**УДК 681.3, 004.855.5**

**Скоробрещук С.Ю.**
Національний університет "Львівська політехніка"
кафедра електронних обчислювальних машин

# Автопілот на основі нейронної мережі



**Розглянуто реалізацію апаратної та програмної складових самокерованої моделі на базі нейронної мережі. Проведено навчання за допомогою експертної вибірки та перевірку працездатності конкретної реалізації.**

**Ключові слова: нейронні мережі, автопілот.**

**The implementation of hardware and software components of the self - controlled model on the basis of the neural network was considered. The training was conducted with the help of expert sampling and verification of the feasibility of specific implementation.**

**Keywords: neural network, autopilot.**

**Вступ.** Сьогодні досить гостро поставлено питання щодо автоматизації праці людини. Машини замінюють як важку фізичну працю так і рутинну. Працюють замість людей там, де потрібна надточність та відповідальність. Провідні фірми світу такі як Tesla, Google та багато інших розробляють власні самокеровані автомобілі. Все це в майбутньому має звільнити людей від напруженого кермування, оптимізувати трафік та зменшити кількість нещасних випадків. [1].

В сфері самокерованих авто здійснено чимало відкриттів та напрацювань. Проте, одним із лімітуючих факторів є наявність на дорогах загального користування звичайних людей-водіїв. Якщо придивитися до їхнього стилю керування, то одразу помітно, що будь-яка людина відхиляється від правил в тій чи іншій мірі. Тому на даний момент неможливо створити автопілот, який буде здатен протидіяти непердбачуваній поведінці інших людей на дорозі. І саме тому чим швидше буде втілена в життя модель що здатна з високою точністю пересуватися по місту – тим скоріше весь транспорт стане автономним та звільнить людство від більшості проблем великих міст, які існують на даний момент.

**Стан проблеми.** Сьогодні сфера самокерованого транспорту переживає неймовірний підйом. Це пов’язано з розвитком технологій захоплення зображення, накопиченням та аналізу великої кількості інформації (Big Data) та методів навчання штучного інтелекту [2-8]. Серед провідних компаній які мають серйозні напрацювання в даній сфері можна виділити Tesla, Google, Nissan, BMW, Baidu та багато інших. Компанія Tesla має найбільший прогрес в цій галузі серед інших компаній. Вона вже продемонструвала завершений автономний прототип який з легкістю пересувається дорогами загального призначення в США.

Чи не найважливішу роль виконує покриття сенсорами та датчиками. У випадку автопілотів Tesla, сумарно 8 камер забезпечують огляд на 360 градусів навколо машини на протязі 250 метрів [1]. 12 сучасних ультразвукових датчиків доповнюють цю картину та дозволяють розпізнавати тверді та мякі обєкти вдвічі краще ніж дозволяють оптичні методи камер. Передній радар з пришвидшеною обробкою забезпечує збір додаткових даних про світ на надлишковій довжині хвилі, яка може бачити через сильний дощ, туман, пил і навіть крізь машину попереду. Саме така різноманітність технічних давачів інформації дозволяє автопілоту компанії виконувати ті функції, що він наразі виконує. Проте в даній роботі розглянуто спосіб реалізації автопілоту лише з використанням однієї камери, що як зменшує собівартість проекту, так і наближує дану систему датчиків до тих, що використовує людина-водій (лише зір та слух). Оскільки начання подібних автопілотів проводиться здебільшого на основі експертної системи, коли нейронна мережа “запам’ятовує” як реагує людина на дорожню ситуацію та потім повторює ці дії в схожих випадках, то схожі умови для машини та людини можуть покращити навчання та “розуміння” дорожньої ситуації.

**Постановка задачі.** Розробити апаратно-програмну систему автономного керування транспортним засобом на спеціальному треку. Реалізувати систему автоматичного збору

тренувальних даних під час керування людиною для навчання нейронної мережі. Реалізувати модель автомобіля та протестувати її ефективність.

**Розв'язання задачі.** Реалізація складається з 2 основних частин — програмної та апаратної. Для апаратної частини було використано мінімальний набір найбільш доступних компонентів, який дозволить виконати поставлене завдання. На рисунку 2 наведено функціональну схему взаємодії апаратної частини моделі самокерованого транспортного засобу, що складається з: блоку портативного живлення (LiPo акумулятор ємністю 5200 мАгод [9]), блоку керування сигналами двигунів (4 H-мости L293D), двигунів, блоку обробки та керування (Raspberry Pi 3B+) та камери.

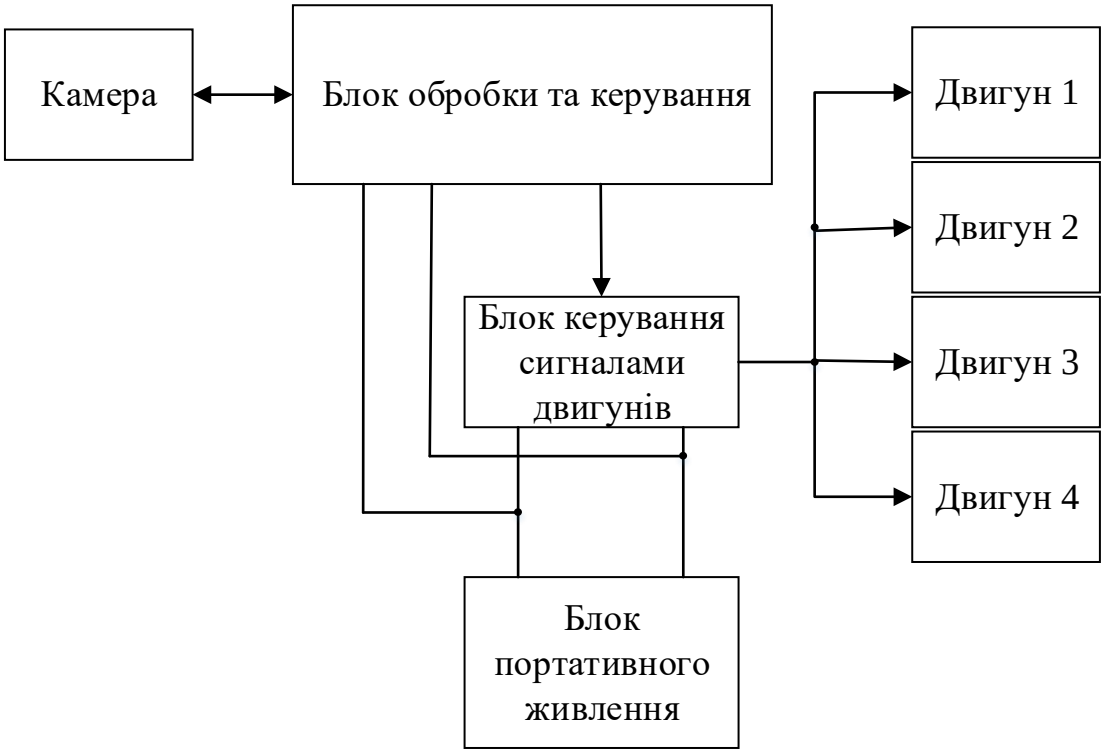


Рис. 1. Функціональна схема взаємодії апаратної частини моделі самокерованого транспортного засобу.

Для реалізації програмної частини — було розроблено програмну систему, структурна схема якої показана на Рис. 2. Основним програмним модулем є менеджер циклу подій, який виконує функції ініціалізації всіх інших модулів, запуск в них функції опитування/оновлення, керує вхідними та вихідними даними всіх модулів, контролює частоту обробки та має функціонал, що дозволяє запускати режим тренування нейронної мережі. У всіх модулях, що працюють в режимі автопілоту є функції оновлення та ініціалізації. Під час режиму тренування нейронної мережі – модуль автопілоту відключений. Контролер камери видає зображення за зображенням під час виклику функції оновлення, менеджер видає це зображення користувачу на веб сервер, користувач задає кут повороту керма відповідно до отриманого зображення, далі модуль роботи з даними зберігає отримані записи. Після того як було зібрано певну кількість таких тренувальних даних – модуль навчання нейронної мережі опрацьовує їх всі та зберігає навчену модель нейронної мережі для модуля автопілоту. Під час безпосередньо режиму автопілоту – контролер камери так само зберігає зображення, проте передає його не лише на веб сервер, а й на модуль автопілоту, який за поточним зображенням видає кут повороту керма та швидкість руху, ці сигнали потрапляють на контролери двигунів, які перетворюють логічні сигнали у силові та приводять модель транспортного засобу у рух.

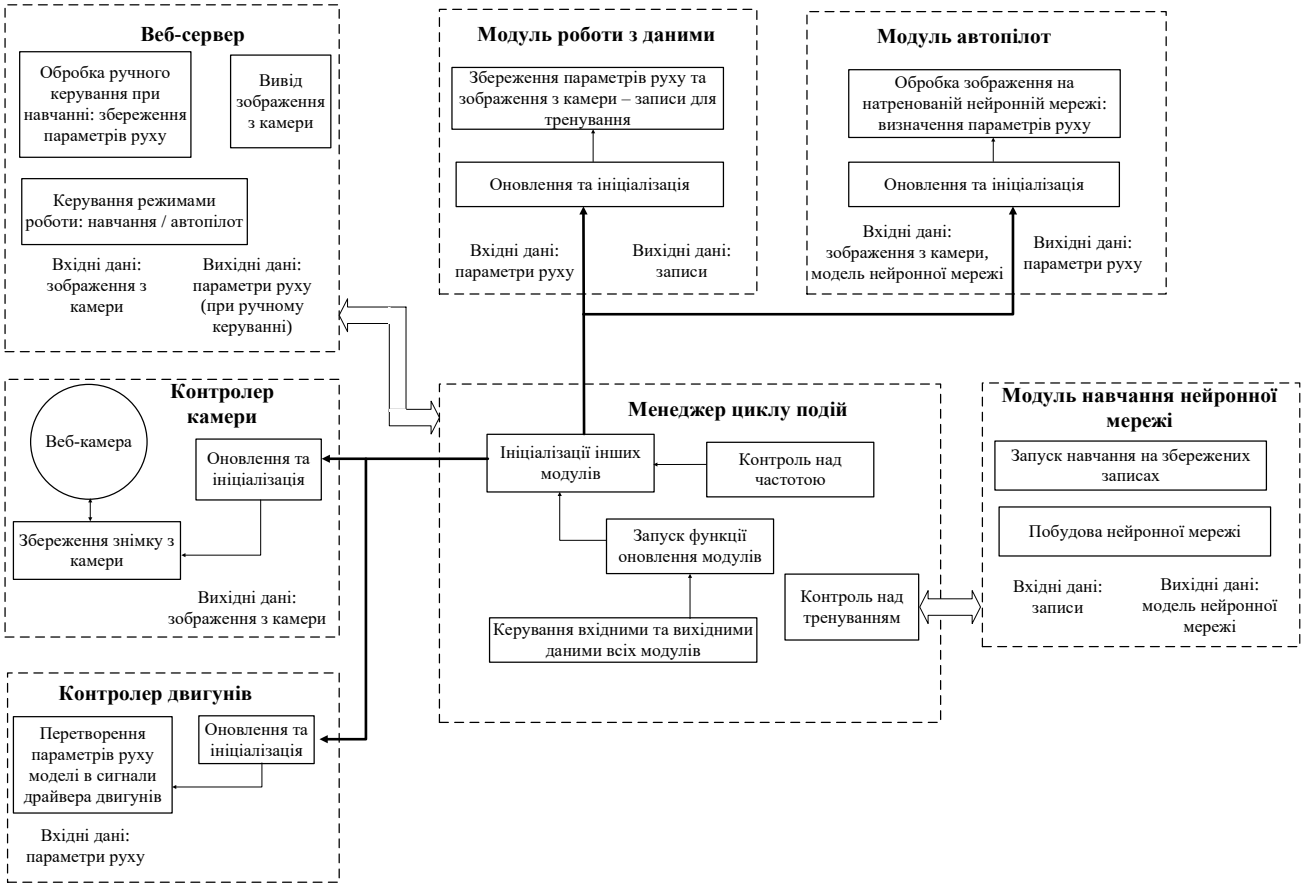


Рис. 2. Структурна схема програмної системи автопілоту на основі нейронної мережі

Найчастіше на Raspberry Pi встановлюють Linux. Існує безліч дистрибутивів, які повністю портовані під архітектуру ARM, що дозволяє використовувати всі можливості плати при низькому енергоспоживанні. Ще одним беззаперечною перевагою є те, що в Linux дуже зручно керувати сигналами на низькому рівні, подаючи їх на вихід з плати через GPIO.

Щодо мови програмування – для обраної ОС найкраще підходить мова програмування Python [10]. Вона надає можливість як низькорівневого контролю над сигналами які виходять до органів управління, так і дуже зручний API для побудови нейронної мережі та складних алогритмів. Python також дуже розповсюджений в суспільстві людей, які цікавляться розробкою під Raspberry Pi, чи взагалі на Linux.

Для побудови структури нейронної мережі використовується бібліотека від Google – Tensor Flow, а саме її реалізація зі зручним API – бібліотека Keras [3]. Keras – API для побудови нейронної мережі на високому рівні, що здатна працювати поверх TensorFlow, CNTK чи Theano. Вона була розроблена з прицілом на швидке розгортання експериментів на її основі.

На процесорі Intel® Core™ i7-8570H процес навчання нейронної мережі зайняв 2,5 години та закінчився через 45 епох, оскільки покращення значення похибки не відбувалося протягом 5 епох поспіль. Та середньоквадратична похибка в кінці навчання склала 0,0586.

Наступним кроком було перенесення моделі (mypilot) утвореної в результаті навчання на робочій станції на пам’ять Raspberry Pi, яка відповідає за керування зменшеною моделлю транспортного засобу по імпровізованому треку. Після тестування навченої моделі можна виділити місця, які складні для її самостійного подолання та провести навчання на цих місцях, додавши їх до отриманих раніше вхідних даних та заново перенавчити мережу.

В результаті експериментів та спостереження було виявлено, що дана система без ніяких покращень дуже чутлива до якості фото, до освітленості треку та до умов, в яких вона була навчена. І, хоча, в теорії підхід описаний в цій роботі підходить для абсолютно всіх умов, будь-якого набору елементів та модулів системи – необхідно приділити велику кількість часу на збір правильних даних, їх підготовку та обробку, також можна застосувати різні типи фільтрацій для вхідних зображень, щоб зменшити вплив освітленості та шумів.

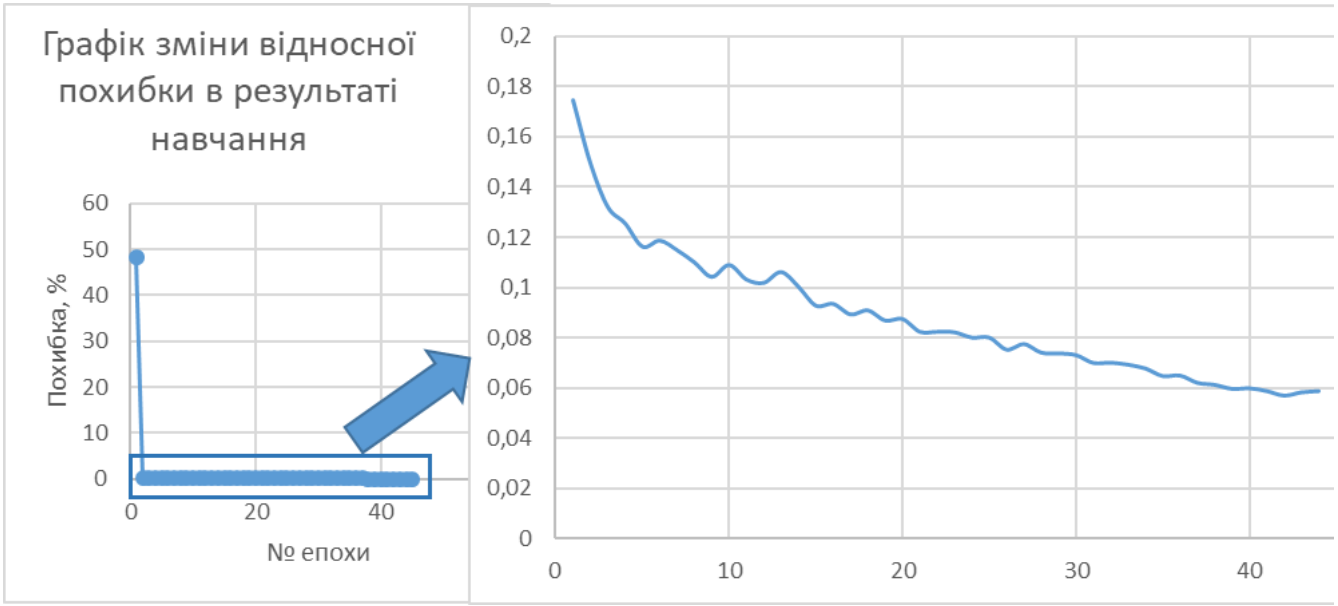


Рис. 3. Графічне представлення зміни відносної похибки нейронної мережі в процесі її навчання

**Висновки.** У даній роботі було розроблено програмно-апаратну систему автономного управління транспортним засобом. Було зібрано тестову модель, побудовано найпростіший трек для тестування, написано програмне забезпечення яке дозволяє легко збирати вхідні дані для навчання, обробляти їх, будувати модель нейронної мережі та врешті-решт керувати розробленим транспортним засобом. Проведено тестування та проаналізовано отриманий результат шляхом фіксації середньоквадратичної похибки,

**Література**

1.Tesla Vision [Електронний ресурс]. – Режим доступу https://www.tesla.com/about (дата звернення 19.06.2019 р.). – Назва з екрана.

2.Штучні нейронні мережі : обчислення / М.А. Новотарський, Б.Б. Нестеренко // Праці Інституту математики НАН України. – Т50. – Київ: Ін-т математики НАН України, 2004. – 408 с.

3.TensorFlow для глибокого навчання / Бхарат Рамсундар, Реза Босаг Заде// БХВ-Петербург, 2016 – 256с.

4.Reconfigurable Cellular Neural Networks and Their Applications [Текст] / Müstak E. Yalçin, Tuba Ayhan, Ramazan Yeniçeri // Springer International Publishing; April 2019 – 416 p.

5.Practical Convolutional Neural Networks [Текст] : Pradeep Pujari, Md. Rezaul Karim, Mohit Sewak, Packt Publishing 2018. – 102 p.

6.Haykin S.S. Neural networks [Текст] / S.S. Hayking – Hamilton: Pearson Education, 2009. – 938 p.

7.Deng L. Deep learning: Methods and applications / Deng L. and Yu D. // Foundations and Trends in Signal Processing, 7(3–4) – 2014. – pp. 197–387.

8.What is Big Data [Електронний ресурс]. – Режим доступу: https://rb.ru/howto/chto-takoe-big-data/ (дата звернення 19.06.2019 р.). – Назва з екрана.

9.Силова електроніка. Керівництво користувача / Кит Сукер// Додека XXI. - 2008. - с. 43, 119-126, 216-218.

10.Python Essential Reference (4th ed.) [Текст] / David Beazley – Pearson Education, 2009. – 415 p.