



DOI: 10.5604/20830157.1176582

MODEL PROCESU SORTOWANIA OBIEKTÓW PRZY WYKORZYSTANIU PODEJŚCIA NEURONOWEGO

Jaroslav Lotysh

Łucki Narodowy Uniwersytet Techniczny, Wydział Automatycznego Sterowania Procesami Produkcyjnymi

Streszczenie. W artykule zaproponowane zostały imitacyjne komputerowe modele sortowania za pomocą normalnego trybu pracy oraz trybu neuronowego. Na podstawie modelu opracowano algorytm oraz otrzymane oprogramowanie, które implementuje system kontroli sortowania obiektów za pomocą wykorzystania podejścia neuronowego.

Słowa kluczowe: automatyczna kontrola, sieci neuronowe, modelowanie komputerowe

МОДЕЛЬ ПРОЦЕСУ СОРТУВАННЯ ОБ'ЄКТІВ З ВИКОРИСТАННЯМ НЕЙРОПІДХОДУ

Анотація. В роботі пропонуються імітаційні комп'ютерні моделі сортування зі звичайним режимом роботи та з режимом нейроуправління. На базі моделі розроблено алгоритм та отримано програмне забезпечення, яке реалізує систему управління сортуванням об'єктів з використанням нейропідходу.

Ключові слова: автоматичне управління, нейронні мережі, комп'ютерне моделювання

THE MODEL OF OBJECTS' SORTING PROCESS BY USING NEURO APPROACH

Abstract. Imitational sorting computer models with ordinary operating regime and with neurooperating regime are proposed in the article. On the basis of the model the algorithm is developed and the software is received, which realizes the system of sorting operating of the objects by using neuro approaches.

Keywords: automatic control, neural networks, computer simulation

Wstęp

Rozwój teorii sterowania przeszedł kilka etapów, a mianowicie, od rozwiązania równań różniczkowych (liniowych, nieliniowych) poprzez włączenie komputera w obwód sterowania – do użytku informatyki neuronowej w nasze czasy.

W obecnych czasach, sztuczne systemy neuronowe są powszechnie wykorzystywane do rozwiązywania różnych zadań inteligentnego sterowania, prognozowania w warunkach niepewności, nieliniowości oraz stochastyczności, ze względu na ich możliwości procesu pozyskiwania wiedzy z danych charakteryzujących funkcjonowanie obiektu badań.

Systemy te stanowią nowy interdyscyplinarny obszar badań, przybliżony do systemów inteligentnych, ponieważ w przeciwieństwie do systemów kontrolowanych rozwijających się na skrzyżowaniu cybernetyki, matematyki, elektroniki i innych nauk, systemy neuronowe są syntetyzowane przez informatykę neuronową, szczególnie przez sterowanie neuronowe.

Algorytmy sieci neuronowej różnią się od tradycyjnych wszechstronnością (w ramach określonej klasy problemów) i łatwością realizacji programu. Podczas procesu uczenia, sieć neuronowa automatycznie bierze pod uwagę właściwości konkretnego systemu sterowania bez konieczności jego matematycznego opisu. Ponadto sieci neuronowe mogą pozyskiwać wiedzę podczas eksploatacji, wykorzystując swoje doświadczenie sterowania.

1. Kluczowe wyniki badania

Przy opracowaniu modelu symulacyjnego brane jest pod uwagę sortowanie obiektów odbywające się w dwóch trybach:

- elastyczne ramie odbiornika reaguje na pojawienie się obiektu z jednego z trzech kierunków i wraca do pozycji neutralnej po zakończeniu operacji (normalne sortowanie);
- najpierw elastyczne ramie odbiornika znajduje się w położeniu neutralnym, a po pierwszej operacji sortowania staje w położeniu, w którym jest największe prawdopodobieństwo pojawienia się obiektu sortowania (sortowanie neuronowe).

Вступ

Розвиток теорії управління пройшов ряд етапів від рішення диференціальних рівнянь (лінійних, нелінійних) через включення комп'ютера в контур управління до використання нейроінформатики на сучасному етапі.

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.

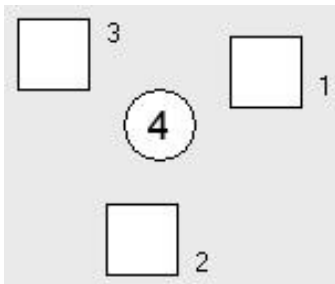
Такі системи представляють собою новий міждисциплінарний науковий напрям, близький до інтелектуальних систем, тому, що, на відміну від керованих систем, які розвиваються на стику кібернетики, математики, електроніки та інших наук, нейронні системи синтезуються з нейроінформатикою, зокрема з нейрокеруванням.

Нейромережеві алгоритми відрізняються від традиційних універсальністю (в межах певного класу задач) і простотою програмної реалізації. При навчанні нейромережа автоматично враховує особливості конкретної системи управління, не вимагаючи її математичного опису. Окрім того, нейронні мережі можуть доучуватися в процесі експлуатації, використовуючи свій досвід управління.

1. Основні результати дослідження

При розробці імітаційної моделі враховувалось, що сортування об'єктів відбувається в двох режимах:

- гнучкий приймач реагує на появу об'єкта з одного з трьох напрямків та повертається в нейтральне положення після завершення операції (звичайне сортування);
- спочатку гнучкий приймач знаходиться в нейтральному положенні та після першої операції сортування стає в положення, де найбільша ймовірність появи об'єкта сортування (сортування з нейроуправлінням).



Rys. 1. Wizualna reprezentacja pracy elastycznego ramienia odbiornika (widok z góry): 1, 2, 3 – pozycja możliwego złożenia obiektów (modelowane za pomocą przycisków), 4 – neutralne położenie odbiornika

Мал. 1. Візуальне представлення роботи гнучкого приймача (вид зверху): 1, 2, 3 – позиції можливої подачі об'єктів (модельуються кнопками), 4 – нейтральне положення приймача

Danymi wejściowymi jest początkowe położenie elastycznego ramienia odbiornika, stan przycisków na ekranie, które imitują ujawnienie się obiektu sortowania, a także tryb pracy modelu (bez neuro, neuro).

Jeśli model symulacyjny znajduje się w trybie "bez neuro" reakcją na dane wejściowe jest pochylenie elastycznego ramienia odbiornika w jedną z trzech stron, po czym powraca ono do swojej pierwotnej pozycji. Kierunek pochylenia będzie zależał od tego, który z przycisków został naciśnięty.

Jeśli model jest w trybie "neuro", odbywa się proces pozyskiwania wiedzy z baz danych. Po naciśnięciu przycisku odbywa się pochylenie. Następnie jest przetwarzanie danych na podstawie których określa się położenie elastycznego ramienia odbiornika. Przetwarzanie danych odbywa się przy użyciu wyuczonej sieci neuronowej, za pomocą której jest zdeterminowana reakcja modelu, co będzie miało wygląd wyszukiwania prawdopodobieństwa naciśnięcia jednego z przycisków oraz pochylenia elastycznego ramienia odbiornika

Podczas tworzenia trybu pracy modelu przy użyciu podejścia neuronowego zostały zastosowane jednokierunkowe sieci neuronowe.

Jednokierunkowe sieci neuronowe otrzymały swoją nazwę ze względu na zasadę pracy algorytmu wykorzystywaną przez sieć neuronową. W podanym algorytmie błąd przemieszcza się w kierunku przeciwnym od sygnału, to znaczy od warstwy wyjściowej do wejściowej.

Sieć neuronowa składa się z kilku warstw, które są ze sobą połączone, to znaczy że jednokierunkowa sieć neuronowa jest całkowicie połączona.

Uczenie sieci neuronowych polega na tym, że trzeba znaleźć funkcjonalny związek $Y = F(X)$, gdzie X – wektor wyjściowy, a Y – wejściowy. Takie zadanie ma wiele rozwiązań przy ograniczonej liczbie danych wejściowych. Aby ograniczyć przestrzeń poszukiwania podczas uczenia, należy zminimalizować funkcję celu błędów sieci neuronowej, która uzyskuje się za pomocą metody najmniejszych kwadratów:

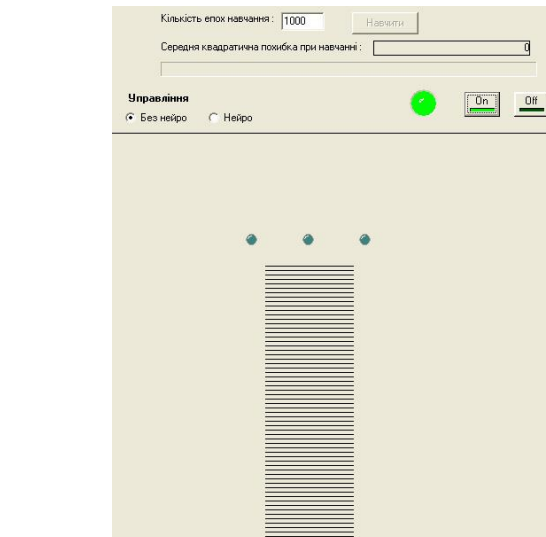
$$E(w) = \frac{1}{2} \sum_{j=1}^p (y_j - d_j)^2 \quad (1)$$

gdzie y_j – wartość j-tego wyjścia sieci neuronowej, d_j – wartość celu j-tego wyjścia, p – liczba neuronów w warstwie wyjściowej.

Uczenie odbywa się poprzez metodę gradientu. W każdej iteracji wagi zmieniają się według formuły:

$$\Delta w_{ij} = -\eta \cdot \frac{\partial E}{\partial w_{ij}} \quad (2)$$

gdzie η – parametr, który określa szybkość uczenia



Rys. 2. Podstawowa forma modelu symulacyjnego
Мал. 2. Основна форма імітаційної моделі

Vхідними даними є початкове положення гнучкого приймача, стан кнопок на екранній формі, які імітують появу об'єкта сортування, а також режим роботи моделі (без нейро, нейро).

Якщо імітаційна модель знаходиться в режимі «без нейро», то реакцією на вхідні дані буде нахил гнучкого приймача 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 нахил. Далі проходить обробка даних, на основі якої визначається положення гнучкого приймача. Обробка даних проходить з використанням навченої нейронної мережі, завдяки якій визначається реакція моделі, що матиме вигляд пошуку найбільшої ймовірності натискання однієї з кнопок та нахилу гнучкого приймача в певне положення.

При розробці режиму роботи моделі з нейронним підходом були використані нейронні мережі зворотнього поширення.

Мережі зворотнього поширення отримали свою назву завдяки принципу роботи алгоритму, за яким працює нейронна мережа. В даному алгоритмі помилка поширюється в протилежному напрямку від сигналу, тобто від вихідного шару до вхідного.

Дана нейронна мережа складається з декількох шарів, які є зв'язаними між собою, тобто нейронна мережа зворотнього поширення є повнозв'язною.

Навчання нейронної мережі полягає в тому, що потрібно знайти функціональну залежність $Y = F(X)$, де X – вхідний, а Y – вихідний вектори. В основному при обмеженій кількості вхідних даних така задача має безліч рішень. Для обмеження простору пошуку при навчанні ставиться завдання мінімізації цільової функції помилки нейронної мережі, яка знаходиться за методом найменших квадратів:

де y_j – значення j-го виходу нейромережі, d_j – цільове значення j-го виходу, p – кількість нейронів у вихідному шарі.

Навчання проходить методом градієнтного спуску. При кожній ітерації ваги змінюються за формулою:

де η – параметр, що визначає швидкість навчання.

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{dS_j} \cdot \frac{\partial S_j}{\partial w_{ij}} \quad (3)$$

gdzie y_j – wartość wyjścia j-tego neuronu, S_j – ważona suma sygnałów wejściowych. W tym mnożnik

де y_j – значення виходу j-го нейрона, S_j – зважена сума вхідних сигналів. При цьому множник

$$\frac{\partial S_j}{\partial w_{ij}} \equiv x_i \quad (4)$$

gdzie x_i – wartość wejścia i-tego neuronu.

де x_i – значення i-го входу нейрона.

Następnie przeanalizujemy definicję pierwszego mnożnika formuły (3)

Далі розглянемо визначення першого множника формули (3)

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{dS_k} \cdot \frac{\partial S_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{dy_k}{dS_k} \cdot w_{jk}^{(n+1)} \quad (5)$$

gdzie k – liczba neuronów w warstwie $n + 1$. Po wprowadzeniu zmiennej dodatkowej

де k – кількість нейронів у шарі $n + 1$. Після введення допоміжної змінної

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{dS_j} \quad (6)$$

Dalej zdefiniowana jest formuła rekurencyjna w celu określenia wartości warstwy n , w tym, wartość następnej warstwy ($n + 1$) jest wiadoma.

Далі визначається рекурсивна формула для визначення n -го шару, при тому, що є відомим наступний ($n + 1$) – ий шар.

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} \cdot w_{jk}^{(n+1)} \right] \cdot \frac{dy_j}{dS_j} \quad (7)$$

Ostatnia warstwa sieci neuronowych oblicza się za pomocą docelowego wektora, czyli wektora tych wartości, które sieć powinna podawać przy określonym zbiorze wartości wejściowych.

Останній шар нейронної мережі знаходиться завдяки наявності цільового вектора, тобто вектора тих значень, які повинна видавати мережа при даному наборі вхідних значень.

$$\delta_j^{(N)} = (y_i^{(N)} - d_i) \cdot \frac{dy_i}{dS_i} \quad (8)$$

Formuła (2) w postaci rozwiniętej:

Формула (2) в розкритому вигляді:

$$\Delta w_{ij}^{(n)} = -\eta \cdot \delta_j^{(n)} \cdot x_i^n \quad (9)$$

Kompletny algorytm uczenia sieci neuronowej:

Повний алгоритм навчання нейромережі:

1. Podać na wejście neuronowej sieci jeden z potrzebnych obrazów i określić wartość wyjść sieci neuronowych.
2. Obliczyć wartość wyjściowej warstwy sieci neuronowej za pomocą formuły (8) oraz obliczyć zmianę wagi wyjściowej warstwy N według formuły (9). Również na tym etapie jest przedstawienie wektorów w kolejności losowej z bazy danych.
3. Obliczyć według formuł (7) i (9) odpowiednio $\Delta w_{ij}^{(N)}$ dla innych warstw sieci neuronowej, $n = N-1 \dots 1$.
4. Skorygować wszystkie wagi sieci neuronowej

1. Подати на вхід нейронної мережі один із потрібних образів і визначити значення виходів нейронів мережі.
2. Розрахувати значення вихідного шару нейронної мережі за формулою (8) і розрахувати зміни ваг вихідного шару N за формулою (9). Також на даному етапі проводиться представлення векторів у випадковому порядку з навчальної вибірки.
3. Розрахувати за формулами (7) і (9) відповідно $\Delta w_{ij}^{(N)}$ для інших шарів нейронної мережі, $n = N-1 \dots 1$.
4. Скорегувати всі ваги нейронної мережі.

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t)$$

5. Jeżeli błąd jest znaczący przejść do kroku 1 [12].

5. Якщо помилка istotna, то перейти на крок 1 [12].

Programowa realizacja otrzymanego algorytmu wykonana jest za pomocą oprogramowania Delphi z wykorzystaniem biblioteki danych NeuralBase [11].

Програмна реалізація отриманого алгоритму здійснена на мові програмування Delphi з використанням бібліотеки NeuralBase [11].

Biblioteka danych NeuralBase jest przeznaczona do programowej realizacji sieci neuronowych. Biblioteka ta tworzy następujące dwa główne typy sieci neuronowych: wielowarstwową sieć neuronową i sieć Hopfilda (rekurencyjna sieć neuronowa), które uczą się z algorytmu zwrotnego rozprzestrzeniania się błędów.

Бібліотека NeuralBase призначена для програмної реалізації нейронних мереж. Дана бібліотека створює такі два основні типи нейронних мереж, як багатoshарова нейронна мережа та мережа Хопфілда (рекурентна нейронна мережа), які навчаються за алгоритмом зворотнього поширення помилки.

Wymieniona biblioteka danych może być łatwo wbudowana do innych programów. Jest to realizowane poprzez dostępność otwartego kodu i zasadę tworzenia sieci neuronowej w postaci komponentów.

Дана бібліотека легко вбудовується в інші програми. Це реалізовано завдяки наявності відкритого коду та принципу побудови нейронної мережі у вигляді компонентів.

Podczas realizacji programowej po naciśnięciu na "Naucz", liczba okresów kształcenia jest przypisywana do zmiennej prbEpoch.Max.

При програмній реалізації, після натиску на кнопку «Навчити», змінній prbEpoch.Max присвоюється кількість епох навчання.

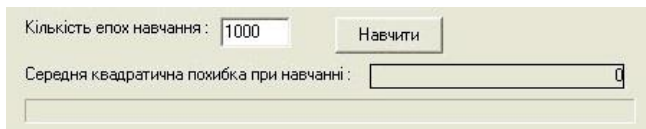


Рис. 3. Уczenie sieci neuronowej

Мал. 3. Навчання нейронної мережі

Po uruchomieniu tej procedury aktywuje się procedurę NeuralNetBPEpochPassed dla wizualizacji procesu uczenia. Proces uczenia odbywa się z wykorzystaniem funkcji NeuralNetBP.TeachOffLine biblioteki NeuralBase.

Po kliknięciu na jeden z trzech przycisków model symulacyjny realizuje nachylenie elastycznego ramienia odbiornika w odpowiednie położenie i przywraca go do położenia początkowego.

Następnie oprogramowanie przypisuje dostępnej zmiennej wartość przycisku, który został naciśnięty oraz wysyła ją do głównej procedury przetwarzania danych. W zależności od wysłanej wartości zmiennej odbywa się konfiguracja wektora wejściowego.

Następnie skonfigurowany wektor wejściowy jest podawany na wejście sieci neuronowej. Uzyskane wartości z wektora wyjściowego są rejestrowane w wyjściowej tablicy Mas_imov.

Położenie elastycznego ramienia odbiornika jest określone przez funkcję position(), do której są wysyłane dane z tablicy Mas_imov. Uzyskane wyniki są zapisywane w zmiennej R1.

Zmienne R1 oraz R0 zawierają wartości: największe prawdopodobieństwo występowania obiektu sortowania (R1) oraz poprzednie położenie mechanizmu elastycznego ramienia (R0).

Procedura Mechanism odzwierciedla pozycję, w której będzie znajdowało się elastyczne ramię odbiornika, czekając na następny obiekt.

Aby przetestować otrzymany model, został stworzony zestaw do fizycznego modelowania procesu sortowania. Zestaw ten składa się z trzech części: sprzętu oraz systemów sterowania niższego i wyższego poziomu.

Sprzęt obejmuje platformę programistyczną Arduino, czujniki sortowania i urządzenie wykonawcze (trzy serwomechanizmy oraz odbiornik elastyczny).

Arduino – to wygodne narzędzie do szybkiego rozwoju urządzeń elektronicznych, z którego mogą korzystać zarówno amatorzy, jak i profesjonaliści. Platforma ta jest popularna na całym świecie ze względu na jej użyteczność i łatwość oprogramowania. Równie ważnym czynnikiem jest to, że architektura i kod programowania ma charakter otwarty. Urządzenie jest programowane przez USB bez wykorzystania programatorów.

Urządzenia zaprojektowane w oparciu o Arduino, mogą pozyskiwać informacje z różnych czujników oraz mogą być podłączone do różnych urządzeń poprzez port USB i inne kanały komunikacji (Bluetooth, Wi-Fi). Za pomocą tej platformy możliwe jest zarządzanie różnymi mechanizmami wykonawczymi.

Mikrokontroler jest zaprogramowany w języku Arduino (oparty o Writing) oraz przez środowisko opracowania Arduino (oparte o środowisko Processing). Projekty urządzeń, opracowane w oparciu o Arduino, mogą pracować samodzielnie lub współpracować z oprogramowaniem komputerowym. Wszystkie dane na temat platformy programistycznej Arduino i wszystko, co jest z nią związane, jest otwarte dla użytkowników. Elementy składowe oraz dodatkowe platformy rozszerzeń są szeroko rozpowszechnione w świecie. Schematy platform programistycznych oraz kod oprogramowania typowych modeli można znaleźć zarówno na oficjalnej stronie firmy Arduino, jak i analogicznych stronach krajowych. Praca z taką informacją jest łatwa i wygodna.

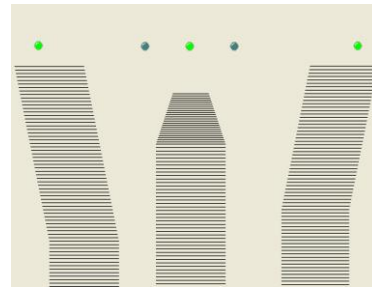


Рис. 4. Моżliwe pozycje elastycznego ramienia odbiornika w oczekiwaniu obiektu

Мал. 4. Можливі позиції гнучкого приймача при очікуванні об'єкта

Під час запуску цієї процедури активізується процедура NeuralNetBPEpochPassed для відображення процесу навчання. Процес навчання проходить завдяки використанню функції NeuralNetBP.TeachOffLine бібліотеки NeuralBase.

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

Потім програма присвоїть біжучій змінній значення кнопки, яка була натиснута, та відправить його в процедуру основної обробки даних. Залежно від того, яке саме значення біжучої змінної було відправлене, налаштується вхідний вектор.

Далі сформований вхідний вектор подається на вхід нейронної мережі. Отримані значення з вихідного вектора записуються у вихідний масив Mas_imov.

Позиція гнучкого приймача визначається завдяки функції position(), в яку відправляються дані з масиву Mas_imov. Отримані результати записуються у змінну R1.

Змінні R1 та R0 відображають значення, де найбільша ймовірність появи об'єкта сортування (R1) та попереднє положення гнучкого приводу (R0).

Процедура Mechanism відображає положення, в якому буде перебувати гнучкий приймач, очікуючи наступний.

Для перевірки отриманої моделі створено програмно-апаратний комплекс фізичного моделювання процесу сортування. Програмно-апаратний комплекс сортування об'єктів складається з трьох частин: апаратної частини, систем управління нижнього та високого рівня.

В склад апаратної частини системи сортування входять плата Arduino, давачі та виконавчий механізм (три сервоприводи та гнучкий приймач).

Arduino – це зручний інструмент для швидкої розробки електронних пристроїв, який можуть використовувати як новачки, так і професіонали. Платформа популярна в усьому світі завдяки своїй зручності у користуванні і простоті мови програмування. Також вагомим фактором є те, що вся архітектура та програмний код знаходяться у відкритому доступі. Пристрій програмується через USB без використання програматорів.

Пристрій, розроблений на базі Arduino, можуть отримувати інформацію від різноманітних давачів, можуть з'єднуватись з різноманітними пристроями як через USB, так і іншими способами (Bluetooth, WiFi). За допомогою даної платформи ми маємо змогу управляти різними виконавчими механізмами.

Мікроконтролер програмується завдяки мові Arduino (заснований на мові Wiring) і середовищу розробки Arduino (заснованому на базі середовища Processing). Проекти пристроїв, розроблених на базі Arduino, можуть працювати самостійно, або ж взаємодіяти з програмним забезпеченням комп'ютера. Всі дані по Arduino і все, що з ним пов'язано, є відкритим для загального доступу. Складові частини платформи та додаткові плати розширення широко розповсюджені у світі. Схеми плат та програмний код типових програм можна знайти як на офіційних сайтах фірми Arduino, так і на вітчизняних аналогах. Працювати з даною інформацією дуже просто та зручно.

Platforma Arduino składa się z: mikrokontrolera Atmel AVR (ATmega328, ATmega168 oraz ATmega8 w nowych i starych wersjach) oraz elementów komunikacji do programowania i połączenia z innymi systemami. Wiele platform zawiera liniowy stabilizator napięcia +5 V lub +3,3 V. Taktowanie odbywa się przez rezonator kwarcowy o częstotliwości 16 lub 8 MHz. Wewnętrzny programator nie jest potrzebny, ponieważ w mikrokontrolerze jest preinstalowane oprogramowanie BootLoader.

Na poziomie koncepcyjnym, wszystkie platformy są programowane poprzez złącze RS-232, ale realizacja tej metody jest różna. Platforma Serial Arduino zapewnia prosty obwód do konwersji poziomów sygnału RS-232 na poziom TTL. Potoki sygnałów wysyłane są do platform, na przykład do Diecimila, programowanej przez port USB, co jest realizowane przez układ konwertera USB-to-FTDI FT232R. W wersji platformy Arduino Uno jako konwerter występuje mikrokontroler ATmega8U2.

Jako czujnik informujący o obecności obiektów sortowania w układzie używany jest czujnik podczerwieni Sharp GP2Y0A21, ponieważ ma on czas reakcji wynoszący 38 ± 10 ms, co odpowiada postawionym wymaganiom oraz ma napięcie 4,5–5,5 V i jest oferowany w niskiej cenie.

Jedną z głównych funkcji w zestawie oprogramowania pełni mechanizm wykonawczy, który wykorzystuje serwomechanizm. Serwomechanizm to układ, w którym sterowanie odbywa się poprzez ujemne sprzężenie zwrotne, co z kolei pozwala dokładnie kontrolować parametry ruchu.

Serwomechanizm składa się z:

1. Dowolnego rodzaju mechanizmu mechanicznego.
2. Czujnika – do śledzenia położenia, prędkości czy przyspieszenia itp.
3. Bloku sterowania mechanizmem – jest to system mechanicznych cięgien lub obwód elektroniczny, który automatycznie utrzymuje wymagane parametry na czujniku zgodnie z górną określoną wartością (znanej wartości zwrotnej).

Dla danego zadania wykorzystano serwomechanizm Micro 9g Servo ze względu na jego kompaktowe wymiary $22 \times 12 \times 29$ mm, zasilanie 3–7,2 V oraz moment wału 1,5 kg/cm.

Dla realizacji systemu sterowania niskiego poziomu został opracowany algorytm pracy urządzenia sortowania w oparciu o mikrokontroler Arduino.

Danymi wejściowymi jest informacja o obecności obiektu sortowania. Uzyskanie tych danych odbywa się poprzez trzy czujniki. Jeśli w otoczeniu czujnika pojawi się obiekt, z mikrokontrolera przesyłane jest polecenie, które obraca mechanizm wykonawczy (serwomechanizm) i ustawia elastyczne ramię odbiornika w odpowiedniej pozycji do przyjęcia obiektu. Mikrokontroler wysyła również informacje o pojawieniu się obiektu do systemu wysokiego poziomu.

Dla wdrożenia programowego wykorzystywane jest oprogramowanie Wiring, które jest przystosowane do pracy z serwomechanizmem i wybranymi czujnikami.

Wraz z opracowaniem systemu sterowania niskiego poziomu został opracowany system sterowania o poziomie wysokim. Jego dane wejściowe to początkowe położenie mechanizmu elastycznego oraz dane dotyczące pojawienia się obiektu otrzymane z ComPorta.

Aby zacząć pracę niezbędne są przykłady dla zbioru danych trenujących. Przy następnej operacji, za pomocą zbioru danych trenujących odbywa się uczenie sieci neuronowej. Po wielokrotnym przetwarzaniu dużej ilości przykładów, system zrównoważy współczynnik wagi, który jest gotowy do użycia.

Reakcją na dane o zadziałaniu jednego z czujników, które nadchodzą z ComPort, jest tworzenie się wektora wejściowego w sieci neuronowej oraz uzyskanie wektora wyjściowego, który wskazuje na prawdopodobieństwo wystąpienia obiektu. Proces zapisu do wektora wejściowego i uzyskania wyjściowego dla każdego czujnika odbywa się oddzielnie.

W skład płyty Arduino wchodzi: mikrokontroler Atmel AVR (ATmega328 i ATmega168 w nowych wersjach i ATmega8 w starych), a także elementy złącza dla programowania ta z'ednany z innymi schemami. Na bogatych płytach przysutni lniyni stabilizator napruy +5 V abo +3,3 V. Taktuwnni zdiysnuється na chstoty 16 abo 8 MGc kwarcowym rezonatorom. Zowniynni programator ne potreben, tomu, cho u mikrokontroleri poperedny proshivaється zavanthauvach BootLoader.

Na koncepcyjnym poziomie wszystkie płyty programują się przez RS-232, ale realizacja danego sposobu różni się. Płyta Serial Arduino posiada prostą invertującą schemę dla konwersji poziomów sygnałów RS-232 do poziomów TTL. Aktualne sygnały przesyłane są na płyty, na przykład Diecimila, programują się przez USB, co realizowane jest przez układ konwertera USB-to-Serial FTDI FT232R. W wersji platformy Arduino Uno jako konwerter wykorzystuje mikrokontroler ATmega8U2.

Jako czujnik informujący o obecności obiektów sortowania w programno-aparatnym kompleksie wykorzystano IЧ-давач Sharp GP2Y0A21, oскільки він має час реакції 38 ± 10 ms, що відповідає поставленим вимогам, напругу 4,5–5,5 V, а також низьку ціну.

Одна з основних функцій в програмно-апаратному комплексі належить виконавчому механізму, який базується на сервоприводах.

Сервопривід – це привід, управління якого здійснюється через негативний зворотний зв'язок, що в свою чергу дозволяє точно керувати параметрами руху.

В склад сервоприводу входить:

1. Будь-який механічний привід.
2. Давач – для відслідковування положення, швидкості, зусилля і т.д.
3. Блок керування приводом – це механічна система тяг чи електронна схема, яка автоматично підтримує необхідні параметри на давачі згідно наперед заданому значенню (заданому значенню повороту).

Для даної розробки використано сервопривід Micro 9g Servo, оскільки його малогабаритні розміри $22 \times 12 \times 29$ мм, напруга 3–7,2 V, момент на валу 1,5 кг/см.

Для реалізації системи управління низького рівня розроблений алгоритм роботи сортувального пристрою на базі мікроконтролера Arduino.

Вхідними даними є інформація про наявність об'єкта сортування. Отримання цих даних здійснюється завдяки трьом давачам. Якщо в зоні давача з'являється об'єкт, тоді з мікроконтролера відправляється команда, яка повертає виконавчий механізм (сервопривід) та встановлює гнучкий приймач у відповідне положення для прийняття об'єкта. Мікроконтролер також відправляє інформацію про появу об'єкта на систему високого рівня.

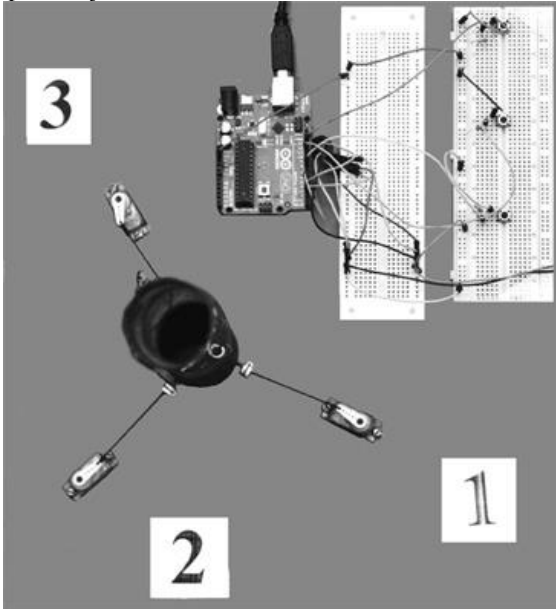
Для програмної реалізації застосовується мова програмування Wiring, яка пристосована для роботи з сервоприводами та обраними давачами.

Після розробки системи управління низького рівня була розроблена система управління високого рівня. Вхідними даними є початкове положення гнучкого механізму та дані про появу об'єкта, отримані з ComPorta.

Для початку роботи необхідні приклади для навчальної вибірки. В подальшому за допомогою навчальної вибірки проводиться навчання нейронної мережі. Після того, як система вказує n-ну кількість разів здійснити перебір прикладів, вона урівноважує вагові коефіцієнти та стає готовою до роботи.

Реакцією на дані про спрацювання одного з давачів, що надходять через ComPort, є формування вхідного вектора нейронної мережі та отримання вихідного вектора про ймовірність появи об'єкта. Процес запису у вхідний вектор та отримання вихідного проходить для кожного давача окремо.

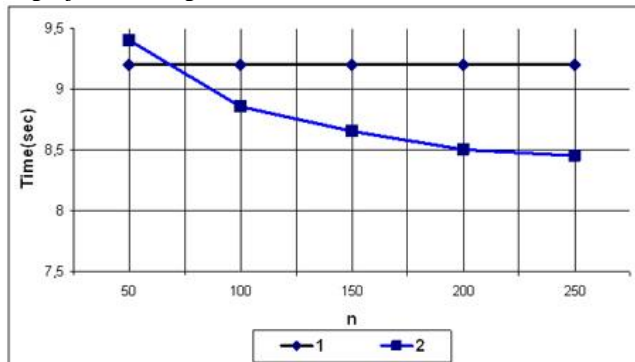
Dane tablicy są wysyłane do funkcji, która określa położenie obrotów mechanizmów wykonawczych. Funkcja ta jest oparta na wyborze największego prawdopodobieństwa występowania obiektu w obszarze jednego z czujników. System niskiego poziomu, po przetworzeniu przez system sterowania wysokiego poziomu, otrzymuje odpowiednią reakcję w postaci liczby, która konfiguruje mechanizmy wykonawcze do nowej pozycji lub pozostawia go w poprzedniej.



Rys. 5. Model fizyczny zestawu do sortowania obiektów: 1, 2, 3 – serwomechanizmy (widok z góry)

Мал. 5. Фізична модель програмно-апаратного комплексу сортування об'єктів: 1, 2, 3 – сервоприводи (видок зверху)

W celu przetestowania wpływu elementu sterowania neuronowego na szybkość sortowania przeprowadzono szereg eksperymentalnych prac układu zarówno w trybie sterowania neuronowego, jak i bez niego.



Rys. 7. Zależność czasu reakcji od liczby obiektów sortowania z wykorzystaniem zestawu oprogramowania: 1 – zwykłe sortowanie, 2 – sortowanie z podejściem neuronowym

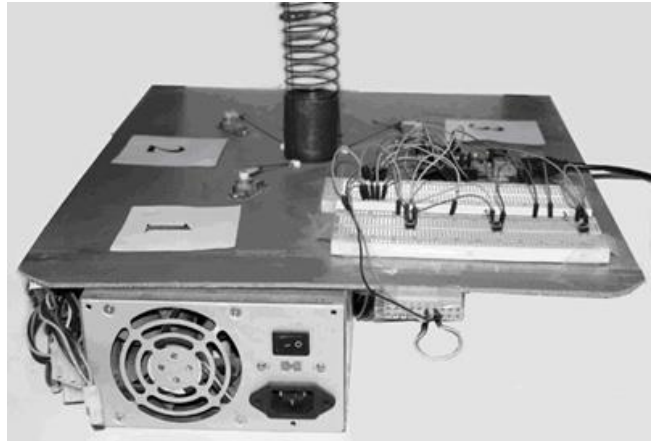
Мал. 7. Залежність часу реакції від кількості об'єктів сортування з використанням програмно-апаратний комплексу: 1 – звичайне сортування, 2 – сортування з нейроуправлінням

Wykres z rys. 7 pokazuje, że przy małej liczbie obiektów sortowania, użycie sterowania neuronowego nie jest skuteczne, ale przy liczbie obiektów większej niż 75 stosowanie neuronowe jest uzasadnione.

Wykres na rys. 8 wskazuje, że komputerowy model symulacyjny odpowiednio opisuje proces sortowania i może być stosowany do badania systemu neuronowego.

Масив даних відправляється у функцію, яка визначає позицію повороту виконавчих механізмів. Дана функція основана на виборі найбільшої ймовірності появи об'єкта в зоні одного з датчиків.

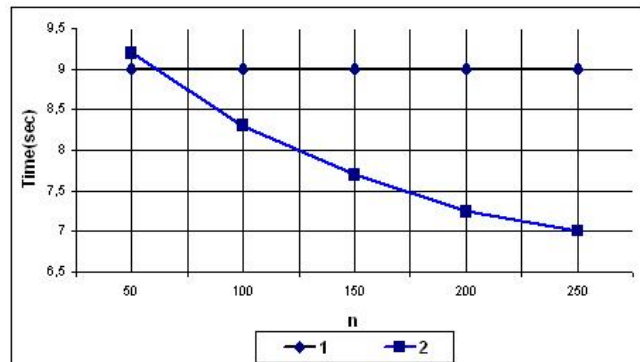
Система низького рівня, після обробки програмою системи управління високого рівня, отримує реакцію у вигляді числа, яке переналаштовує виконавчі механізми в нове положення чи залишає його у попередньому.



Rys. 6. Model fizyczny zestawu do sortowania obiektów (widok z boku)

Мал. 6. Фізична модель програмно-апаратного комплексу сортування об'єктів (видок збоку)

Для перевірки впливу елемента нейроуправління на швидкість сортування проведено ряд експериментів роботи комплексу як в режимі нейроуправління, так і без нього.



Rys. 8. Zależność czasu reakcji od liczby obiektów sortowania za pomocą komputerowego modelu symulacyjnego: 1 – zwykłe sortowanie, 2 – sortowanie sieci neuronowej

Мал. 8. Залежність часу реакції від кількості об'єктів сортування з використанням комп'ютерно-імітаційної моделі: 1 – звичайне сортування, 2 – сортування з нейроуправлінням

З представлених графіків (мал. 7) видно, що при малій кількості об'єктів сортування застосування нейроуправління не оправдане, але при кількості об'єктів більше 75 використання нейроуправління є оправданим.

З графіків на рис. 8. видно, що комп'ютерно-імітаційна модель адекватно описує процес сортування і може застосовуватись для вивчення системи нейроуправління.

2. Wnioski

Otrzymano komputerowy model symulacyjny odpowiednio opisuje proces sortowania, zarówno w trybie normalnym i neuronowym. Opracowane oprogramowanie niskiego poziomu pracuje z elementem neuronowym (element sterowania wysokiego poziomu). Zastosowanie podejścia neuronowego jest uzasadnione przy większej liczbie elementów sortowania niż 75.

Podziękowania

Praca powstała w ramach projektu *PL-NTU Transgraniczna wymiana doświadczeń* PBU.03.01.00-06-386/11-00 współfinansowanego w ramach Programu Współpracy Transgranicznej Polska – Białoruś – Ukraina 2007–2013 finansowanego ze środków Unii Europejskiej w ramach Europejskiego Instrumentu Sąsiedztwa i Partnerstwa.

Niniejsza publikacja została stworzona przy pomocy Unii Europejskiej. Wyłączną odpowiedzialność za zawartość niniejszej publikacji ponosi Jaroslav Lotysz oraz w żaden sposób nie może być ona postrzegana jako odzwierciedlenie poglądów Unii Europejskiej.

Literatura || Література

- [1] Baskanova T. F., Lankyn Yu. P.: Neurosetevye alhorytmy samostoyatel'noy adaptatsyy. Sbornyk nauchnykh trudov: Vserossyyskaya nauchno-tekhnicheskaya konferentsyya "Neuroynformatyka-99". CH. 1, 1999, 17–24.
- [2] Dorohov A. Yu., Alekseev A. A.: Matematycheskiye modeli bystrykh neyronnykh setey. V sb. nauchn. tr. "Systemy upravlenyya y obrabotky ynformatsyy". Vyp. 490, 1996.
- [3] Gill P. E., Murray W., Wright M. H.: Practical Optimization, Academic Press, New York 1981.
- [4] Hopfield J.: Neural networks and physical systems with emergent collective computational abilities, Proceedings of National Academy of Sciences, vol. 79 no. 8, 2554–2558.
- [5] Loskutov A. YU., Mykhaylov A. S.: Vvedeniye v synerhetyku, Nauka, Moskva 1990.
- [6] Mkrtychyan C. O.: Neyrony y neyronnye sety (Vvedeniye v teoriyyu formal'nykh neyronov), Énerhiya, Moskva 1971.
- [7] Muller B., Reinhardt J.: Neural Networks. An introduction, Springer-Verlag, Berlin 1991.
- [8] Surovtsev Y. S., Klyukyn V. Y., Pyvovarova R. P.: Neyronnye sety, VHU, Voronezh 1994.

Mgr Jaroslav Lotysz
e-mail: jaropost@gmail.com

Jaroslav Lotysz urodził się w 1992 roku w Łucku. W 2009 roku dostał się na studia do Narodowego Uniwersytetu Technicznego w Łucku, specjalność "Automatyczne Sterowanie Procesami Technicznymi". Magistrant Wydziału Automatycznego Sterowania Procesami Produkcyjnymi na Narodowym Uniwersytecie Technicznym w Łucku. Otrzymał stypendium Prezydenta Ukrainy oraz stypendium Funduszy Charytatywnych "Igora Pałyci". Posiada doświadczenie w zakresie zintegrowanej komputerowej technologii, zwłaszcza w opracowaniach sprzętu i oprogramowania opartych na aplikacji mikrokontrolera Arduino i platformy Android. Zainteresowania badawcze obejmują komputerowe i symulacyjne modelowanie, oprogramowanie.



Магістр Ярослав Лотиш
e-mail: jaropost@gmail.com

Лотиш Ярослав Володимирович народився в 1992 році в місті Луцьку. В 2009 році поступив до Луцького національного технічного університету, на спеціальність «Автоматизоване управління технологічними процесами». Магістр кафедри автоматизованого управління виробничими процесами Луцького національного технічного університету. Стендіат академічної стипендії Президента України та степендії благодійного фонду «Фонд Ігоря Палиці». Має досвід роботи в галузі комп'ютерно-інтегрованих технологій, зокрема в розробці програмно-апаратних комплексів на базі мікроконтролера ARDUINO та додатків на платформі Android. Наукові інтереси: комп'ютерне та імітаційне моделювання, програмування.

otrzymano/отримано/received: 11.03.2015

przyjęto do druku/прийнято до друку/accepted: 27.08.2015

2. Висновок

Отримана комп'ютерно-імітаційна модель адекватно описує процес сортування як у звичайному режимі, так і в режимі нейроуправління. Розроблене програмне забезпечення низкого рівня узгоджено працює з нейроелементом (елемент управління високого рівня). Застосування нейроуправління є оправданим при кількості елементів сортування більше 75.

Завдяки

Робота створена в рамках проекту PL-NTU Прикордонного обміну досвідом PBU.03.01.00-06-386/11-00, фінансованих в рамках Програми Транскордонного Співробітництва Польща – Білорусь – Україна 2007–2013 фінансується Європейським Союзом в рамках Європейського Інструменту Суспільства та Партнерства.

Ця публікація була створена за допомогою Європейського Союзу. Відповідальність за зміст цієї публікації лежить на Ярославові Лотишові, і жодним чином не може розглядатися як відображення поглядів Європейського Союзу.

- [9] Vol'kenshteyn M. B.: Byofizyka: Ucheb. rukovodstvo, Nauka, Moskva 1988.
- [10] Wasseman P. D.: Neural computing: theory and practice, Van Nostrand Reinhold Co., New York 1989.
- [11] Analysis of cryptographic algorithms on neural networks: http://scsp.org.ua/files/spring/2010/Olga_Provotar-Analysis_of_cryptographic_algorithms_on_neural_networks.pdf
- [12] BaseGroup Labs: <http://www.basegroup.ru/>
- [13] Vykypedyya. Neyronnaya set' Khopfylda: 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
- [14] Yspol'zovaniye alhorytma obratnoho rasprostraneniya oshybky: http://sernam.ru/book_gen.php?id=69
- [15] Neyronnaya set' Khopfylda y ee pryimeneniye: <http://iasa.org.ua/lections/tpr/neuro/hopfield.htm>