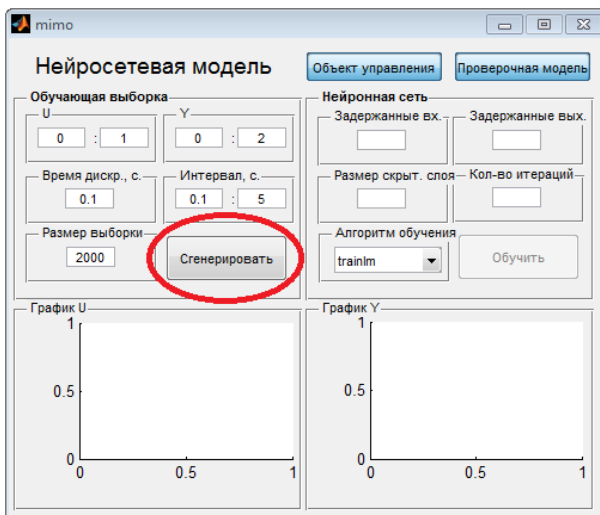


Рис. 5. Генерація навчальної вибірки

У полі навчальної вибірки необхідно задати параметри необхідні для генерації навчальної вибірки:

1. Задати діапазон зміни входу (U_{\max} , U_{\min}) і виходу (Y_{\max} , Y_{\min}) ОУ. При виборі Y_{\max} , не по-

трібно обмежувати вихід об'єкта управління, тобто Y_{\max} вибирати великим усталеного значення при U_{\max} .

2. Розмір навчальної вибірки слід вибирати таким, щоб вона містила всі можливі комбінації амплітуд і частот з робочого діапазону системи (при часу дискретизації 0.1 рекомендуємо брати розмір навчальної вибірки порядку 2000-4000).

3. Інтервал визначає мінімальну і максимальну тривалість тестового сигналу. Причому максимальне значення повинно бути не менше часу встановлення перехідного процесу ОУ при стрибкоподібному впливі. Якщо всі поля групи «Навчальна вибірка» заповнені, то в області *PlantInput* і *PlantOutput* повинні відображатися вид вхідного впливу на ОУ та реакція ОУ на вхідні дії.

Після того як дані готові, переходимо до завдання структури МНСПР і її навчання. Для цього необхідно заповнити поле «Нейронна мережа».

1. В полі «Розмір прихованого шару» вказуємо кількість нейронів n .
2. В полі «Кількість затриманих входів» вводимо значення i .
3. В полі «Кількість затриманих виходів» вводимо значення j .
4. В групі параметрів «Нейронна мережа», вибираємо алгоритм навчання (trainlm) і кількість навчальних ітерацій (200).
5. Натискаємо кнопку «Обучить».

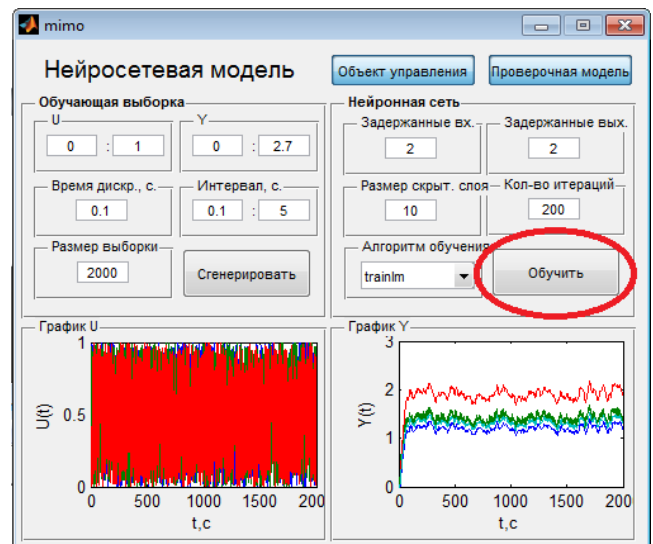


Рис. 6. Навчання нейромережесевих моделей МІМО-об'єктів управління

Результатом виконання процедури навчання буде згенерована нейромережа, яка являє собою набір нейромережесевих моделей ОУ (рисунком 7).

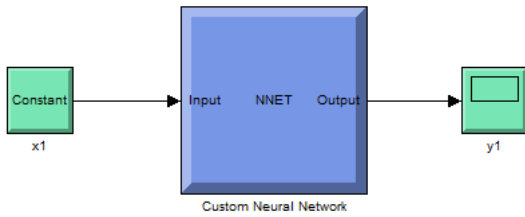


Рис. 7. Згенерована нейронна мережа

Для того, щоб подивитися перехідні процеси нейронної мережі та багатовимірного об'єкту управління, необхідно скористатися перевіркою моделлю, яка визивається при натисканні кнопки «Перевірочна модель». В результаті цих дій ми відкриємо наступну модель:

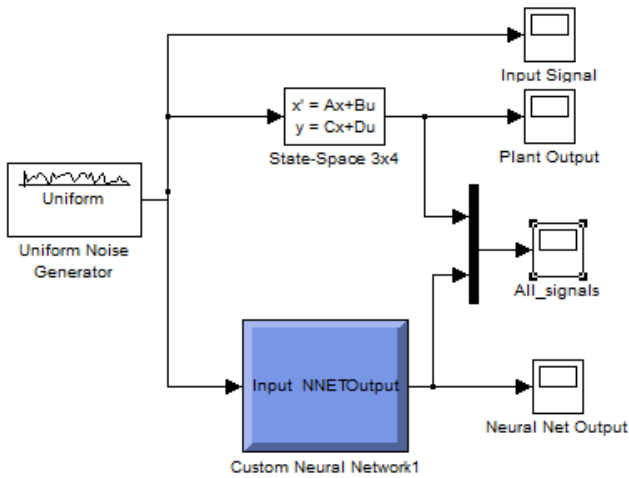


Рис. 8. Перевірочна модель

Нейронна мережа «Custom Neural Network1», яка зображена на рисунку 8, була отримана в результаті копіювання з вікна що з'являється після навчання нейронної мережі та підставлена у перевірочну модель.

При натисканні на блок «All_signals» перевірочної моделі ми можемо побачити графічне зображення виходу нейронної мережі та стандартного об'єкту управління, а також візуально порівняти отримані результати (рисунок 8).

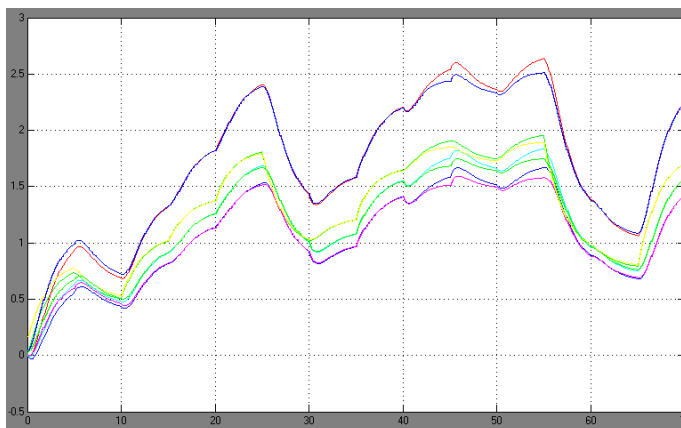


Рис. 9. Графічне зображення виходу нейронної мережі та стандартного об'єкту управління

Для обчислення сумарної середньоквадратичної помилки потрібно скористатися розробленою моделлю, де:

- 1) На вхід подається вхідний сигнал генератором сигналу.
- 2) Об'єкт управління задається блоком State-Space.
- 3) Під об'єктом управління знаходяться 16 варіацій структури нейромережі.
- 4) У блоці «Display» відображається сумарна середньоквадратична помилка, яка математично описується наступною формулою:

$$\varepsilon = \sqrt{\frac{\sum_{i=1}^z (\hat{y} - y)^2}{z}}$$

де z – кількість значень похибки (вибирати рівне кількості елементів вектора $tout$ з workspace Matlab); y – вихід моделі об'єкта або системи; \hat{y} – вихід нейромережової моделі об'єкта або системи.

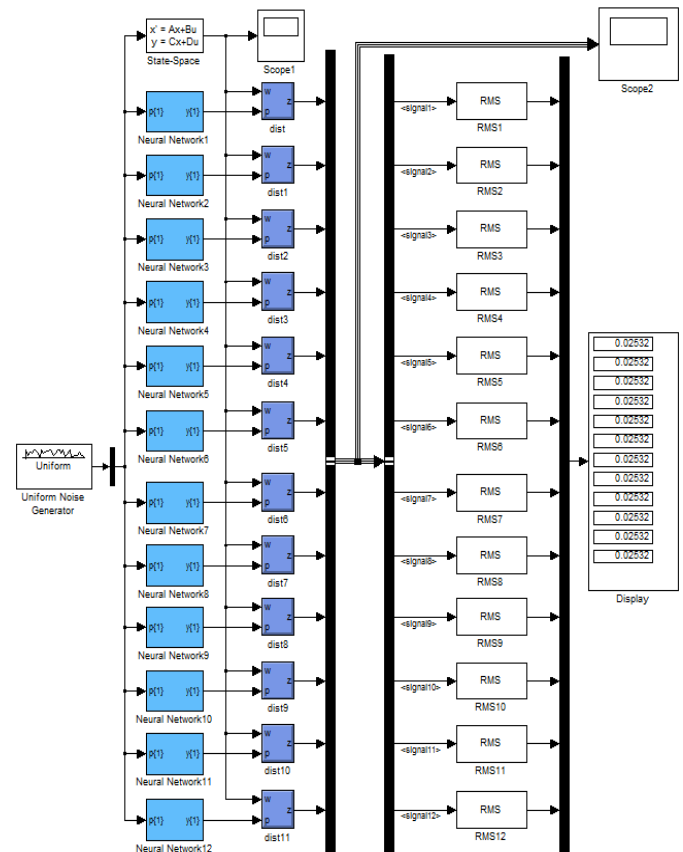


Рис. 10. – Модель обчислення сумарної середньоквадратичної помилки

Далі розглянемо приклад дослідження нейромережових моделей МІМО-об'єкту управління, який має 3 вхідних та 4 вихідних параметри, в програмному пакеті «МІМО-Plant».

Крок 1. Задаємо МІМО-об'єкт:

$$\mathbf{A} = \begin{pmatrix} -1.23 & 0 & 0 \\ 2.5 & -0.56 & 0 \\ 0 & 3 & -2.3 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0.2 & 1 & 0.3 \\ 0 & 0.4 & 0.1 \\ 0 & 0.2 & -1 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} 0 & 0 & 0.3 \\ 1 & 0 & 0.2 \\ 0.1 & 0.2 & 0.3 \\ 0.2 & 0 & 0.3 \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Для того, щоб задати об'єкт управління, вводим матриці A,B,C,D як параметри у вікно блоку «State-space», який визивається при натисканні кнопки «Объект управления» головного меню програми.

Крок 2. Вводим необхідні параметри для генерації навчальної вибірки, а саме:

- 1) Задаємо діапазон зміни входу від 0 до 1 та діапазон зміни виходу від 0 до 2.7.
- 2) Обираємо час дискретизації 0.1 с.
- 3) Обираємо інтервал від 0.1с до 5с.
- 4) Обираємо розмір навчальної вибірки 2000.
- 5) Натискаємо кнопку «Згенерувати».

Крок 3. Створюємо та навчаємо нейронну мережу. Для цього заповнюємо поля блоку «Нейронная сеть» головного меню програми, а саме:

1. У полі «Розмер скритого слоя» вказуємо кількість нейронів $n=5$.
2. У полі «Количество задержанных входов» вводим значення $i=0$.
3. У полі «Количество задержанных выходов» вводим значення $j=1$.

4. У полі «Алгоритм навчання» обираємо алгоритм навчання (trainlm) і кількість навчальних ітерацій (200). Нажимаємо кнопку «Обучить».

Крок 4. Після того, як нейронна мережа створена та навчена, підставляємо її у модель обчислення сумарної середньоквадратичної помилки. У блок «State-space» цієї моделі задаємо МІМО-об'єкт. В блок «Uniform Noise Generator» задаємо матрицю нижнього значення вхідного сигналу([0 0 0]) та матрицю верхнього значення вхідного сигналу ([1 1 1]). Кількість параметрів генератора залежить від обраного об'єкту управління. В даному випадку, оскільки маємо об'єкт управління з 3 вхідними параметрами, записуємо відповідно наведені вище параметри блоку «Uniform Noise Generator».

Крок 5. Повторюємо крок 2-3 і створюємо 108 нейронних мереж з різними параметрами затриманих входів та виходів, а також різною кількістю нейронів у прихованому шарі. Для цього ми варіюємо вводимі патаметри поля «Розмер скритого слоя» від 5 до 16 нейронів, поля «Количество задержанных входов» від 0 до 2 входів, а також поля «Количество задержанных выходов» від 1 до 3 виходів.

Крок 6. Після того, як всі нейронні мережі створені, навчені та підставлені у модель обчислення сумарної середньоквадратичної помилки, запускаємо модель та отримуємо значення середньоквадратичної помилку у блоці «Display»

Табл. 1. Результати дослідження нейромережевих моделей

| $\begin{matrix} i \setminus j \\ n \end{matrix}$ | 0_1 | 0_2 | 0_3 | 1_1 | 1_2 | 1_3 | 2_1 | 2_2 | 2_3 |
|--|--------|---------|--------|----------|----------|----------|----------|----------|---------|
| 5 | 0.593 | 0.6415 | 0.663 | 0.02072 | 0.03368 | 0.01907 | 0.007993 | 0.03166 | 0.934 |
| 6 | 0.6336 | 0.565 | 0.6564 | 0.012 | 0.01389 | 0.05195 | 0.01004 | 0.878 | 0.04302 |
| 7 | 0.6814 | 0.912 | 0.6008 | 0.007073 | 0.02914 | 0.0373 | 0.01277 | 0.789 | 0.00856 |
| 8 | 0.7195 | 0.4416 | 0.8858 | 0.02093 | 0.02749 | 0.0159 | 0.01474 | 0.3705 | 0.06729 |
| 9 | 0.7148 | 0.5647 | 0.7339 | 0.01271 | 0.01793 | 0.01482 | 0.01554 | 0.02248 | 0.927 |
| 10 | 0.6226 | 0.899 | 0.3688 | 0.009068 | 0.01324 | 0.04971 | 0.01829 | 0.7073 | 0.956 |
| 11 | 0.675 | 0.5014 | 0.841 | 0.03134 | 0.01825 | 0.03142 | 0.07851 | 0.913 | 0.00784 |
| 12 | 0.5785 | 0.50435 | 0.7661 | 0.007977 | 0.003114 | 0.02115 | 0.004742 | 0.03676 | 0.03685 |
| 13 | 0.6581 | 0.936 | 0.856 | 0.003391 | 0.01355 | 0.01059 | 0.0186 | 0.3052 | 0.03426 |
| 14 | 0.783 | 0.3992 | 0.887 | 0.009263 | 0.009352 | 0.01895 | 0.02771 | 0.002146 | 0.878 |
| 15 | 0.981 | 0.731 | 0.734 | 0.01303 | 0.01212 | 0.05857 | 0.02063 | 0.02022 | 0.02057 |
| 16 | 0.837 | 0.6442 | 0.558 | 0.001715 | 0.001401 | 0.003874 | 0.005804 | 0.1045 | 0.1761 |

З таблиці 1 видно, що оптимальною структурою для даної моделі з 3 вхідними та 4 вихідними параметрами є структура з 16 нейронами у прихованому шарі, 1 затриманим входом та 2 затриманими виходами. При цьому отримано значення середньоквадратичної помилки 0.001401. Також у роботі було досліджено різні багатовимірні об'єкти 2x2, 4x2, 4x3, 3x3. Для системи 2x2 оптимальною структурою виявилась структура з 5 нейронами у прихованому шарі, 1 затриманим входом та 3 затриманими виходами (5-1-3). Середньоквадратична помилка при цьому становила 0.004196. Для моделі 4x2 – структура (5-2-3) з помилкою 0.001372,

моделі 4x3 – структура (16-1-2) з помилкою 0.001452, моделі 3x3 – структура (16-1-3) з помилкою 0.005575.

Висновок

У даній роботі розглядається комплексний формалізований підхід до реалізації багатоетапної процедури ідентифікації складних динамічних об'єктів з використанням нейромережевих модельних структур. Розроблено програмний пакет «МІМО-Plant» в середовищі MatLab, для оцінювання показників нейромережевих моделей МІМО-об'єктів управління, представлено його детальний опис та приклади дослідження.

Перелік посилань

1. Е.М. Воронов. Оптимизация многообъектных многокритериальных систем. – М.: Изд-во МГТУ, 2001. – 576 с.
2. Егупов Н.Д. Методы робастного, нейро-нечеткого и адаптивного управления. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. – 744 с.
3. Терехов В.А. Нейросетевые системы управления: Учеб. пособие для вузов. – М: Высшая школа. 2002. – 183с.
4. Cybenco G. Approximation by superposition's of a sigmoidal function // Math. Of Control, Signal and Systems. 1989. № 2. P. 303 – 314.
5. Funahashi K. On the approximate realization of continuous mapping by neural networks. Neural Networks. 1989. № 2. P. 182 – 192.
6. Barron A. R. Universal approximation bounds for superposition of a sigmoidal function // IEEE Trans. Inform. Theory. 1993. Vol. 39. P. 930 – 945.
7. Саймон Хайкин. Нейронные сети. Полный курс. – М.: Вильямс, 2006. – 1104с.
8. Сигеру Омату. Нейроуправление и его приложения. – М.: ИПРЖР, 2000. – 272с.
9. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И.Д. Рудинского. – М.: Горячая линия – Телеком, 2007. – 452 с.
10. Кравець П.І., Шимкович В.М., Зубенко Г.А. Технологія апаратно-програмної реалізації штучного нейрона та штучних нейронних мереж засобами FPGA / Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. – К.: Век+, – 2012. – №55. 174-180с.
11. Hornik K., Stinchcombe M., White H. Multilayer feedforward networks are universal approximators // Neural Networks. 1989. № 2. P. 359 – 366.
12. Гаврилов А.И. Перспективы применения нейросетевых технологий в системах автоматического управления // Вестник МГТУ им. Н.Э. Баумана. Приборостроение. - 1998. -№1.-С. 119-126.