

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»
Ройтлінгенський університет техніки та економіки (Німеччина)
Еслінгенський університет прикладних наук (Німеччина)
Технічний університет Фрайберзька гірничо-академія (Німеччина)
Університет Кобленц-Ландау (Німеччина)
Краківська гірничо-металургійна академія (Польща)
Вроцлавський технічний університет (Польща)
Дніпропетровський національний університет імені Олеся Гончара
ДКХ «Дніпровський машинобудівний завод»
Міжнародна науково-промислова корпорація «ВЕСТА»
ДАТ «КБ Дніпровське»

ПРОБЛЕМИ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ОСВІТІ, НАУЦІ ТА ПРОМИСЛОВОСТІ

XV МІЖНАРОДНА КОНФЕРЕНЦІЯ

м. Дніпро
15 – 17 грудня 2020 року

Збірник наукових праць
№ 5

Дніпро
НТУ «ДП»
2021

УДК 622 (06)

П 78

Редакційна колегія:

Г.Г. Півняк, О.Б. Іванов, М.О. Алексєєв, В.В. Ткачов, В.І. Корнієнко, В.В. Гнатушенко, В.В. Слесарєв, Г. Грюллер, Н. Нойбергер, Л.І. Мещеряков, Б.І. Мороз, А. Дерен, Я. Сконечний, І.М. Удовик, М.І. Стадник, О.С. Шевцова.

Проблеми використання інформаційних технологій в освіті, науці та промисловості : XV міжнар. конф. (15– 17 грудня 2020 р.) : зб. наук. пр. / ред. кол.: Г.Г. Півняк та ін.; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2021. – № 5. – 198 с.

ISBN 978-966-350-760-6

Подано результати теоретичних та експериментальних досліджень з різних аспектів використання інформаційних технологій в освіті, науці та управлінні промисловістю. У публікаціях розглянуто питання створення та вдосконалення програмних засобів обробки та передачі інформації, математичного моделювання, дистанційної освіти, інформаційної безпеки та телекомунікації.

Для наукових, інженерно-технічних співробітників і студентів, які спеціалізуються в галузі обчислювальної техніки та інформаційних технологій.

УДК 622 (06)

ISBN 978-966-350-760-6

НТУ «Дніпровська політехніка», 2021

РОЗДІЛ 1

МІЖНАРОДНЕ СПІВРОБІТНИЦТВО У СФЕРІ ОСВІТИ, НАУКИ І ВИРОБНИЦТВА

УДК 004.415.3:681.6

М.О. Алексєєв¹, І.М. Удовик¹, М.М. Дмитровська¹, Т. Поврожнік²

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

²Технічний університет «Краківська гірничо-металургійна академія
ім. Станіслава Сташиця» (AGH), м. Краків, Польща

МЕТОДИ, АЛГОРИТМИ ТА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПОКРАЩЕННЯ ПРОЦЕСІВ ПЕРЕДАЧІ ДАНИХ РАДІОСИГНАЛАМИ

Анотація. Описано процес покращення процесів передачі даних радіосигналами в інформаційній системі бездротової мережі, яка згодом набула також практичного застосування для користувачів. Покращення досягається за рахунок використання математичних методів для збільшення необхідної пропускної здатності сигналу та збільшення смуги пропускання каналу приймача сигналів.

Ключові слова: *інформаційний сигнал, бездротова мережа, радіохвилі, сенсори, пакетування даних, фільтрування даних, технологія LoRa, веб-додаток, Angular.*

Вступ. Стрімким ривком сучасного розвитку інформаційних технологій, безумовно, стало використання бездротових мереж для передачі інформації до користувачів. Основними перевагами таких мереж є зручність, достатньо велика відстань передачі сигналу, низький рівень споживання енергії у кінцевих пристроїв, велика проникаюча здатність радіосигналів у міській забудові, тощо. У роботі найбільшу увагу приділено використанню технології модуляції LoRa, основна мета якої полягає у забезпеченні збільшення дальності зв'язку, стійкості сигналу та чутливості приймачів, у порівнянні з відомими аналогами технології. Для вирішення задач, поставлених для виконання цілі кваліфікаційної роботи були використані основні принципи покращення процесів передачі даних радіосигналами вищезгаданої технології модуляції, розроблено веб-додаток користувача для практичного застосування проведеного дослідження.

Попри все, різноманітні технології вдосконалення роботи бездротових мереж тільки набирають обертів свого розвитку та покращення у застосуванні, що стирає ліміти для розробників їх апаратного та програмного забезпечення у постановці цілей для майбутніх можливостей збільшення якісних характеристик таких мереж для розробки оптимізованих модерних інформаційних систем.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані та вирішені такі завдання:

- викласти принципи проектування та оптимізації процесів передачі даних радіосигналами у сфері застосування бездротових мереж;
- визначити основні, мінімально необхідні компоненти системи;
- спроектувати та розробити програмну частину бездротової мережі на базі технології LoRaWan, шляхом аналізу та залучення найкращих методів даної технології для покращення процесів передачі даних радіосигналами;
- спроектувати користувацький веб-застосунок для впровадження та практичного застосування розроблених методів;
- зробити висновки щодо доцільності створення подібної системи.

Основний зміст роботи. Під час роботи були використані наступні методи та інструменти.

- математичні методи розширення спектру теореми Шеннона-Хартлі;
- математична модель процесів фільтрування даних, плинності частоти сигналу;
- TypeScript, Slim, Docker, VSCode, Angular, PHP, C++, HyperText Markup Language.

Бездротовий зв'язок - це електромагнітна передача інформації між двома або більше точками, які пов'язані електричним провідником. Найбільш поширені бездротові технології використовують радіохвилі. Він охоплює різні типи фіксованих, мобільних та портативних додатків, включаючи двосторонній радіозв'язок, стільникові телефони, персональні комп'ютери (ПК) і відповідно бездротові мережі.

Енергоєфективна мережа далекого радіусу дії (LPWAN) або широкополосна мережа низької потужності (LPWA) або мережа низького енергоспоживання (LPN) – це тип бездротової телекомунікаційної ширококутної мережі, призначений для забезпечення зв'язку на великих діапазонах при низькому бітній швидкості серед підключених предметів, таких як датчики, що працюють від акумулятора. Низька потужність, низька швидкість передачі даних та передбачуване використання відрізняють цей тип мережі від бездротової глобальної мережі, призначеної для підключення користувачів та передачі більшої кількості даних, використовуючи більше енергії. Швидкість передачі даних LPWAN коливається від 0,3 кбіт/с до 50 кбіт/с на канал.

На фізичному рівні принципу передачі даних за технологією LPWAN лежить властивість радіосистем – збільшення енергетики, а значить і дальності зв'язку при зменшенні швидкості передачі. Чим нижче бітова швидкість передачі, тим більше енергії вкладається в кожен біт і тим легше виділити його на тлі мережевих перешкод в приймальні частини системи. Таким чином, низька швидкість передачі даних дозволяє відповідно домогтися більшої дальності їх прийому.

Пристрій або модем з LPWAN-модулем передає дані по радіоканалу на базову станцію. Станція приймає сигнали від всіх пристроїв в радіусі своєї дії,

оцифровує і передає на віддалений сервер, використовуючи доступний канал зв'язку: Ethernet, стільниковий зв'язок, VSAT.

Отримані на сервері дані використовуються для відображення, аналізу, побудови звітів та прийняття обумовлених рішень.

Управління пристроями, оновлення програмного забезпечення відбувається з використанням зворотного каналу зв'язку.

Технологія модуляції LoRa (Long Range) являє собою метод модуляції, який забезпечує значно більшу дальність зв'язку (зону покриття), ніж інші конкуруючі з ним способи. Метод ґрунтується на технології модуляції з розширеним спектром і варіації лінійної частотної модуляції (Chirp Spread Spectrum, CSS) з інтегрованою прямою корекцією помилок (Forward Error Correction, FEC). Технологія LoRa значно підвищує чутливість приймача і, аналогічно іншим методам модуляції з розширеним спектром, використовує всю ширину смуги пропускання каналу для передачі сигналу, що робить його стійким до каналним шумів і нечутливим до зсувів, викликаних неточностями в налаштуванні частот при використанні недорогих опорних кварцових резонаторів. Технологія LoRa дозволяє здійснювати демодуляцію сигналів з рівнями на 19,5 дБ нижче рівня шумів, притому що для правильної демодуляції більшості систем з частотної маніпуляцією (Frequency Shift Keying, FSK) потрібна потужність сигналу як мінімум на 8-10 дБ вище рівня шуму.

Важливе місце у наданні змоги передання даних від технічних засобів (сенсорів) до користувача займає правильний підбір програмних засобів та їх функціоналів для вирішення поставлених задач. У зв'язку з чим, в процесі виконання завдання кваліфікаційної роботи було використано наступні програмні засоби: мова програмування PHP на базі фреймворків Angular та Slim, платформа для розробки API Postman, інструмент для візуального проектування баз даних MySQL Workbench 8.0., Apache веб-сервер 2.4.46., мова програмування C як база для написання технічних алгоритмів, HTML та CSS як допоміжні інструменти для створення інтерфейсу користувача.

Для проведення розробки та дослідження інформаційної системи і покращення процесів передачі даних, необхідно використовувати інструменти, які дозволяють маніпулювати об'єктами "Запит" та "Відповідь". Ідея проміжного програмного забезпечення фреймворку Slim чудово допомагає з цією метою, так як і з захистом програмних засобів для міжсайтових запитів та їх аутентифікації перед запуском запитів. Створюйте програми за допомогою проміжного програмного забезпечення, налаштовуйте HTTP-запити та об'єкти відповіді навколо програми Slim. Наступний принцип розкриває сутність роботи Slim: він додає проміжне програмне забезпечення як шари навколо

основного додатка. Після цього структура відповідно розширюється, додаючи нові проміжні шари.

Після отримання необхідного пакету даних, постає вимога використання правильного підходу у процесі фільтрації отриманої інформації. Структура побайтового фільтрування («відбілювання») даних базується на 9-бітному поліномі $x^9 + x^5 + 1$ (рис. 1).

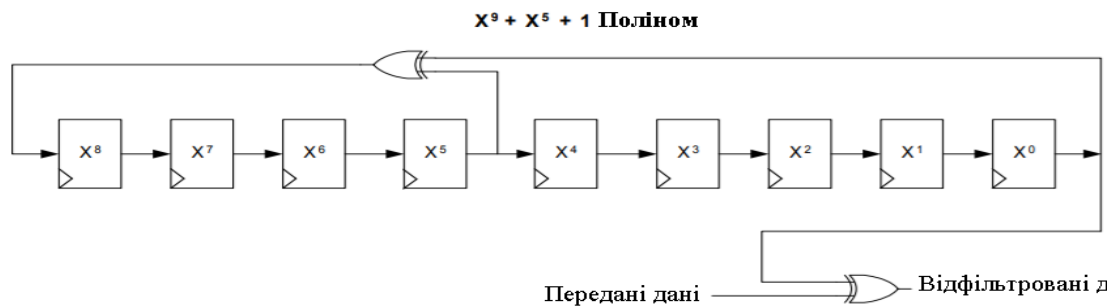


Рис. 1. Структура побайтового фільтрування даних

На відміну від попереднього алгоритму, алгоритм побітового фільтрування даних базується на роботі з пакетами інформації, розподілених побітово. Ці процеси дуже подібні, але послідовність зберігання результату зовсім інша, тому виникає необхідність правильної реалізації належного алгоритму, що використовується для коректного фільтрування даних і подальшої передачі користувачу (рис. 2).

На стороні приймача відбувається зворотній процес, і виникає необхідність «відбілити» пакет під час передачі інформації, щоб його довжина пакета могла бути отримана з вхідних даних. Тут важливо зауважити, що пристрій не знає розміру пакета, який він збирається отримати, інакше готове корисне навантаження ніколи не згенеруються.

Оскільки приймач не володіє інформацією про довжину пакету, необхідно встановити поріг рівня потоку інформації досить низьким, щоб генерувалося принаймні одне переривання заданого рівня пропуску даних, і було можливо «реверсивно відбілити» весь пакет даних, починаючи з підраховування їх кількості, поки не буде отримано усі.

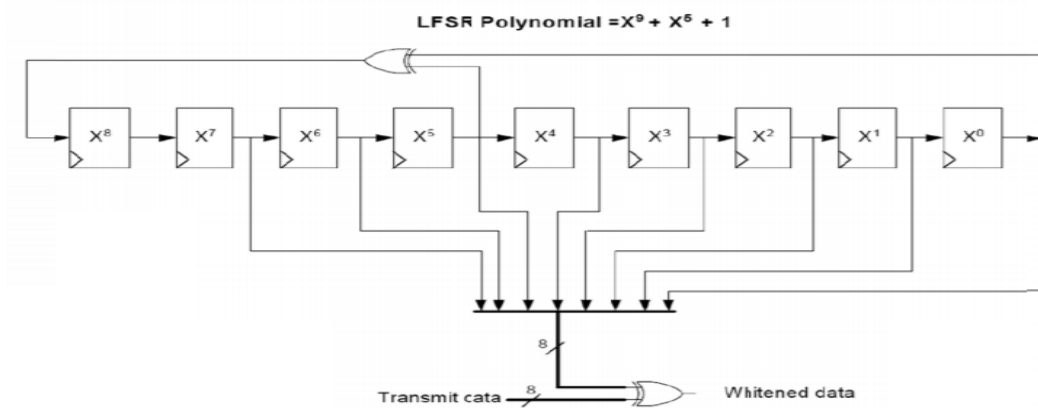


Рис. 2. Структура побітового фільтрування даних

Після використання алгоритмів фільтрування даних, у ході вдосконалення процесів передачі пакетів інформації радіосигналами, впроваджується використання принципу розширення спектру, в основу якого покладено теорему Шеннона–Хартлі, що встановлює максимальну швидкість передачі інформації через канал зв'язку заданої смуги пропускання в присутності шуму. Теорема встановлює пропускну здатність каналу Шеннона для лінії зв'язку та визначає максимальну швидкість передачі даних (інформації), яка може передаватися в межах заданої смуги пропускання за наявності перешкод і визначається за формулою:

$$C = B * \log_2 \left(1 + \frac{S}{N} \right), \quad (1)$$

де C – ємність каналу (біт/с), B – пропускну здатність каналу (Гц), S – середня потужність отриманого сигналу (Вт), N – середня потужність шуму (Вт).

Після підставлення значення натурального логарифму з основою 2 та наступних перетворень рівняння (1), стає зрозуміло, що для підвищення швидкості передачі пакетів інформації в каналі з фіксованим відношенням рівня перешкод до сигналу, потрібно збільшувати лише пропускну здатність переданого сигналу. Як було зазначено вище, збільшуючи смугу пропускання сигналу, можна компенсувати погіршення відношення «сигнал/шум» (або «шум/сигнал») радіоканалу.

У традиційних системах розширеного спектру прямого розподілу початкова фаза (фаза передавання) передавача змінюється відповідно до послідовності кодів. Зазвичай цей процес досягається множенням сигналу даних, що передаються, на розширювальний код. Таким чином, передача певної послідовності коротших кодованих бітів відбувається набагато швидше, ніж цілий сигнал даних, поширюється смуга пропускання сигналу за вихідну смугу, зайняту лише вихідним сигналом. Усі ці дії спрямовані на підвищення швидкості передачі цілих пакетів даних, що є метою виконання роботи. Наступний рисунок ілюструє принцип розширення спектру (рис.3.).

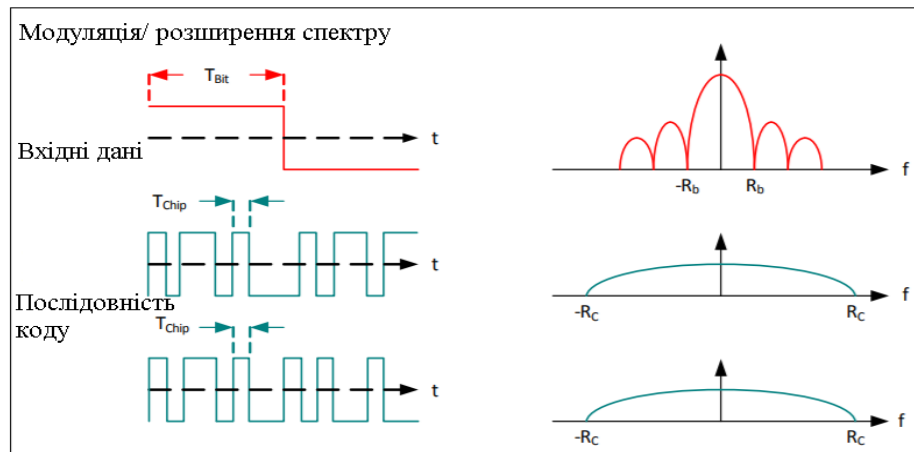


Рис. 3. Процес модуляції/розширення спектру

Після чого, у ході виконання експерименту чотири вузькосмугові сигнали було пропущено через широкосмуговий сигнал. Однак тривалість вузькосмугових сигналів така, що у часовій області період перешкод є коротким у відношенні до широкосмугового сигналу. Завдяки надмірності, асоційованій із широкосмуговою модуляцією широкого спектра, модуляція є досить стійкою до механізму перешкод, який виглядає як короткочасні імпульси. Цю тенденцію відображає наступний графік (рис. 4.).

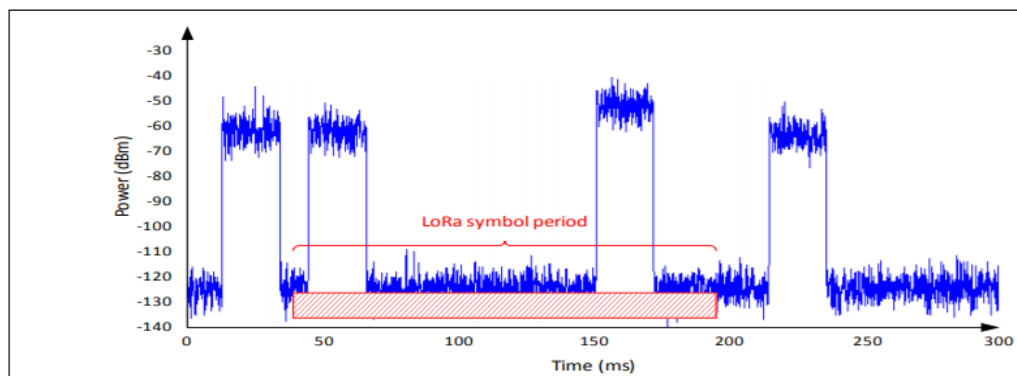


Рис. 4. Візуалізація підвищення стійкості сигналів до серійних перешкод

У якості прикладу практичного застосування мережі сенсорів з покращеним алгоритмом передачі даних за допомогою радіосигналів на базі технології LoRa (у порівнянні з відомими аналогами, сенсорами з вузькосмуговою маніпуляцією зі зміщенням частоти) в непростих міських умовах, розглянуті базові дані, що передаються з магнітних та температурних сенсорів з паркувальних місць реального паркувального комплексу, розташованого у м. Рим (Італія).

Після проведення дослідження, розробки та впровадження алгоритму було покращення процесу передачі інформації за допомогою радіосигналів через приймачі на базі технології LoRa, з метою систематизації, обробки та передачі даних до користувача з подальшою практичною експлуатацією.

Наукова новизна: досліджено та удосконалено алгоритми процесів підвищення швидкості передачі даних радіосигналами; отримані методи мають подальший розвиток та практичне застосування у вигляді інформаційної системи, яка збирає, контролює, упорядковує та передає дані до користувача.

Висновки. В результаті вивчено та досліджено методи передачі радіосигналів за допомогою технології LoRa та мови програмування PHP на базі фреймворків Slim та Angular. Досягнуто поставленої мети, яка полягала у дослідженні та вдосконаленні методів підвищення швидкості передачі та обробки даних з магнітних та температурних сенсорів за допомогою технології LoRa. Поставленої цілі було досягнуто за рахунок впровадження покращення в структуру вже існуючих алгоритмів фільтрації та передачі даних за рахунок розширення приймаючого каналу математичним методом, що дозволяє приймання більшої кількості пакетів інформації, і прискорює швидкість процесу передачі даних. Разом з цим збільшено також місткість та чутливість каналів приймання інформації, що дозволило досягти покращення стійкості приймачів до мережових перешкод. У ході експерименту було також проведено порівняння показників з відомим аналогом пристрою. Використовуючи створені покращення, було розроблено інформаційну систему для систематизації процесів передачі даних до користувача у вигляді веб та мобільних додатків, що мають також практичне використання.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Ullo, Silvia Liberata; Sinha, G. R. (2020-05-31). "Advances in Smart Environment Monitoring Systems Using IoT and Sensors". *Sensors* (Basel, Switzerland). 20 (11): 3113. doi:10.3390/s20113113. ISSN 1424-8220. PMC 7309034. PMID 32486411. Bertier Luyt, Samuel N. Bernier, Tatiana Reinhard *Design for 3D Printing: Scanning, Creating, Editing, Remixing, and Making in Three Dimensions / Make Community, LLC—New York, 2015.—162с.*
2. Xia, Feng; Tian, Yu-Chu; Li, Yanjun; Sun, Youxian (2007-10-09). "Wireless Sensor/Actuator Network Design for Mobile Control Applications". *Sensors* (Basel, Switzerland). 7 (10): 2157–2173. doi:10.3390/s7102157. ISSN 1424-8220. PMC 3864515. PMID 28903220. Dr. Dan-Andrei Marinescu *How to build a 3D Printer: DIY project: "EASY CoreXY 3D Printer Model 350" / LM Publishing House – Volendam, 2019 – 87 с.*
3. Semtech Application Note AN1200.13, "SX1272/3/6/7/8: LoRa Modem Designer's Guide".
4. FCC Office of Engineering and Technology Laboratory Division "Guidance for Performing Compliance Measurements on Digital Transmission Systems (DTS) Operating".
5. OFCOM, "Use of Short Range Devices alongside mobile broadband services operating in the 800MHz band".

ПЕРСПЕКТИВИ РОЗВИТКУ МІЖНАРОДНОЇ СПІВПРАЦІ ВНЗ З МАЛИМ ТА СЕРЕДНІМ БІЗНЕСОМ

Анотація. Описано можливості поширення сфери міжнародної співпраці випускаючих кафедр вищих навчальних закладів (ВНЗ) з зарубіжними роботодавцями за рахунок надання інформаційних послуг та стимулювання умов співпраці з представниками малого і середнього бізнесу. Досягається це за рахунок реструктуризації умов функціонування традиційної системи «РОБОТОДАВЕЦЬ – ПРАЦІВНИК» у «РОБОТОДАВЕЦЬ – ВНЗ – ПРАЦІВНИК» шляхом заохочення до співпраці на малих підприємствах зарубіжжя здобувачів вищої освіти спеціальностей 121 «Інженерія програмного забезпечення» та 122 «Комп'ютерні науки» та розповсюдження їх компетентностей на вирішення проблемних питань сьогодення в сферах міжнародної діяльності малого і середнього бізнесу.

Ключові слова: міжнародна співпраця ВНЗ, Інженерія програмного забезпечення, Комп'ютерні науки, GeoBit.

Вступ. В умовах сьогодення відомі зарубіжні компанії та концерни формують свій кадровий персонал через співпрацю з науковими центрами та ВНЗ. Стейкхолдери співпрацюють з ВНЗ у всіх країнах і надають студентам випускаючих кафедр, ще при їх навчанні, ті компетентності, які будуть необхідні їм для роботи саме в тій чи іншій сфері. Великі компанії будь якої країни мають умови для наукової співпраці з ВНЗ і, завдяки спільним проектам із студентами, відбирають найкращих випускників для працевлаштування їх на своїх підприємствах за заздалегідь сформованим напрямом діяльності. Досвід наукової співпраці з іноземними галузевими фірмами показує, що серед підприємців середнього та малого бізнесу зарубіжжя поширюється позитивна тенденція співпраці з представниками наукових шкіл і випускниками ВНЗ України за технічними та інформаційними напрямами діяльності. Зараз при багатьох ВНЗ створені бізнес-інкубатори, наукові центри та центри розвитку бізнесу куди кожен підприємець може звернутися зі своїми ідеями або для отримання кваліфікованої консультації чи допомоги для вирішення проблемних питань. Конкуренція в сферах діяльності середнього та малого бізнесу, спонукає їх представників брати участь у міжнародних виставках, форумах та наукових конференціях. В той же час, в умовах цифровізації, однією з сучасних задач будь якого ВНЗ, є пошук нових міжнародних контактів з метою отримання і поширення світового досвіду впровадження цифрових технологій в сфери професійної діяльності своїх випускників [1]. Тому перспективи міжнародної співпраці ВНЗ, що готують здобувачів вищої освіти за

спеціальностями 121 «Інженерія програмного забезпечення» та 122 «Комп'ютерні науки», з малим та середнім бізнесом інших країн є актуальною темою сьогодення.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані наступні завдання:

- визначити принципи міжнародної співпраці з іноземними бізнес-структурами в умовах цифровізації;
- сформувати базу даних щодо потреб фахівців з інформаційних технологій на зарубіжному ринку праці;
- визначити основні, мінімально необхідні компетенції фахівців, для реалізації їх у відповідних сферах діяльності малого і середнього бізнесу;
- спроектувати програму взаємодії системи «РОБОТОДАВЕЦЬ – ВНЗ – ПРАЦІВНИК» шляхом аналізу напрацювань у цій сфері;
- зробити висновки щодо доцільності створення подібної системи.

Основний зміст роботи. В умовах інтенсивного впровадження цифровізації в різноманітні сфери діяльності людства власники малого та середнього бізнесу все частіше стикаються з нехваткою працівників зі знанням інформаційних технологій. Це розробники та адміністратори баз даних, java-розробники, web-розробники та web-дизайнери, адміністратори комп'ютерних мереж, спеціалісти з комп'ютерної графіки та ін. Попит на фахівців напряму 121 «Інженерія програмного забезпечення» та 122 «Комп'ютерні науки» настільки великий, що вони мають привілеї при отриманні робочої візи в деяких країнах ЄС. [2].

Але для отримання таких привілеїв треба мати банк даних роботодавців, зацікавлених у співпраці з виконавцями відповідного напряму діяльності. Звичайний механізм пошуку та роботи персоналу при тимчасовій праці простий - «РОБОТОДАВЕЦЬ – ПРАЦІВНИК». Перший шукає працівника і оплачує його роботу, другий – шукає і виконує роботу. Але дуже часто трапляються ситуації, коли працівник по закінченню пробного строку не задовольняє роботодавця. Переважно це трапляється через мовний бар'єр і в деяких випадках через низьку компетентність працівника в сфері інформаційних технологій. На практиці роботодавець частіше сплачує не за те, що хотів получить в кінцевому результаті, а за старанно виконану роботу, працівник же, за відсутності відповідних компетентностей, багато працює, часом виконує зайву роботу, а задоволення від праці не отримує. Якщо у стандартну модель взаємодії «РОБОТОДАВЕЦЬ – ПРАЦІВНИК» вживити вищі навчальні заклади, що формують фахівців з інформаційних технологій, то у виграші будуть усі учасники моделі «РОБОТОДАВЕЦЬ – ВНЗ – ПРАЦІВНИК».

Роботодавець буде мати для себе наступні переваги:

- можливість оперативного отримання інформації щодо динаміки зміни попиту на продукт його праці;
- кваліфіковано виконану роботу;
- можливі інноваційні рішення щодо поширення сфери діяльності свого бізнесу;

- використання досвіду провідних фахівців з суміжних напрямів діяльності;
- можливість замовлення на підготовку кваліфікованих працівників в умовах ВНЗ.

Звичайний працівник може працювати і паралельно отримувати дуальну освіту, що дозволить йому набути нових знань, компетентностей і досвіду для виконання роботи та кар'єрного росту.

До переваг ВНЗ слід віднести:

- можливість отримання інформативної бази даних щодо потреб фахівців на зарубіжному ринку праці;
- формування інноваційних напрямів в сфері наукових досліджень;
- реалізація наукових напрацювань в зарубіжних виданнях та конференціях;
- можливість стажування та проходження практики за кордоном;
- використання досвіду та вимог стейкхолдерів в освітньо-професійних програмах підготовки фахівців.

Варіанти міжнародної співпраці ВНЗ з малим та середнім бізнесом вже існують [3]. Наочним прикладом може бути співпраця НТУ «Дніпровська політехніка» з фірмою GeoBit (м. Хшанув, Польща).

Ще у 2011 році молодий геолог з польської шахти Piast (м. Берунь, Польща) приїхав до України на Міжнародний форум гірників. На форумі він познайомився з представниками Дніпровської політехніки (тоді Національного гірничого університету) з якими виявилися спільні інтереси і з того часу він веде наукову співпрацю з ВНЗ України. Спочатку як науковець, а з 2014 р. як підприємець. Сфера діяльності його фірми пов'язана з геологією та гірництвом. З 2016 р. викладачі Дніпровської політехніки проходять стажування з метою отримання міжнародного досвіду у сферах буріння, кар'єрного обладнання для видобування родовищ відкритим способом, геології тощо. У відповідь Дніпровська політехніка приймала на ознайомчу практику спочатку студентів-геологів, а потім і головних геологів з 7 областей Польщі. У 2017 р. на стажування були запрошені й викладачі кафедри програмного забезпечення комп'ютерних систем (ПЗКС), які разом з польськими колегами запровадили можливість перекладу й адаптування для українського користувача деяких програм обробки геологічних даних.

Це тільки один з прикладів міжнародної співпраці ВНЗ з малим бізнесом і на цьому поприщі є завдання які можуть дати старт багатьом проектам для фахівців спеціальностей 121 «Інженерія програмного забезпечення» та 122 «Комп'ютерні науки».

Висновки. В умовах сьогодення фінансова підтримка освітньої діяльності ВНЗ ґрунтується на поєднанні бюджетних і позабюджетних джерел. Рекомендована малодосліджена модель інформаційно-комунікаційної взаємодії ВНЗ з малим і середнім бізнесом базується на впровадженні в діючі бізнес-структури здобувачів вищої освіти, що володіють цифровими технологіями. Вважаємо, що подібне партнерство навчальних закладів і зарубіжних

підприємств малого і середнього бізнесу в умовах нестабільності ринку праці України забезпечить:

- попит на престижну університетську освіту;
- підвищення якості освіти;
- модернізацію системи фінансування ВНЗ;
- підвищення рівня конкурентоздатності ВНЗ на ринку праці;
- визначення найбільш привабливих сфер професійної діяльності.

Передбачається також, що адаптація системи вищої освіти до сучасних вимог ринку праці і збільшення обсягів попиту на освітні послуги ВНЗ України суттєво зменшить відтік вітчизняних студентів для навчання за кордон.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Данніков О. В., Розвиток цифрових компетенцій в умовах інформатизації суспільства. – 2018.

2. Zareba P., Zareba M. The Legal Aspects of Foreigners' Employment in the Context of the Polish Labour Market Needs //Economic and Regional Studies. – 2020. – Sciendo. – №.13. – P. 212-232.

3. Мацкевич Ю. І. Взаємодія закладів вищої освіти та бізнесу в умовах сучасного економічного розвитку //Вісник приазовського державного технічного університету. Серія: економічні науки. – 2019. – №. 37. – С. 134-140.

УДК 004.415.3:681.6

Н.П. Уланова¹, М.М. Одновол¹, В.В. Приходько¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ЯКІСТЬ ОСВІТИ – ГАРАНТІЯ ЕФЕКТИВНОСТІ НАВЧАЛЬНОГО ЗАКЛАДУ

Анотація. Представлено основні напрями забезпечення якості освіти в Національному технічному університеті «Дніпровська політехніка», що значною мірою залежить від рівня підготовки абітурієнтів, викладачів та організації технологічного процесу навчання. Висвітленні фактори, що впливають на якість вищої освіти та сприяють підвищенню ефективності вузу. Якість освітнього процесу забезпечується у першу чергу якістю технологій освіти.

Ключові слова: *якість освіти, потенціал вищого навчального закладу, якість освітньої діяльності, віртуальне навчальне середовище, навчально-методичне забезпечення, інформаційно-методичні бази, підвищення ефективності вузу.*

Вступ. Економічне зростання нації багато в чому залежить від рівня освіти. Стан освіти є показником сталого розвитку та конкурентоспроможності держави. Важливим елементом розвитку вищої школи, як умови становлення інноваційного суспільства в Україні, є якість освіти.

Згідно з дослідженнями Світового економічного форуму, Україна входить в першу десятку країн світу за охопленням населення країни вищою освітою. Разом з тим відзначається невідповідність викладання потребам ринку праці, низький рівень освіти українських ВНЗ в порівнянні зі світовими. Все це свідчить про необхідність трансформації кількісних показників у якісні. Ці перетворення повинні здійснюватися на основі збереження кращих національних освітніх традицій, урахуванням європейського та світового досвіду, підпорядкування вищої освіти законам ринкової економіки. Інтеграція України в Європейський простір вищої освіти активізувало увагу держави до питань якості вищої освіти. Зокрема, до Закону України «Про вищу освіту» включені такі терміни як:

– **якість освіти:** відповідність результатів навчання вимогам, встановленим законодавством, відповідним стандартом освіти та/або договором про надання освітніх послуг;

– **якість освітньої діяльності:** рівень організації освітнього процесу у ЗВО, що відповідає стандартам вищої освіти, забезпечує здобуття особами якісної вищої освіти та сприяє створенню відповідних нових знань [1].

Проблема якості освіти завжди перебувала в полі зору педагогічної теорії і практики. Однак на сучасному етапі суспільного розвитку вона набуває нового звучання. Актуальним питанням реформування вищої освіти, створення і впровадження систем якості присвячена велика кількість публікацій. Питання забезпечення якості були предметом наукових досліджень Ю. Захарова, Л. Кайдалова, С. Ніколаєнко та ін. Створенню та впровадженню системи якості в освітній процес присвячені роботи В. Логачова, В. Белова, Л. Віткіна. Аналіз критеріїв якості підготовки сучасних фахівців проводився Г. Борліковим, О. Єсіною, М. Кісіль. Моніторингу якості освіти присвячені роботи О. Локшиної, О. Авраменко, В. Котяк. Інформатизацію освіти, як важливу складову підвищення результативності навчального процесу, розглянуто в роботах М. Шишкіної, Т. Вдовичина, В. Кременя [2, 3]. Окреслені проблеми знаходяться у центрі досліджень і зарубіжних науковців [3, 4].

Аналіз публікацій дозволяє зробити висновок, що проблема якості освіти є складною проблемою і вимагає подальшої розробки.

Постановка задачі полягає у висвітленні факторів, що впливають на якість вищої освіти та сприяють підвищенню ефективності вузу.

Основний зміст роботи. Якість освіти є багатовимірною категорією, яка за своєю суттю відображає різні аспекти освітнього процесу – філософські, соціальні, педагогічні, економічні, інформаційні та ін. Саме тому в рішенні проблем підвищення якості мають бути задіяні численні напрямки функціонування вищих навчальних закладів.

Одним з важливих чинників, що впливає на якість освіти і є відправним параметром якісного освітнього процесу – це потенціал вищого навчального закладу. Саме тому велика увага в національному технічному університеті «Дніпровська політехніка» приділяється змісту, особливостям організації та реалізації навчального процесу, якості стандартів і програм підготовки бакалаврів, магістрів і аспірантів. Міждисциплінарні освітні програми

розробляються за участю представників різних напрямів підготовки і спеціальностей з урахуванням потреб ринку праці. Наявність комплексів навчально-методичного забезпечення дисциплін та інформаційно-методичної бази, якість професійної підготовки та кваліфікації педагогічних і науково-педагогічних кадрів дозволяють впроваджувати в навчальний процес сучасні педагогічні та інформаційні технології. Розгорнуте на базі платформи Moodle віртуальне навчальне середовище реалізує в університеті «Дніпровська політехніка» філософію педагогіки соціального конструктивізму, сприяє поєднанню традиційних і комп'ютерно-орієнтованих засобів і форм навчання, створенню дистанційних курсів, організації змішаного та очного навчання [6, 7]. Започатковане також навчання з використанням мобільних пристроїв (m-learning).

Забезпечення якості освіти значною мірою залежить від рівня підготовки абітурієнта. Профорієнтаційна робота ведеться за різними напрямками. Одним із таких напрямків є популяризація серед старшокласників та випускників шкіл технічних спеціальностей. Наприклад, компанія Інтерпайп разом з кафедрою електропривода університету розпочала освітньо-тренувальну програму з мехатроніки Interpipe Mechatronic Lab. Навчання за цією програмою допоможе молоді ознайомитись з принципами роботи сучасного промислового обладнання, розвинути професійні інтереси.

З метою залучення талановитої молоді, надання допомоги у виборі професії у дні «відкритих дверей» вступники мають змогу зустрітися з провідними вченими та професорсько-викладацьким складом випускаючих кафедр, ознайомитися з лабораторіями університету, організацією навчання та побуту. Вступники-молодші спеціалісти можуть взяти участь у профорієнтаційній олімпіаді, яку проводить Центр моніторингу знань та тестування. Під час Skype-зустрічей учні 11-тих класів районних шкіл отримують інформацію про університет, його факультети та спеціальності, особливості вступної кампанії. Інноваційні технології дозволяють проводити Інтернет-олімпіади. Завданням Всеукраїнської олімпіади «+20» є виявлення обдарованих школярів, які бажають вступити на природно-математичні та інженерно-технічні спеціальності університету. Перший тур олімпіади дистанційний, а другий – очний. Переможці отримують в результаті додаткові бали до одного з конкурсних предметів.

Одним з чинників, що характеризують якісний потенціал ВНЗ є рівень його інтеграції в світовий освітній процес. В НТУ «Дніпровська політехніка» успішно реалізуються програми між університетського обміну студентами, аспірантами, педагогічними та науково-педагогічними працівниками Standart Grants Programme, Rita Programme, Jean Monnet Programme, ERASMUS MUNDUS PROGRAMME з European Commission, Visegrad Fund, Polish-American Freedom Foundation та The Education, Audiovisual and Culture Executive Agency. Закладені в Законі України «Про вищу освіту» можливості міжнародного співробітництва дозволили, зокрема, започаткувати спільну з Університетом Вітауса Великого міжнародну міждисциплінарну програму підготовки бакалаврів «Світова політика та економіка», в співпраці з

Університетом Комбленц-Ландау (Німеччина) реалізувати міжнародну магістерську освітню програму «Системний аналіз і управління» (спеціальність «Моделювання складних систем»). В університеті створюються всі умови для студентів, які бажають отримати спільні, подвійні або потрійні дипломи.

Якість освітнього процесу забезпечується якістю технологій освіти. Ці технології повинні базуватися на використанні форм, методів і засобів навчання, в тому числі й Інтернет-технологій, які активізують пізнавальну діяльність студентів, розвивають природні здібності та здатність самостійного пошуку рішень, дозволяють викликати інтерес до предмета і, як наслідок, до навчальної та інтелектуальної професійної діяльності. Невід'ємною частиною процесу формування сучасного фахівця є організація на належному рівні самостійної діяльності студентів, її педагогічна підтримка, створення системи контролю на різних етапах професійного навчання.

Використання активних методів навчання таких, як дискусії, ділові ігри, тренінги, формує мислення, вміння визначати мету діяльності та спрямованість на досягнення цієї мети [8]. Таким чином відбувається розвиток дуже важливих в сучасному житті особистісних інтелектуальних якостей: вміння отримувати, аналізувати, самостійно переробляти інформацію, залучати для досягнення результату власний досвід. Робота в групі передбачає продуктивну співпрацю з партнерами, виконання різних ролей і функцій в колективі, вміння брати на себе відповідальність за прийняття і виконання рішень. Прикладами використання інтерактивних методів в НТУ «Дніпровська політехніка» можуть служити застосування методу проектів на випускаючих кафедрах, зокрема на кафедрі програмного забезпечення комп'ютерних систем, участь студентів і аспірантів в міжнародних тренінгах зі створення міні-проектів, участь у конкурсах, виконання дослідницьких завдань тощо.

Творчий підхід і ефективність педагогічної роботи професорсько-викладацького складу дозволяє не тільки допомагати студенту отримувати професійну підготовку, а й домогтися переходу отриманих знань на рівень компетенцій. При цьому викладач не тільки передає знання, а й формує особистість студента, його світогляд і духовність. Постійне професійне вдосконалення, висока інформаційна комунікація, використання результатів наукових досліджень в підготовці майбутніх фахівців, висвітлення основних досягнень науки і техніки та можливих тенденцій їх розвитку робить викладача цікавим для сучасного студента. Закономірним є той факт, що ефективність сучасної науково-дослідної діяльності стає одним з важливіших показників при визначенні кращих вузів країни. Слід зазначити, що сталий розвиток кожного вузу є неможливим без надійного фінансування, оновлення нормативно-правової бази згідно з базовими вимогами чинного законодавства країни.

Свідченням результативності освітнього процесу є відповідність досягнутого рівня професійної підготовки фахівця державним стандартам вищої освіти. Саме таким бачить випускника сучасний роботодавець: спеціалістом з достатніми практичними та теоретичними знаннями та навичками, спроможного по закінченню короткого адаптаційного періоду ефективно працювати; спеціалістом, що володіє методологією самоосвіти,

іноземними мовами, має розвинуті професійні якості (ініціативність, організаторські здібності), високу працездатність.

Наукова новизна: досліджено алгоритми організації освітнього процесу в Національному технічному університеті «Дніпровська політехніка», що відповідає стандартам вищої освіти, забезпечує здобуття особами якісної вищої освіти та сприяє створенню нових інтелектуальних знань.

Висновки. В сучасних умовах підвищення якості освіти можливе лише з урахуванням тенденцій ринку освітніх послуг та ринку праці, бо саме вони визначають стратегію позиціонування вищого закладу.

Для цього необхідно:

- аналізувати потреби ринку, виявляти пріоритетні напрямки розвитку вузу;
- аналізувати відгуки споживачів на появу нових пропозицій;
- взаємодіяти з замовниками і споживачами освітніх послуг;
- забезпечувати вуз фінансовими коштами з різних джерел;
- оновлювати матеріальну-технічну базу університету;
- реалізовувати вимоги до освіти з урахуванням міжнародних стандартів;
- створювати умови для участі студентів в укладанні навчальних програм, науково-практичній і культурній діяльності вузу;
- підвищувати кваліфікацію викладачів для впровадження інноваційних технологій в освітню практику ВНЗ, в тому числі таких, які базуються на використанні інформаційно-комунікаційних технологій і вимагають спеціальної підготовки;
- залучати до освітнього процесу провідних фахівців з виробництва та професорів університетів-партнерів;
- удосконалювати та адаптувати методи тестування як об'єктивні методи діагностики рівня знань;
- вдосконалювати освітній процес з урахуванням моніторингу рівня знань.
- відстежувати якість працевлаштування і досягнення випускників.

Таким чином, досягти високих результатів в напрямку забезпечення якості вищої освіти можливо тільки при комплексному підході до розглянутих питань планування, організації та управління освітньою діяльністю.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Про вищу освіту: Закон України від 01.07.2014 №1556-VII [Електронний ресурс] – Режим доступу: <https://zakon.rada.gov.ua/laws/show/1556-18>.
2. Шишкіна М. П. Проблеми інформатизації освіти України в контексті розвитку досліджень оцінювання якості засобів ІКТ / М.П. Шишкіна, О.М. Спірін, Ю.Г. Запорожченко // Електронне фахове видання. Інформаційні технології і засоби навчання. – № 1 (27). – 2012. [Електронний ресурс]. – Режим доступу: <http://journal.iitta.gov.ua/index.php/itlt/article/view/632/483>.
3. Кремень В.Г. Інформатизація освіти – провідний напрям підвищення результативності навчального процесу / В. Г. Кремень // Комп'ютер у школі та сім'ї. – 2011.(1). – С. 3-6.

4. Kucharčikova A. Possibilities to increase the quality of education at universities. Proceeding of the International Conference of Education Capacity of University Teachers, Tn UAD Trenčín, 11-12.3.2008.– P.77-82.– Режим доступу: <https://www.sciencedirect.com/science/article/pii/S1877042813049689>.

5. Brdulak J. Ocena jakości kształcenia w Polsce – problemy i rekomendacje. Nauka i szkolnictwo wyższe. 2(48). 2016.–P.81-94. – Режим доступу: <https://pressto.amu.edu.pl/index.php/nsw/article/view/7637>.

6. Рибалко А.Я. Комп'ютерна візуалізація технологій складання електричних машин в умовах дистанційного навчання /Рибалко А.Я., Мещеряков Л.І., Уланова Н.П., Руссу А.Ю. //Наукові записки Тернопільського національного педагогічного університету ім.В.Гнатюка. Серія Педагогіка.– 2011.– №1.– С.49-54.

7. В.И.Сулаев. Проектирование и реализация сетевого учебного курса "высшая математика"/В.И. Сулаев, Л.И. Мещеряков, Н.П. Уланова, В.В. Приходько // Збірник наукових праць Національного гірничого університету.- 2013.-№ 40.-С. 222-228.

8. Чепіль М. Педагогічні технології : навч. посіб. [рек. Міністерством освіти і науки, молоді та спорту України як навчальний посібник для студентів вищих навчальних закладів (лист № 1/11 –1506 від 22.02.11)] / Марія Чепіль, Надія Дудник. –К. : Альма-матер, 2012. – 224с.

УДК 004.415.3:681.6

Л.І. Мещеряков¹, О.М. Галушко¹, М.В. Курило¹, К. Палаш²

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

²Технічний університет «Краківська гірничо-металургійна академія ім. Станіслава Сташиця» (AGH), м. Краків, Польща

УДОСКОНАЛЕННЯ ПРОЦЕСУ ПЕРЕДАЧІ ФАЙЛІВ БЕЗДРОТОВИМ СПОСОБОМ МІЖ ПРИСТРОЯМИ З РІЗНИМИ ОПЕРАЦІЙНИМИ СИСТЕМАМИ НА БАЗІ ТЕХНОЛОГІЇ WEBRTC

Анотація. Описано методи удосконалення процесу передачі файлів між пристроями з різними операційними системами за рахунок використання технології WebRTC та мережевої архітектури peer-to-peer.

Ключові слова. багатоплатформовий, бездротова передача файлів, графічний інтерфейс, клієнт-серверна архітектура, peer-to-peer, WebRTC, JavaScript, Firebase.

Вступ. За останнє десятиліття всі стали свідками як фізичні носії, на кшталт компакт-дисків та флеш-носіїв були майже витіснені бездротовими способами передачі файлів. Це було досягнуто за рахунок повсюдного розвитку мережі інтернет та інших бездротових технологій. Поділитися файлом тепер можна через хмарне сховище, або навіть без нього. Великі компанії серед яких Apple, Microsoft, Huawei розробляють рішення, що дозволяють передавати дані

прямо з одного пристрою на інший без проміжного сервера. Наприклад, Apple AirDrop, Windows Nearby Share, Huawei Share та інші. Однак кожне з цих рішень провідних корпорацій працює лише в окремій екосистемі. В роботі представлено розробка рішення, що націлене на використання в пристроях під керуванням різних операційних систем.

Постановка задачі. Для досягнення поставленої мети сформовані і вирішені такі завдання: проаналізовані принципи побудови багатоплатформових додатків; визначено методи передачі даних за допомогою технології WebRTC; розроблено архітектуру додатку; розроблено додаток; зробити замір швидкості передачі даних; сформовані висновки, щодо ефективності даного способу.

Основний зміст роботи. В процесі дослідження були отримані наступні результати. При вивченні принципів побудови додатку для передачі файлів між пристроями з різними операційними системами, було вирішено розділити проектування даної системи на декілька етапів: вирішення проблем з підтримкою різних операційних систем. Для вирішень проблем з підтримкою багатьох операційних систем було вирішено розробити веб-додаток з використанням багатоплатформової мови JavaScript. Такий підхід дозволяє додатку працювати в веб-браузері, а також бути вбудованим в додаток та скомпільованим для будь-якої операційної системи. Таким чином не потрібно розробляти окреме рішення для кожної окремої системи.

Для забезпечення оптимального користувальницького досвіду додатком було прийнято рішення використовувати парадигму адаптивного дизайну. Таким чином, зроблено один дизайн в рамках одного додатку, що буде підлаштовуватись під будь-який тип пристроїв: персональний комп'ютер, ноутбук, планшет або смартфон. Оскільки це веб-додаток, то основний тип архітектури – клієнт-серверна. Цей тип використовується, коли дані про кожного користувача заносяться в базу даних, а також на сервері виконується розмежування їх доступу. Однак додаток передбачає передачу файлів, що можуть мати великий розмір та створити надлишкове навантаження на сервер. Щоб цього уникнути для передачі даних файлу було використано архітектуру peer-to-peer. Цей тип архітектури призначений для обміну даними між користувачами напряму без використання проміжного серверу.

Для передачі даних самого файлу обрана технологія WebRTC. WebRTC – це технологія, що дозволяє веб-додаткам та сайтам захоплювати та вибірково передавати аудіо та/або відео медіа-потіки, а також обмінюватись довільними даними між браузерами без обов'язкового використання посередників. Набір стандартів, що включає в себе технологія WebRTC дозволяє обмінюватись даними та проводити пірингові телеконференції без необхідності користувачу встановлювати плагіни або будь-яке стороннє програмне забезпечення. WebRTC була обрана для того, щоб спростити процес передачі між пристроями, оскільки вона вже має в собі всі потрібні механізми для передачі та встановлення зв'язку. Таким чином значно спрощується розробка та підтримка додатку, а також підвищується стабільність його роботи. WebRTC є технологією, що підтримується багатьма веб-браузерами та операційними

системами. З цією технологією можна взаємодіяти за допомогою скриптової мови JavaScript, що є багатоплатформовим рішенням.

Для функціонування WebRTC потрібен STUN сервер. STUN – це сервер, що повідомляє вузлу його знаходження в мережі. Тобто він повертає IP адресу та порт вузла в глобальній мережі. Це необхідно для подолання NAT на маршрутизаторах та передачі файлів напряму між вузлами, що знаходяться навіть в різних мережах. Ключовим моментом для встановлення базового зв'язку у WebRTC є генерація ICE Candidate. Це інформація про вузол, яку потрібно передати іншому вузлу для встановлення контакту. Відповідно це можна зробити будь-яким способом, але найбільш поширеним є використання веб-сокетів. Для цього потрібен простий сокет-сервер, що буде передавати невеликі повідомлення між вузлами та виконувати роль координатора їх роботи. Для передачі файлів через WebRTC необхідно встановити з'єднання з сокет-сервером, що буде транслювати повідомлення від одного вузла до іншого, згенерувати ICE Candidate для кожного з користувачів та відправити його одне одному. Після цього відкрити потік даних та правильно налаштувати розмір буферу даних через який передається файл. При завершенні передачі файлу зберегти його на стороні клієнта.

Загальна архітектура роботи системи передачі даних на основі технології WebRTC наведена на рис. 1. Тут зображено ситуацію, коли Вузол 1 має намір передати Вузлу 2 файл. Вузол 2, що щойно приєднався до сесії, робить запит на STUN-сервер з метою дізнатися свою зовнішню IP-адресу. Дізнавшись її, вузол повідомляє сокет-сервер про своє знаходження в мережі. Сервер одразу ж передає ці дані Вузлу 1, що приєднався до мережі сесії. Тепер у Вузла 1 є все необхідне, щоб розпочати передачу файлу напряму до Вузла 2, що відбувається саме на базі технології WebRTC та остаточно реалізує архітектуру передачі даних peer-to-peer.

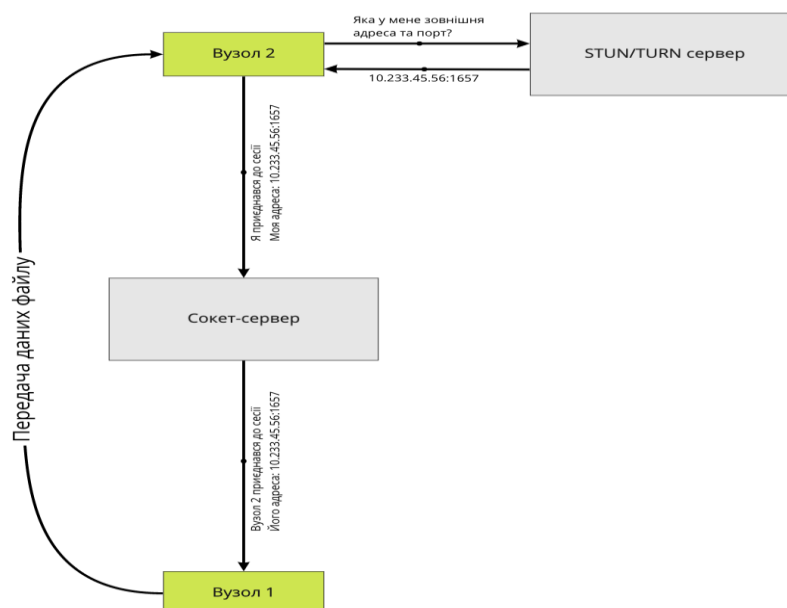


Рис. 1. Архітектура роботи додатку для передачі файлів на базі технології WebRTC

Після дослідження інформаційно доступних технологій та розробки архітектури додатку, було створено сам додаток. Оскільки базова архітектура функціонування додатку – клієнт-серверна, то остаточно розробку було поділено на дві структурні частини. В якості основного серверного рішення обрано Firebase від Google. Це безкоштовне рішення, що дозволяє запустити базу даних та сокет-сервер і взаємодіяти з ними в режимі реального часу. При цьому не потрібно вирішувати жодних інфраструктурних задач, на кшталт підключення хостингу або виділення фізичних ресурсів.

Ще однією перевагою такого рішення є відсутність необхідності писати серверний код. Це дуже економить час розробки, та спрощує її в цілому. Вся взаємодія клієнта з сервером відбувається через веб-сокети. Якщо потрібно занести в базу даних будь-яке повідомлення, клієнт відправляє серверу сформоване повідомлення і той заносить необхідні дані та розсилає все, що потрібно іншим клієнтам. Клієнтська сторона базується на JavaScript-фреймворку Ember. Використання фреймворку значно спрощує контроль над навігацією та шаблонізацію елементів користувальницького інтерфейсу. JavaScript було обрано як найбільш універсальне та поширене рішення для багатоплатформових додатків. Готовий код може бути використано як у браузері, так і зібрано для Android, iOS, MacOS, Windows та Linux. Середовище виконання коду – програмна платформа NodeJS. Саме в даному середовищі відбувається збірка коду, та видача його на клієнт.

Після розробки додатку потрібно оцінити ефективність його роботи, замірявши швидкість передачі даних. Це в цілому комплексне завдання, що поділене на декілька функціональних етапів. На передачу даних значний вплив має мережа, в якій знаходяться пристрої. Передача даних може проходити при різних обставинах: коли обидва пристрої знаходяться в одній локальній мережі; коли обидва пристрої знаходяться в різних мережах та підключені до швидкісних мереж (100Мб/с); коли один з пристроїв підключено до мобільної мережі 4G. Також важливе значення на передачу може мати розмір самого файлу. В даному експерименті досліджено три типи файлів з різними розмірами: медіа-файл – відео формату mp4 (1.5ГБ та 170МБ); медіа-файл – зображення формату jpeg 37МБ та 3.6МБ); текстовий документ формату docx 1.7МБ та 0.15КБ). Всі файли передаються по п'ять разів від пристрою-відправника до пристрою-приймача в кожній з зазначених типів мереж. При цьому фіксується час передачі файлу. Відлік часу розпочинається коли приймач натискає кнопку прийняти файл і рахується до моменту завершення передачі файлу. На основі отриманого часу та відомого розміру файлу розраховується середня швидкість передачі даних (в Мб/с) для кожного з типів файлів.

Результати продемонстрували достатню швидкість, так, наприклад, в локальній мережі вона становила 95 Мб/с. Також було виміряно і середній час передачі файлів між пристроями. Час передачі зображення в 3.6 МБ навіть в повільній мобільній мережі становив всього 4 секунди, а в швидкій локальній мережі 0.3 секунди. На швидкість передачі тип та розмір файлу не впливають, таким чином його можна використовувати не лише для передачі медіа-файлів, а і будь-яких інших файлів.

Зі зростаючим попитом на бездротові засоби передачі даних розроблене рішення може знайти застосування одразу в декількох сферах. По-перше, для персонального користування, щоб ділитися файлами між декількома власними пристроями. По-друге, для відправки файлу іншому користувачу, який може знаходитися взагалі в іншому місці та бути підключеним до іншої мережі. Маючи посилання на спільну кімнату з користувачем, передача даних також можлива. По-третє, як рішення для бізнесу для покращення комунікації між працівниками. Це дозволить не створювати зайвої мережевої інфраструктури, а ділитися файлами між працівниками напямую.

Наукова новизна. Обґрунтована та дістала подальшого розвитку технологія передачі файлів за допомогою засобів WebRTC.

Висновки. В результаті спроектовано систему передачі файлів між пристроями з різними операційними системами за допомогою сучасної технології передачі даних WebRTC.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Ambler T. JavaScript Frameworks for Modern Web Dev / T. Ambler, N. Cloud Apress, 2015.520 p.
2. Burnett D. C. WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web / Burnett D. C. Burnett, Johnston A. B. Johnston – Digital Codex LLC, 2014. – 350p.
3. Loretto S. Real-Time Communication with WebRTC: Peer-to-Peer in the Browser / Loretto S., Romano S.P. – O'Reilly Media, 2014. – 164p.
4. Moore D. Peer-to-Peer: Building Secure, Scalable, and Manageable Networks / D. Moore, J. Hebler – McGraw-Hill Companies, 2001. – 512p.
5. Куроуз Д. Компьютерные сети. Нисходящий подход, 6-е издание / Д. Куроуз, К. Росс – Москва : Издательство «Э», 2016 – 912 с.

УДК 004.415.3:681.6

Б.І. Мороз¹, А.А. Мартиненко¹, В.О. Мірошніченко¹, М. Пужьо²

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

²Технічний університет «Краківська гірничо-металургійна академія ім. Станіслава Сташиця» (AGH), м. Краків, Польща

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ЗНАХОДЖЕННЯ МІСЬКИХ ОСТРОВІВ ТЕПЛА ЗА ДОПОМОГОЮ СУПУТНИКОВИХ ЗНІМКІВ В ІНФРАЧЕРВОНОМУ ТЕПЛОВОМУ ДІАПАЗОНІ

Анотація. Описано процес покращення алгоритму знаходження міських островів тепла за допомогою космічних знімків, отриманих з супутника Landsat-8. Покращення досягається за рахунок виділення територій із критичними температурами.

Ключові слова: *R, urban heat islands, land surface temperature, brightness temperature, ДЗЗ, електромагнітне випромінювання, електромагнітний спектр.*

Вступ. За останні півтора століття дуже сильно покращилася якість і продуктивність праці, зріс рівень урбанізації, почався стрімкий розвиток технологій. Внаслідок цього, на сьогоднішній день, ми маємо неймовірно високий рівень технологій, міста-мегаполіси та все більше і більше автономних виробництв. Однак, цей розвиток вкрай негативно позначається не тільки на мікроклімат населених пунктів, а й на глобальному кліматі.

З моменту, як людство почало займатися дистанційним зондування Землі (ДЗЗ) за допомогою супутників – з'явилася можливість вивчати і займатись моніторингом багатьох факторів, що впливають на навколишнє середовище. Зважаючи на ці можливості, в останні десятиліття, зростає науковий інтерес до вивчення проблеми так званих міських островів тепла (далі УНІ – urban heat islands). Сам по собі цей термін описує феномен різниці температури в центрі міста і за її межами. Це добре помітно в містах-мегаполісах, де різниця може досягати від 1,0-3,0 °С вдень та до 12,0 °С і більше вночі. На інтенсивність даних островів тепла впливає безліч факторів, таких як: ландшафт, малий рівень альbedo асфальту і будівель в порівнянні з природним ландшафтом, кількість та якість рослинності на території міста, повітряні потоки і т.п.

Даний феномен не тільки робить внесок у погіршення навколишнього середовища (як приклад: вплив на глобальне потепління), але і негативно позначається на людському здоров'ї.

Пошуки, дослідження і оцінка міських островів тепла є важливим завданням в сучасних дослідженнях навколишнього середовища. Її рішення дозволить використовувати ефективні методи боротьби і запобігання цих аномалій.

У даній роботі такі теплові острови були досліджені на території міста Дніпро, Дніпропетровської області. Дніпро – четверте місто-мільйонник за чисельністю населення України. Промисловість - головний фактор погіршення навколишнього середовища в місті та підвищення температури, зокрема.

Постановка задачі. Для досягнення поставленої мети в роботі сформульовані і вирішені такі завдання:

- пошук та завантаження супутникових даних з Landsat-8;
- знаходження температури поверхні території міста Дніпро та виділення теплових аномалій завдяки різним алгоритмам;
- дослідження результатів пошуку найгарячіших точок міста та дати обґрунтування їх появи;
- вдосконалення алгоритму пошуку найгарячіших точок.

Основний зміст роботи. Під час роботи над даною роботою були використані наступні методи та інструменти.

- активний метод дистанційного зондування Землі;
- метод розрахунку температури поверхні землі (land surface temperature – LST) [2];

- метод пошуку містких островів тепла (urban heat islands – UHI) [4] та покращений метод пошуку UHI;
- R, GDAL, RStudio, Nominatim. API.

ДЗЗ можна дати визначення як процесу або методу добутку інформації про об'єкт, частину поверхні або явище без прямого контакту із цим об'єктом. Суть методу – інтерпретація результатів вимірювання електромагнітного випромінювання, яке відбивається або випромінюється об'єктом і реєструється в віддаленій від нього точці простору. Методи, які використовуються у цій роботі пов'язані із використанням даних, отриманих з сенсорів космічного апарату Landsat-8.

Основу дистанційного зондування складає вимір енергії електромагнітного випромінювання від різних джерел, найважливішим з яких є Сонце. Більшість сенсорів, які використовуються в дистанційному зондуванні, призначені для вимірювання параметрів відбитого сонячного світла (так зване пасивне ДЗЗ). Інші реєструють власне випромінювання Землі або самі є джерелами випромінювання (активне ДЗЗ).

Атмосфера значно впливає на отримані дані, як правило двома способами: шляхом розсіювання в атмосфері та поглинання енергії земною поверхнею. Через усі ці фактори, сканування земної поверхні може вестися лише у так званих «вікнах прозорості». Вони існують в інтервалах 0,3-5мкм, 8-14мкм, а також 30-80мкм. У цій роботі була використана зйомка у перших двох інтервалах (3й та 4й канали – перший інтервал, 10й та 11й канали – другий). В останньому інтервалі провадиться радіотеплова зйомка. На рис. 1 показані вікна прозорості та типи зйомок, які у них робляться.

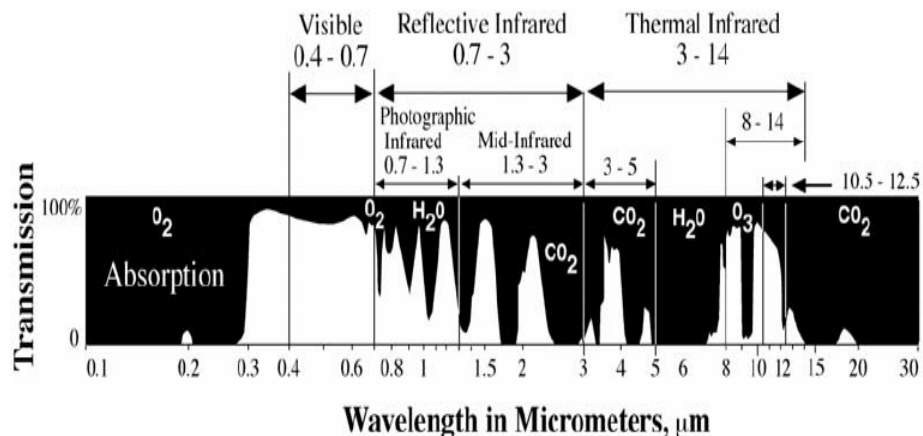


Рис. 1. Схема вікон прозорості

Усі тіла, якщо мають температуру більшу за абсолютний нуль (-273,15), можуть випромінювати електромагнітні хвилі, які, у свою чергу, мають різну довжину. Діапазон від гама-випромінювання до радіохвиль має назву «електромагнітний спектр».

Земля з космосу переважно досліджується в оптичній частині електромагнітного спектра (0.3 - 15мкм). Оптичний діапазон поділяється на:

- видиму частину спектру (0.4 - 0.72 мкм);

- ближній інфрачервоний (далі ІЧ) діапазон (0.72 - 1.3 мкм);
- середній ІЧ діапазон (1.3 - 3.0 мкм);
- теплової ІЧ діапазон (8 - 15 мкм).

У даній роботі найбільший інтерес представляє тепловий ІЧ діапазон (10й або 11й канали Landsat-8), тому що саме за його допомогою можемо отримати температуру поверхні Землі, а також ближній інфрачервоний (5й канал) та 4й канал (червоний), бо завдяки ньому ми можемо розрахувати нормалізований диференційний вегетаційний індекс (NDVI).

При виборі знімку треба завжди дивитися на те, наскільки великим був відсоток хмарності на момент зйомки, бо при його занадто сильному показнику буде неможливо отримати коректну температуру. Саме захмареність (Scene Cloud Cover – SCC) грає основну роль у виборі сцени з якою ми будемо працювати. Для отримання коректного результату аналізу LST необхідно шукати знімок літнього або зимового періоду. Тому що, у ці сезони найкраще видно контраст температур.

Окрім SCC, треба звернути увагу на розташування предмета дослідження, бо може статися так, що об'єкт дослідження знаходиться на самому краю сцени, або взагалі його частина не попала на знімок. Для коректного вибору шляху супутника (path) та рядка (row) слід використовувати спеціальні сервіси.

Територією дослідження було обрано місто Дніпро. Воно було засновано в 1776 році і на момент 01.12.2018 року мало населення 1000717 чоловік. Крім цього, Дніпро – один з найбільших промислових, транспортних і економічних центрів, а також центр металургії України (особливо чорної металургії).

Слід зауважити, що температура повітря, яку ми можемо отримати подивившись на термометр та температура поверхні землі, яку ми отримуємо завдяки супутникам не є одним і тим самим, але вони мають сильний зв'язок. Температура поверхні землі (Land surface temperature, LST) – це радіаційна температура «шкіри» землі, отримана від сонячного випромінювання. Спрощеним визначенням було б те, наскільки гаряче «поверхня» Землі відчувала б на дотик в певному місці. З точки зору супутника, «поверхня» – це те, що він бачить, коли дивиться через атмосферу до землі. Це може бути сніг і лід, трава на газоні, дах будівлі або листя в лісу. Температура земної поверхні не є такою ж, як температура повітря, яка включена в щоденний звіт про погоду. Саме завдяки LST ми можемо знайти UHI. У свою чергу, температура повітря на поверхні землі (LSAT) – це температура повітря поблизу поверхні Землі що зазвичай вимірюється від 1,5 до 2 м загальними метеорологічними станціями, розподіленими по території.

Для отримання LST потрібно здійснити обробку продукту Landsat Collection Level-1 – сцени, що отримана завдяки сенсору TIRS та вже геометрично та радіометрично калібрована. Даний метод оснований на праці. Зображення далі необхідно обробити для отримання значень енергетичної яскравості верхньої частини атмосфери (Top Of Atmosphere - TOA), для цього використовується формула [2].

$$L_{\lambda} = M_{\lambda} \cdot Q_{cal} + A_{\lambda}, \quad (1)$$

де, L_λ – енергетична яскравість; Q_{cal} – відкаліброване значення пікселя;
 M_λ, A_λ – каналні фактори мультиплікативного та адитивного перемасштабування з файлу метаданих відповідно.

Після отримання TOA необхідно перетворити його на температурну яскравість [2].

$$T_B = \frac{K_2}{\ln\left(\frac{K_1}{L_\lambda} + 1\right)}, \quad (2)$$

де, T_B – ефективна температура на супутнику (у Кельвінах);

L_λ – спектральна енергетична яскравість (Вт/(м²·ср·мкм));

K_1, K_2 – калібрувальні константи з файлу метаданих.

Проте значення температури яскравості не є LST, а швидше за все змішаним сигналом, що включає енергію випромінювання Землі, висхідну та низхідну енергетичну яскравість. Випромінювальна здатність земної поверхні є одним з основних параметрів для одержання LST. Будемо використовувати метод граничних значень стандартизованого індексу відмінностей рослинного покриву (NDVI) [2].

$$NDVI = \frac{NIR - RED}{NIR + RED} \quad (3)$$

де, NIR – канал близького інфрачервоного випромінювання (Band 5);

RED – червоного випромінювання (Band 4)

Випромінювальна здатність земної поверхні може оцінюватись за методом, представленим у праці. Перекласифікація значень NDVI у випромінювальну здатність наведена у табл. 1

Таблиця 1

Оцінка випромінювальної здатності за NDVI

$NDVI < 0.185$	0.995
$-0.185 \leq NDVI \leq 0.157$	0.970
$0.157 \leq NDVI \leq 0.727$	$1.0094 + 0.047 \ln(NDVI)$
$NDIV > 0.727$	0.990

Тепер, коли визначена випромінювальна здатність, то можливо розрахувати LST, використовуючи метод з праці [2].

$$T = \frac{T_B}{1 + (\lambda \cdot \frac{T_B}{\rho}) \ln \varepsilon} \quad (4)$$

де, λ – довжина хвилі енергетичної яскравості (для 10го каналу Landsat-8, $\lambda = 12\text{мкм}$); $\rho = h \cdot c / \sigma$ ($1,438 \cdot 10^{-2} \text{ м} \cdot \text{К}$); σ – константа Больцмана ($1,38 \cdot 10^{-23}$

Дж/К); h – константа Планка ($6,626 \cdot 10^{-34}$ Дж·с); c – швидкість світла у вакуумі ($2,998 \cdot 10^8$ м/с).

Після отримання температури необхідно з'ясувати як можна знайти острів тепла. З першого підрозділу було з'ясовано, що УНІ – це територія, температура якої перевищує температуру навколишньої місцевості. Але наскільки вона повинна бути більшою? Чи потрібно вважати різницю у $2-3^\circ$ температурною аномалією, чи ця різниця повинна бути $5-7^\circ$ або більше? Саме для того, щоб коректно виділити аномальні території потрібно мати критерій оцінювання.

Використовуючи формулу для класифікації УНІ з праці [4], ми можемо виділити території, на яких маємо високі температури. На рис. 2 показана карта міста з виділенням усіх УНІ

$$LST > \mu + 0.5 \cdot \delta, \quad (5)$$

де, μ - середнє арифметичне від LST; δ - стандартне відхилення.

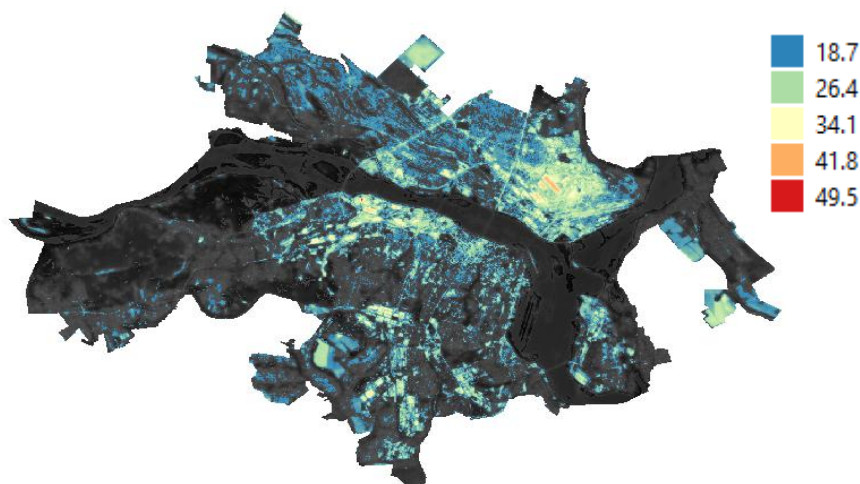


Рис. 2. Карта УНІ на території міста Дніпро

В результаті (рис. 2) дуже значна територія міста була виділена, але покрита територія є досить великою і для впровадження дій, направлених на зниження температур та покращення ситуації, використовуючи ці результати – неможлива. Нам необхідно оптимізувати виділення УНІ, для того щоб отримати тільки ділянки з екстремальною температурою.

Зважаючи на опис території дослідження, можна припустити, що найгарячіші точки міста будуть на територіях різних заводів. Саме тому у цій роботі головна увага буде зосереджена на знаходженні територій з екстремальною високою температурою.

Після отримання звичайних теплових островів, необхідно знайти екстремально гарячі точки (викид). Викидами ми вважаємо значення температур, які виходять за «зовнішні кордони» нашого набору значень – виходить за 3х-разовий межквартильний діапазон.

$$T_{\text{extr}} = 3 \cdot \text{IQR}(T_{\text{LST}}) \quad (6)$$

де, T_{LST} – температура ділянки; IQR – межквартильний діапазон.

Після цих операцій залишаються лише ті ділянки, які матимуть саму найвищу температуру. Це дозволить нам побачити найпроблемніші зони. Результат можна побачити на рис. 3.



Рис 3. Карта з найгарячішими точками на території міста

Наукова новизна результатів роботи визначається тим, що удосконалено алгоритм пошуку міських островів тепла, в результаті чого, покращилася точність знаходження аномально теплих зон.

Висновки. В результаті виконання роботи було програмно реалізовано алгоритм обчислення міських островів тепла, використовуючи космічні знімки супутника Landsat-8. У тому числі, було удосконалено метод виділення критичних теплових островів.

ПЕРЕЛІК ПОСИЛАНЬ:

1. H. Wickham, G. Grolemund R for Data Science: Import, Tidy, Transform, Visualize, and Model Data 1st Edition / H. Wickham, G. Grolemund, 2018 – 605 p
2. Guiyun Tian, Algorithm for Automated Mapping of Land Surface Temperature Using LANDSAT 8 Satellite Data / Guiyun Tian, Volume 2016, Article ID 1480307, 8 p.
3. Urban heat island-A review / Swarnalatha Kannoth, Najiya Nizar, Sheela A.M / Conference: International Conference on materials, mechanics and management At: College of Engineering Trivandrum – March 2020
4. Lin Liu, Yuanzhi Zhang. Urban Heat Island Analysis Using the Landsat TM Data and ASTER Data: A Case Study in Hong Kong [Електронний ресурс] / Lin Liu, Yuanzhi Zhang - Режим доступу: <https://pdfs.semanticscholar.org/1b78/f31b3b780a83166e836d6e4c899359e31c7a.pdf>
5. Чандра А.М., Гош С.К. Дистанційне зондування та географічні інформаційні системи / Чандра А.М., Гош С.К./ М. Техносфера, 2008. – . 305с.

АНАЛІЗ ВПЛИВУ ДИСТАНЦІЙНОГО НАВЧАННЯ НА ПРОЦЕС ФОРМУВАННЯ META SKILLS

Анотація. Описано процес формування гнучких навичок у студентів комп'ютерних спеціальностей. Припускається, що ефективність розвитку meta skills у майбутніх фахівців з інформаційних технологій може бути підвищено за допомогою технологій дистанційного навчання.

Ключові слова: компетенції, навик, soft skills, meta skills, дистанційне навчання, самоосвіта, IT-технології.

Вступ. В даний час система вищої освіти в Україні зіткнулася з серйозними випробуваннями, викликаними переходом до змішаної і дистанційною форми навчання, обумовленим складною епідеміологічною обстановкою в державі. Дистанційне навчання є методом ведення дидактичних процесів в умовах, коли викладач та студенти відокремлені один від одного (іноді суттєво) і не перебувають в одному місці під час передачі інформації, тому, окрім традиційних засобів спілкування, використовуються сучасні телекомунікаційні технології, за допомогою яких відбувається відправка голосових, відео, комп'ютерних даних та друкованих матеріалів [1].

У світовому просторі освітньою методологією виступає компетентнісний підхід у досягненні освітніх результатів, а компетентність, в свою чергу – мірилом освіченості людини, де акценти розставляються на результати навчання, здатності творчо застосовувати набуті знання, уміння, навички. Звичною практикою було розподілення навичок на професійні (hard skills) та гнучкі (soft skills), проте нова реальність змінює підходи як до процесу передачі знань, так і до визначення складу актуальних компетенцій фахівця, і це призводить до утворення нових категорій затребуваних навичок.

Постановка задачі. Здійснити аналіз впливу дистанційного навчання на формування актуальних видів міжпредметних навичок у майбутніх IT-фахівців.

Основний зміст роботи. Soft skills не залежить від специфіки конкретної роботи, тісно пов'язані з особистісними якостями і установками людини. Єдиної класифікації «гнучких навичок» не існує, але зазвичай сюди відносять: відповідальність, дисципліну і самодисципліну, комунікативні здібності, вміння працювати в команді, емоційний інтелект, лідерські якості, здатність до вирішення проблем.

У відповідь на виклики нового часу велике значення придбали мета-навички (meta skills), що займають проміжне положення між особистісними якостями і професійними навичками. Вперше термін «meta skills» був запропонований американським процесуальним терапевтом Е. Мінделл [2]. Навички з приставкою «мета-» – це усвідомленість, управління часом,

прогнозування і планування, особиста продуктивність, критичне мислення і оцінка, здатність до навчання і адаптивність, проактивність.

Мета-навички передують soft skills, це змістотворні, глибинні компетенції, спрямовані на подолання унікальних психологічних труднощів, з якими стикається сучасна людина. Але мета-навички не скасовують soft і hard skills, а дозволяють комплексно удосконалювати їх.

Оскільки на сьогодні конкурентоспроможний ІТ-фахівець повинен володіти як професійними, так і гнучкими навичками, це враховується у якості передбачуваних результатів навчання при організації навчальної діяльності студентів спеціальностей 122 «Комп'ютерні науки» та 121 «Інженерія програмного забезпечення». У дослідженні [3, 4] обґрунтовано використання методу проєктів як одного з обов'язкових механізмів розвитку soft skills у студентів галузі знань 12 «Інформаційні технології», і обрання у якості технологічного компоненту сервісів для управління проєктами (Asana, Trello, Jira, Microsoft Teams) та колективної ІТ-розробки (GitHub, Bitbucket, DeployBot, PhabricatorR, Beanstalk).

Впровадження дистанційної форми навчання у період загальнонаціонального карантину призвело до переосмислення та переоцінки засобів та методик традиційних моделей формування професійних компетенцій. За визначенням [5], дистанційне навчання – це така форма організації навчального процесу та педагогічна технологія, основою якої є керована самостійна робота студентів та широке застосування у навчанні сучасних інформаційно-комунікаційних технологій. Очевидно, що не тільки процеси набуття студентами hard та soft skills зазнали значної трансформації, а й визначення форм та засобів розвитку meta skills набуло нових перспектив.

На прикладі вивчення професійних компонентів освітньої програми студентів спеціальностей 122 «Комп'ютерні науки» та 121 «Інженерія програмного забезпечення» у дистанційному режимі можна виділити найбільш розповсюджені складнощі, притаманні саме дисциплінам ІТ-галузі, та проаналізувати шляхи їх подолання:

1. Неможливість наочної демонстрації виконання або процесу тестування застосунку. Студент наводить код застосунку у звіті з виконання лабораторної роботи, також там знаходяться скріншоти (знімки екрану) з результатами роботи застосунку з використанням тестових даних. Вихідні коди та файли програми архівуються та прикріплюються у якості додатків до роботи у системі дистанційної освіти. Але у випадку, коли потрібно провести перевірку роботи програми із використанням іншого набору вхідних даних, або визначити місце і причини виникнення помилки (некоректної роботи застосунку), може виникнути проблема з несумісністю програмного забезпечення або обчислювальної потужності комп'ютерів студента та викладача.

Шляхи вирішення проблеми: використання інструментів демонстрації екрану та технологій дистанційного управління пристроями. Студенти навчаються формулювати питання та обговорювати проблеми, що виникають у процесі створення та налагодження програмного забезпечення,

ознайомлюються з техніками та протоколами тестування. Студенти виконують пошук та аналіз інформації з різних джерел для самостійного вирішення проблеми.

2. Складнощі у орієнтуванні файлами програмного проекту студента, читання вихідного коду у випадку, якщо студент не використовує загальноприйнятих норм та практик іменування змінних, методів та інших складових елементів програми. При дистанційному навчанні, коли процес перевірки робіт студентів часто є асинхронним, викладач втрачає зайвий час на читання неправильно оформленого коду та пояснювального тексту, не маючи можливості задати уточнюючі питання.

Шляхи вирішення проблеми: вивчення правил оформлення та документування програмного коду та створення технічної документації. Формування навиків технічної комунікації, оформлення та презентації кінцевих та проміжних результатів роботи.

3. Необхідність самостійного встановлення та конфігурування програмного забезпечення, необхідного для виконання лабораторних та практичних робіт, на персональний комп'ютер студента. Студент не має змоги користуватися встановленим та потрібним чином налаштованим програмним забезпеченням на машинах у лабораторіях університету.

Шляхи вирішення проблеми: оволодіння навиками системного адміністрування. Студенти виконують пошук та аналіз інформації з різних джерел для самостійного вирішення проблеми, обмінюються релевантним досвідом.

Аналіз загального досвіду дистанційного навчання у студентів спеціальностей 121 та 122, дозволяє стверджувати, що впровадження дистанційних технологій у навчальний процес сприяє глибшому розумінню навчального матеріалу та формуванню таких навиків, як:

- комунікативні (безпосереднє відео-спілкування, письмове спілкування, точне та грамотне формулювання думок та питань);
- інформаційні (пошук інформації, її аналіз та критичне осмислення);
- здатність до вирішення проблем (вміння виявляти проблему та знаходити шляхи її вирішення, використовуючи логічне мислення та креативні підходи);
- керування часом;
- самодисципліна;
- самоосвіта (вміння навчатись самостійно).

Окрім того, дистанційне навчання сприяє формуванню таких якостей особистості, як активність, самостійність, самовдосконалення. Як показує практика, для успішного здійснення самостійної роботи здобувачу освіти необхідно навчитися самостійно приймати рішення, визначати зміст своєї навчальної діяльності та знаходити засоби її реалізації.

Як видно із приведеного вище переліку, навики та уміння, що найбільш ефективно здобуваються під час здійснення дистанційного навчання, переважно належать до категорії meta skills. Оскільки мета-навички є базою будь-якого комунікаційного процесу і забезпечують успішне застосування як

«м'яких», так і «жорстких» навичок, це свідчить про наявність тенденції до позитивного впливу дистанційного навчання на формування у студентів повного обсягу всебічних компетентностей.

Наукова новизна складається у вдосконаленні підходу до аналізу впливу дистанційної освіти на формування компетенцій здобувачів освіти галузі знань 12 «Інформаційні технології» шляхом розширення складу категорій міжпредметних навичок.

Висновки. Досвід провадження дистанційного навчання дає змогу стверджувати, що низка педагогічних та технологічних умов, що були створені, відіграють важливу роль у формуванні самоосвітньої діяльності студентів. Припускається, що ефективність розвитку meta skills у майбутніх фахівців з інформаційних технологій може бути підвищено за допомогою технологій дистанційного навчання.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Борисенко Н. Формування навичок самоосвітньої діяльності студентів педагогічних факультетів в умовах дистанційної освіти / Н. Борисенко, І. Гриценко, В. Денисенко, Н. Сидоренко // Social Work and Education, Vol. 7, No. 3. - Ternopil-Aberdeen, 2020. - pp. 325- 337.

2. Минделл А. Процессуальный ум. / Рипол Классик, 2016. – 408 с.

3. Глазунова О. Г., Волошина Т. В. Розвиток Софт Скілз у майбутніх фахівців з інформаційних технологій засобами гібридного інформаційно-освітнього середовища університету / Інноваційні комп'ютерні технології у вищій школі: матеріали 8-ої Науково-практичної конференції - Львів: Видавництво Наукового товариства ім. Шевченка, 2016.

4. Глазунова О.Г. Розвиток «Soft skills» у майбутніх фахівців з інформаційних технологій: методи, засоби, індикатори оцінювання / О.Г. Глазунова, Т.В. Волошина, І.В. Корольчук /Open educational e-environment of modern University, special edition, 2019.

5. Даценко Г.В., Сузанська З.В. Дистанційне навчання як засіб стимулювання самоосвіти / Дистанційне навчання як сучасна освітня технологія [Електронний ресурс]: матеріали міжвузівського вебінару (м. Вінниця, 31 березня 2017 р.) / відп. ред. Л.Б. Ліщинська. – Вінниця : ВТЕІ КНТЕУ, 2017. – с. 17-19.

РОЗДІЛ 2

ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У СФЕРІ ОСВІТИ, НАУКИ І УПРАВЛІННЯ ВИРОБНИЦТВОМ

УДК 681.5.08:622.412.13

М.О. Алексєєв¹, О.В. Голінько¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

МЕТОДИ, АЛГОРИТМИ ТА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДІАГНОСТИКИ ПРАЦЕЗДАТНОСТІ СТАЦІОНАРНИХ ТЕРМОКАТАЛІТИЧНИХ ГАЗОАНАЛІЗАТОРІВ.

Анотація. Наведено результати теоретичних та експериментальних досліджень з розробки методу автоматичного контролю чутливості стаціонарних аналізаторів метану, шляхом аналізу їх вихідних характеристики при зміні струму через чутливі елементи на дільниці виходу вихідної напруги вимірювального моста на горизонтальну частину (плато) характеристики.

Ключові слова: *вугільні шахти, вміст метану, контроль, аналізатор, автоматична діагностика, первинні перетворювачі, C, C++, чутливість.*

Вступ. Складні гірничо-геологічні умови вугільних родовищ України, значна газовість, схильність вугілля до самозаймання, раптові викиди вугілля та газу обумовлюють значний ризик виникнення аварій, у тому числі таких небезпечних подій як вибухи метаноповітряних сумішей. Однією з причин вибухів є неспрацьовування шахтних стаціонарних аналізаторів метану за наявності недопустимої концентрації метану в атмосфері гірничих виробок шахт [1], що обумовлюється недостатньою їх надійністю.

Зниження ймовірності вибуху метаноповітряних сумішей досягається при збільшенні ймовірності безвідмовного функціонування системи контролю метану і захисного вимкнення напруги. Це пояснюється тим, що ймовірність безвідмовної роботи є зворотною функцією ймовірності відмови і в разі збільшення ймовірності безвідмовної роботи системи контролю метану зменшується ймовірність її відмови, що призводить до зменшення ймовірності загазування виробок і як результат до зменшення ймовірності вибуху метану у виробках [2]. В свою чергу важливе значення для збільшення ймовірності безвідмовної роботи апаратури має збільшення частоти профілактичних заходів із встановлення її працездатності оскільки це призводить до зменшення часу, на який розраховується безвідмовність.

Важливим заходом контролю справності роботи апаратури контролю метану є пробні перевірки її спрацювання. Нині для перевірки справності системи контролю метану застосовують випробувальну суміш метану з повітрям. Таку перевірку виконують спеціально навчені працівники.

Встановлено, що щоденна перевірка справності апаратури контролю метану у порівнянні зі щомісячною перевіркою дозволяє знизити ймовірність вибуху у 300 разів, а за умови зменшення терміну між перевірками апаратури до 4 годин ймовірність вибуху за рік може бути зменшена до 10-6 [2]. В той же час щоденна перевірка справності апаратури контролю метану в шахтних умовах з використанням атестованих газових сумішей, обумовлює значні експлуатаційні витрати. Зважаючи на це, актуальною задачею є обґрунтування методів автоматичної перевірки апаратури.

Постановка задачі. Метою публікації є підвищення надійності контролю вибухонебезпечності рудникової атмосфери та зменшення витрат на обслуговування стаціонарних аналізаторів метану за рахунок застосування автоматичної діагностики та корегування чутливості первинних перетворювачів стаціонарних термокаталітичних аналізаторів метану.

Основний зміст роботи. Спираючись на проведені дослідження [3, 4] вдосконалено алгоритм та програму роботи аналізатора з функцією автоматичної діагностики нульових показань, реалізованого на базі мікропроцесора ATMEGA8 [5], що полягає в доповненні програми його роботи модулем, який забезпечує автоматичну діагностику чутливості термокаталітичного датчика та корегування показань аналізатора при її зміні.

Сутністю процесу діагностики чутливості термокаталітичного датчика та корегування показань аналізатора при її зміні полягає в наступному. В режимі налагодження, до виносного датчика аналізатора подається атестована газова суміш з вмістом метану 1,0%, шляхом зміни струму через чутливі елементи (для дослідженого типу датчиків в діапазоні від 180 до 200 мА) визначається величина струму яка відповідає моменту виходу вихідної напруги моста на плато характеристики (рис. 1), визначається та запам'ятовується відносна величина зменшення вихідної напруги вимірювального моста аналізатора $\Delta U_{\text{вн}}$ при зменшенні струму через елементи на 5% відносно величини струму, що відповідає моменту виходу вихідної напруги моста на плато представленої характеристики (1):

$$\Delta U_{\text{вн}} = \frac{U_{1\text{н}} - U_{2\text{н}}}{U_{1\text{н}}}, \quad (1)$$

де $U_{1\text{н}}$ та $U_{2\text{н}}$ – відповідно значення вихідної напруги вимірювального моста в режимі налаштування при роботі на плато характеристики та зменшенні струму через елементи на 5%.

Фіксація моменту виходу вихідної напруги моста на плато характеристики здійснюється тоді, коли збільшення струму на 1 мА не призводить до зростання вихідної напруги моста більше ніж на 1%. Значення встановленої величини струму $I_{\text{ен}}$ і $t_{\text{гн}}$ температура газу при налаштуванні аналізатора заноситься в пам'ять аналізатора і використовується в подальшому для налаштування робочого режиму аналізатора.

В робочому режимі безперервно виконуються підтримка сталого температурного режиму термоелементів при зміні температури газу що для

робочого елементу є сталою величиною і для досліджуваних термоелементів пелістерного типу становить біля $9 \cdot 10^3 \text{ Ом} \cdot \text{°C/Вт}$. Для забезпечення сталого температурного режиму термоелементів при зміні температури газу, величина струму через елементи повинна коригуватися як (2) [4]:

$$I_e = \sqrt{I_{eH}^2 - \frac{1}{b}(t_{\Gamma} - t_{\Gamma H})}, \quad (2)$$

де $I_{eH}, t_{\Gamma H}$ – відповідно значення величини струму і температура газу при налаштуванні аналізатора.

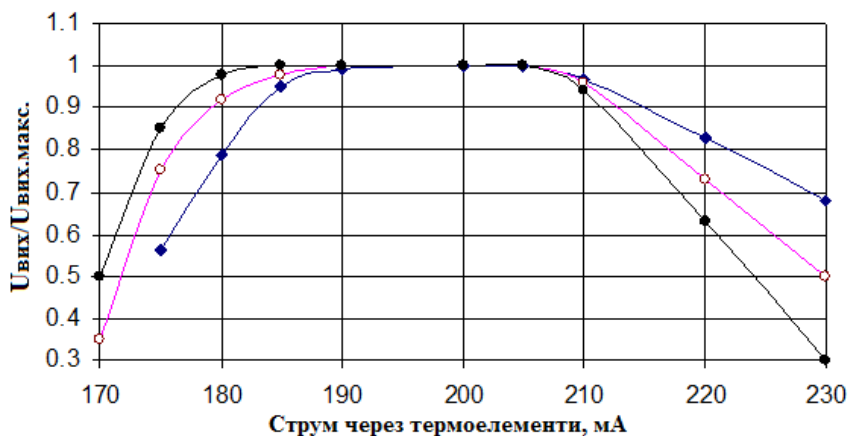


Рис. 1. Вихідні характеристики датчика з подвійним дифузійним фільтром ($C=1,02\text{об.}\%$): \blacklozenge - $\gamma_e = 2\gamma_{отв}$; \circ - $\gamma_e = 4\gamma_{отв}$; \bullet - $\gamma_e = 6\gamma_{отв}$ [6]

Таким чином здійснюються операції вимірювання напруг на елементах, визначення різниці напруг на елементах, розрахунок та передача інформації про вміст метану. За зовнішнім запитом оператора чи періодично, згідно визначеного програмою роботи аналізатора часу, здійснюється перевірка чутливості датчика та корегування показань аналізатора при її зміні. Для цього величина струму через чутливі елементи зменшується на 5% відносно встановленого робочого режиму та визначається відносна величина зменшення вихідної напруги вимірювального моста аналізатора при цьому.

Визначається відносна величина зменшення вихідної напруги $\Delta U_{вр}$, згідно за формулою (3), і порівнюється з $\Delta U_{вн}$. Якщо $\Delta U_{вр} = \Delta U_{вн}$, то у датчика зберігається початкова величина чутливості і корегування її не потрібне. При зниженні $\Delta U_{вр}$ відносно $\Delta U_{вн}$ до 10% здійснюється коригування чутливості аналізатора, шляхом її збільшення на величину $\Delta U_{вн} - \Delta U_{вр}$. В іншому випадку надсилається повідомлення оператору про необхідність здійснення оперативної перевірки аналізатора.

$$\Delta U_{вр} = \frac{U_{1р} - U_{2р}}{U_{1р}}, \quad (3)$$

де U_{1p} та U_{2p} – відповідно значення вихідної напруги вимірювального моста в робочому режимі при роботі на плато характеристики та зменшенні струму через елементи на 5%.

Слід відмітити, що при зміні режиму живлення чутливих елементів датчика виникають перехідні теплові і газодинамічні процеси, тому зняття показань повинне здійснюватися після закінчення перехідного процесу, що становить для досліджуваних первинних перетворювачів біля 5 секунд з моменту зниження струму через термоелементи. В шахтних умовах можлива ситуація, що за цей час змінилась концентрація метану, тому визначення відносної величини зменшення вихідної напруги $\Delta U_{вр}$ необхідно проводити декілька раз, а потім шляхом обробки даних вибирати ті, що неодноразово повторюються і вибраковувати випадкові результати.

Наукова новизна полягає в управлінні електроживленням чутливих елементів аналізаторів метану для визначення зміни чутливості первинних перетворювачів і автоматичної її корекції.

Висновки. Виконані дослідження дозволили обґрунтувати метод автоматичної діагностики стану стаціонарних термokatалітичних аналізаторів метану в частині автоматичного дистанційного контролю чутливості первинних перетворювачів шляхом аналізу їх вихідних характеристики при зміні струму через чутливі елементи на дільниці виходу вихідної напруги моста на плато характеристики. Показано, що при використанні сучасних мікропроцесорних засобів обробки інформації є можливість не тільки автоматично перевіряти стабільність чутливості первинних перетворювачів, а і здійснити корегування показань газоаналізатора при зміні чутливості. Для реалізації результатів досліджень розроблено алгоритм і програмна реалізація роботи аналізатора з функцією автоматичного дистанційного контролю чутливості термokatалітичних датчиків метану.

ПЕРЕЛІК ПОСИЛАНЬ:

1. НПАОП 10.0-1.01-10 Правила безпеки у вугільних шахтах. – К.: Друкарня ДП «Редакція журналу «Охорона праці» 2010. – 430 с.
2. Колосюк А.В., Колосюк В.П. Вибухонебезпека систем електропостачання гірничих машин на шахтах, небезпечних за раптовими викидами вугілля і газу. Монографія. – Кременчук: КрНУ імені Михайла Остроградського, 2017. – 178 с.
3. Алексєєв М.О., Голінько О.В. Автоматична діагностика стану стаціонарних термokatалітичних газоаналізаторів / Збірник наукових праць НГУ. – Д.: ДВНЗ «НГУ», 2018 – № 53 – С. 223-229
4. Алексєєв М.О., Голінько О.В. Автоматичний контроль чутливості датчиків стаціонарних термokatалітичних аналізаторів метану / Гірничий вісник. – Кривий Ріг КНУ, 2020 – № 107 – С. 16-22
5. Голинько В.И., В.В. Белоношко Исследование процесса окисления метана в термokatалитических датчиках / Научный вестник НГУ. - Дніпропетровськ, 2003 - №7, с. 62-65.

6. Голинько В.И., Белоножко В.В. Совершенствование термokatалитических средств контроля содержания метана / Гірничча електромеханіка та автоматика: Наук. техн. зб. – 2003. – Вип. 70. – С. 92 – 100.

УДК 004.415.3:681.6

О.Ю. Гусев¹, А.В. Кожевніков¹, Д.С.Тимофеев¹, В.І. Яриловець¹
¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ЦИФРОВОГО МОДЕЛЮВАННЯ, РОЗРАХУНКУ ТА ВІЗУАЛІЗАЦІЇ ФОРМУЄМИХ В СКЛАДНИХ ОБ'ЄКТАХ ТЕМПЕРАТУРНИХ ПОЛІВ

Анотація. Описано процес розробки інформаційної технології, призначеної для покращення вивчення процесів теплообміну. Покращення досягається за рахунок будівництва найбільш ефективної системи моделювання із застосуванням методів теорії теплопровідності та методу кінцевих різниць.

Ключові слова: програмне забезпечення, теплообмін, теплове поле, термограма, теплопровідність, візуалізація, C#, OpenGL, Tao Framework.

Вступ. За останні десятиліття сфера інтенсивного дослідження та застосування явищ теплообміну надзвичайно розширилася. В даний час спостерігається значний прогрес в розвитку обчислювальних методів вирішення завдань для рівнянь в приватних похідних і збільшення потужності сучасних обчислювальних машин. Виходячи з цього, теоретичне дослідження процесів теплообміну в даний час в значній мірі базується на їх чисельному моделюванні з використанням ЕОМ. У сучасних умовах виробництва математичне моделювання грає дуже важливу роль, виступаючи як засіб оптимізації та прогнозування. Зокрема, моделювання теплових полів є однією з найбільш актуальних проблем при проектуванні електронних компонентів. Візуалізація теплових полів, тобто отримання термограми, являє собою заключну частину моделювання теплових процесів. Тобто при створенні спеціальних засобів розрахунку теплових параметрів і моделювання теплових процесів, розробник зіткнеться з завданням візуалізації теплових полів. Тому розробка в даній сфері буде актуальною ще довгий час.

Постановка задачі. Для досягнення поставленої мети сформовані та вирішені такі завдання: викласти принципи процесів теплообміну; дослідити особливості реалізації математичної моделі теплових процесів; проаналізувати інструменти моделювання процесів теплообміну, які доступні в MS Visual Studio; спроектувати та розробити відповідне програмне забезпечення, яке можна застосовувати на різних конфігураціях ОС; зробити висновки щодо доцільності створення подібної системи.

Основний зміст роботи. Під час роботи над даною роботою були використані наступні методи та інструменти: методи теорії теплопровідності; вирішення системи лінійних алгебраїчних рівнянь (СЛАР) методом кінцевих різниць; Visual Studio, C#, .NET Framework, OpenGL, Tao Framework.

Теоретичне дослідження процесів теплообміну в даний час в значній мірі базується на їх чисельному моделюванні з використанням ЕОМ. Це стало можливим завдяки значному прогресу в розвитку обчислювальних методів вирішення завдань для рівнянь в приватних похідних і збільшення потужності сучасних обчислювальних машин. Потрібно особливо відзначити ще такі обставини. Чисельне моделювання процесів теплообміну в даний час набуває все більш значну роль в зв'язку з тим, що для сучасної науки і техніки необхідний достовірний прогноз таких процесів, експериментальне вивчення яких в лабораторних або натурних умовах дуже складно і дорого, а в деяких випадках просто неможливо. Чисельне моделювання процесів теплообміну все успішніше входить в практику роботи різних науково-дослідних, проектно-конструкторських і виробничих установ.

У сучасних умовах виробництва математичне моделювання грає дуже важливу роль, виступаючи як засіб оптимізації та прогнозування. Моделювання теплових полів є, зокрема, однією з найбільш актуальних проблем при проектуванні електронних компонентів. Візуалізація теплових полів, тобто отримання термограми, являє собою заключну частину моделювання теплових процесів. Іншими словами, при створенні спеціальних засобів розрахунку теплових параметрів і моделювання теплових процесів, розробник зіткнеться з завданням візуалізації теплових полів. Тому функціональне призначення даної системи полягає в наданні користувачу зручного та надійного шляху подивитися та проаналізувати розподіл температурних полів за допомогою термограми. На рис. 1 наведена структура розробленої системи та алгоритмів її функціонування у вигляді UML-діаграми.

Для виконання даної роботи використані бібліотеки Tao Framework і OpenGL для C#. OpenGL – це середовище для розробки портативних, інтерактивних 2D та 3D графічних додатків. З моменту своєї появи в 1992 році OpenGL став найбільш широко використовуваним і підтримуваним в галузі інтерфейсом 2D і 3D графічного програмування програм (API). Tao Framework – це вільно розповсюджена бібліотека з відкритим вихідним кодом, призначена для побудови та зручної розробки крос-платформного мультимедійного програмного забезпечення у середі .NET Framework і Mono. На сьогоднішній день Tao Framework – це один із найкращих шляхів для використання бібліотек OpenGL при розробці у середі .NET на мові C#.

Розроблений програмний компонент складається з чотирьох основних частин - файлу з вихідним масивом, призначеного для користувача інтерфейсу, модуля візуалізації та модуля розрахунку кольору.

Вихідний масив – це матриця температур модельованого кристала, розрахована за допомогою методів теорії теплопровідності та методу кінцевих різниць системи лінійних алгебраїчних рівнянь, зберігається в окремому файлі.

Модуль призначеного для користувача інтерфейсу відображає вихідний масив як у вигляді тексту, так і у вигляді термограми. У цьому модулі передбачені функції збереження термограми в графічному форматі.

У модулі візуалізації відбуваються ініціалізація компонентів бібліотеки OpenGL та відображення вихідного масиву в графічному вигляді. Цей модуль

зчитує вміст комірки вихідного масиву і передає це значення в модуль розрахунку кольору, а у відповідь отримує RGB-параметри кольору, відповідного температурі даного осередку. На основі колірних параметрів кожного осередку будується колірний градієнт, який і є термограмою.

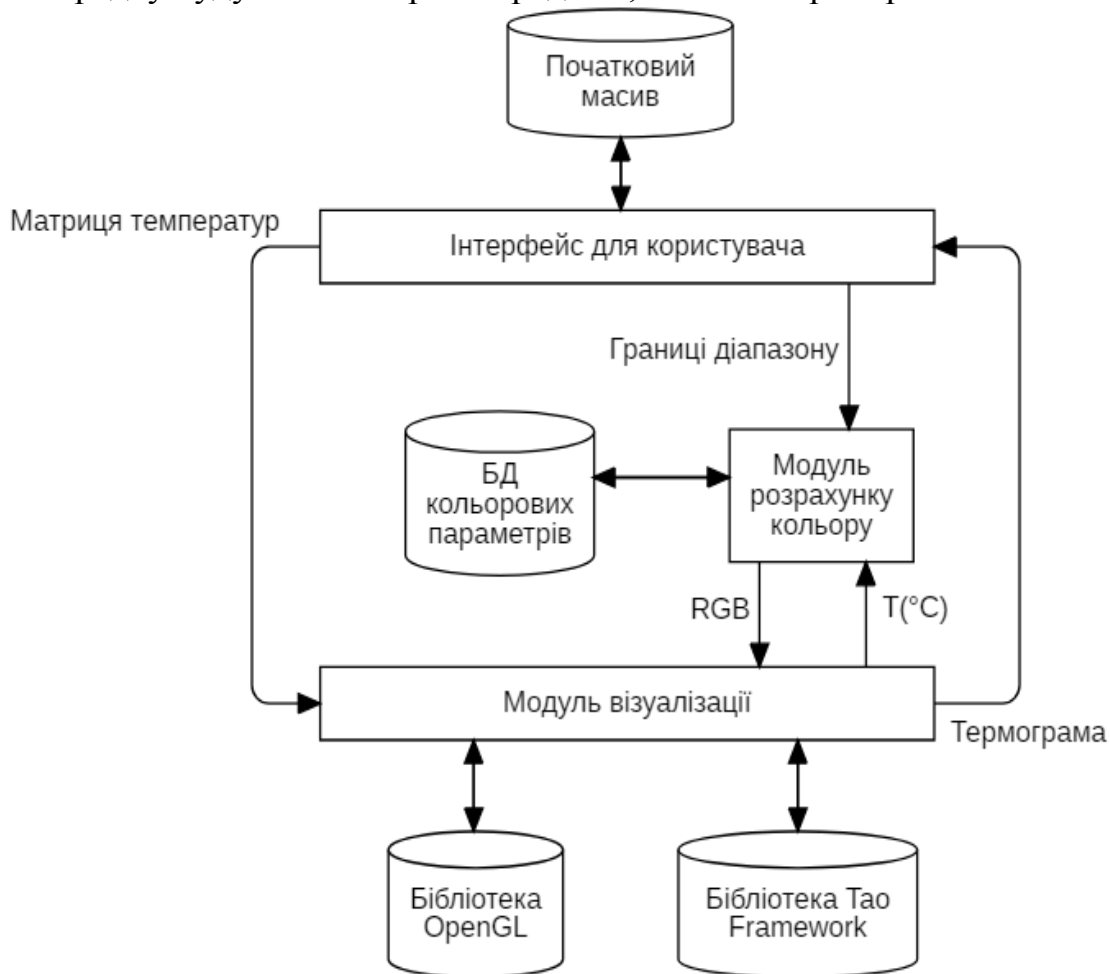


Рис. 1. UML-діаграма структури системи

Модуль розрахунку кольору визначає RGB-складові кольору, які відповідають вхідному параметру температури. Верхня і нижня межі температури визначають інтервал, на якому буде відбуватися візуалізація. Температурі, що виходить за нижню межу, відповідає чорний колір. Температурі, що виходить за верхню межу, відповідає білий колір. У середині інтервалу колірної градієнт змінюється по зростанню температури в такій послідовності: чорний, синій, фіолетовий, червоний, помаранчевий, жовтий, білий. Після натискання кнопки «Рассчитать» з'явиться результат на екрані, прикладом якого є рис. 2.

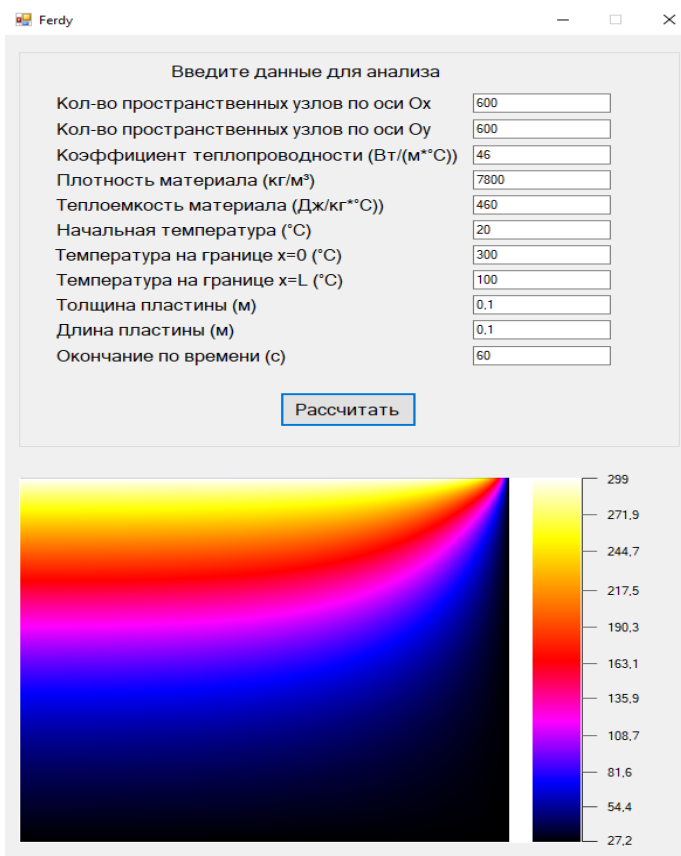


Рис. 2. Кінцевий результат роботи програми візуалізації теплових полів

Наукова новизна в основному полягає в подальшому удосконаленні сучасних методів та алгоритмів цифрового моделювання та візуалізації оперативно формуємих теплових полів в складних об'єктах різного технологічного призначення на підприємствах.

Висновки. Розроблено програмне забезпечення для візуалізації теплових полів, яке дозволяє ефективно та швидко аналізувати, розраховувати та візуалізувати температурні поля у вигляді термограми. У тому числі, були зроблені дослідження, що призвели до часткового удосконалення в області цифрового моделювання процесів теплообміну.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Лыков А.В. Теория теплопроводности.– М.: Высшая школа, 1967. – 600с.
2. Кузнецов Г.В., Шеремет М.А. Разностные методы решения задач теплопроводности: учебное пособие. / Г.В. Кузнецов, М.А. Шеремет. – Томск: Изд-во ТПУ, 2007. – 172 с.
3. Методы и алгоритмы моделирования тепловых полей в трехмерной сборке интегральных схем [Текст] / В.Ф. Барабанов, С.Л. Подвальный, А.В. Ачкасов, М.Н. Аралов // Радиотехника. – 2014. - №6 – С. 82-87.
4. Вержбицкий В.М. Основы численных методов. - М.: Высшая школа, 2002. – 840 с.
5. Райт мл. Р.С., Липчак Б. OpenGL. Суперкнига = OpenGL SuperBible. — 3 изд. — М.: Вильямс, 2006. — С. 1040. — ISBN 5-8459-0998-8.

АЛГОРИТМИ ТА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ РОЗПІЗНАВАННЯ ОБЛИЧЧЯ НА ОСНОВІ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ

Анотація. Розроблена комп'ютерна система з розпізнавання обличчя в режимі реального часу з вебкамери на основі нейронних мереж на мові JavaScript. Присутня функція розпізнавання конкретної особи на основі наявних фото.

Ключові слова: згорткова нейронна мережа, розпізнавання обличчя, режим реального часу, розпізнавання образів, javascript.

Вступ. В даний час спостерігається неминущий інтерес до проблеми розпізнавання обличчя. Розпізнавання обличчя можна умовно поділити на 2 завдання: ідентифікація і верифікація. Завдання ідентифікації полягає в тому, щоб порівняти захоплене обличчя з усіма зображеннями осіб, що зберігаються в базі даних. Мета верифікації полягає в тому, щоб порівняти обличчя людини з фотографією кандидата, з особою на еталонній фотографії, що зберігається в відповідній базі даних.

В даний час для ідентифікації людини використовуються біометричні методи. Біометрією називається сукупність способів і пристроїв для ідентифікації людини, які засновані на його унікальних фізіологічних або поведінкових характеристиках. У зв'язку з простотою розпізнавання особи і великою кількістю камер у всіх аспектах життя людини, все більш актуальним стають розробки в розпізнаванні осіб. Завдання розпізнавання осіб актуальна як в області інтелектуальних середовищ, так і в системах безпеки.

Розпізнавання обличчя можна розбити на три пункти: знайти особу в режимі реального часу; порівняти знайдену особу з особами, що зберігаються в базі даних; порівняти знайдену особу з еталонним в базі даних.

Завдання розпізнавання осіб актуальна як в області інтелектуальних середовищ, так і в системах безпеки.

Постановка задачі. Для досягнення цілі потрібно виконати наступні завдання: проаналізувати існуючі підходи для розпізнавання осіб; виявити класифікацію алгоритмів розпізнавання; провести аналіз існуючих на ринку систем розпізнавання, виявити їхні переваги і недоліки; проаналізувати основні інструментальні засоби для розробки і вибрати оптимальні з них; спроектувати і програмно реалізувати систему розпізнавання.

Основний зміст роботи. Під час роботи над даною роботою були використані наступні методи та інструменти.

- Бібліотека FaceApi та деякі елементи згорткової нейронної мережі Mobilnetv1;
- Активні моделі зовнішнього вигляду (Active Appearance Models, ААМ);

- JavaScript, Visual Studio Code, Open Server.

В даний час існує близько десятка різновиди нейронних мереж (НМ). Один з найбільш широко використовуваних варіантів це мережа, побудована на багат шаровому перцептроні, яка дозволяє класифікувати подане на вхід зображення / сигнал відповідно до попередніх налаштувань / навчань мережі.

Навчаються нейронні мережі на наборі навчальних прикладів. Суть навчання зводиться до налаштування ваг міжнейронних зв'язків в процесі рішення оптимізаційної задачі методом градієнтного спуску. В процесі навчання НМ відбувається автоматичне вилучення ключових ознак, визначення їх важливості та побудова взаємозв'язків між ними. Передбачається, що навчена НМ зможе застосувати досвід, отриманий в процесі навчання, на невідомі образи за рахунок узагальнюючих здібностей.

Найкращі результати в області розпізнавання осіб (за результатами аналізу публікацій) показала Convolutional Neural Network або згорткова нейронна мережа (далі - ЗНМ), яка є логічним розвитком ідей таких архітектур НМ як когнітрон і неокогнітрон. Успіх обумовлений можливістю обліку двовимірної топології зображення, на відміну від багат шарового перцептрона.

Відмінними рисами ЗНМ є локальні рецепторні поля (забезпечують локальну двовимірну зв'язність нейронів), загальні ваги (забезпечують детектування деяких рис в будь-якому місці зображення) і ієрархічна організація з просторовим семплінгом (spatial subsampling). Завдяки цим нововведенням ЗНМ забезпечує часткову стійкість до змін масштабу, зсувів, поворотів, зміні ракурсу і інших спотворень.

Тестування ЗНМ на базі даних ORL, що містить зображення осіб з невеликими змінами освітлення, масштабу, просторових поворотів, положення і різними емоціями, показало 96% точність розпізнавання.

Свій розвиток ЗНМ отримали в розробці DeepFace, яку придбав Facebook для розпізнавання осіб користувачів своєї соцмережі. Всі особливості архітектури носять закритий характер.

У звичайному перцептроні, який представляє собою повнозв'язну нейронну мережу, кожен нейрон пов'язаний з усіма нейронами попереднього шару, причому кожний зв'язок має свій персональний ваговий коефіцієнт. У згортковій нейронній мережі в операції згортки використовується лише обмежена матриця ваг невеликого розміру, яку «рухають» по всьому оброблюваному шару (на самому початку – безпосередньо по вхідному зображенню), формуючи після кожного зсуву сигнал активації для нейрона наступного шару з аналогічною позицією. Тобто для різних нейронів вихідного шару використовуються одна і та ж матриця ваг, яку також називають ядром згортки. Її інтерпретують як графічне кодування якої-небудь ознаки, наприклад, наявність похилої лінії під певним кутом. Тоді наступний шар, що вийшов в результаті операції згортки такою матрицею ваг, показує наявність даної ознаки в оброблюваному шарі і її координати, формуючи так звану карту ознак (англ. Feature map).

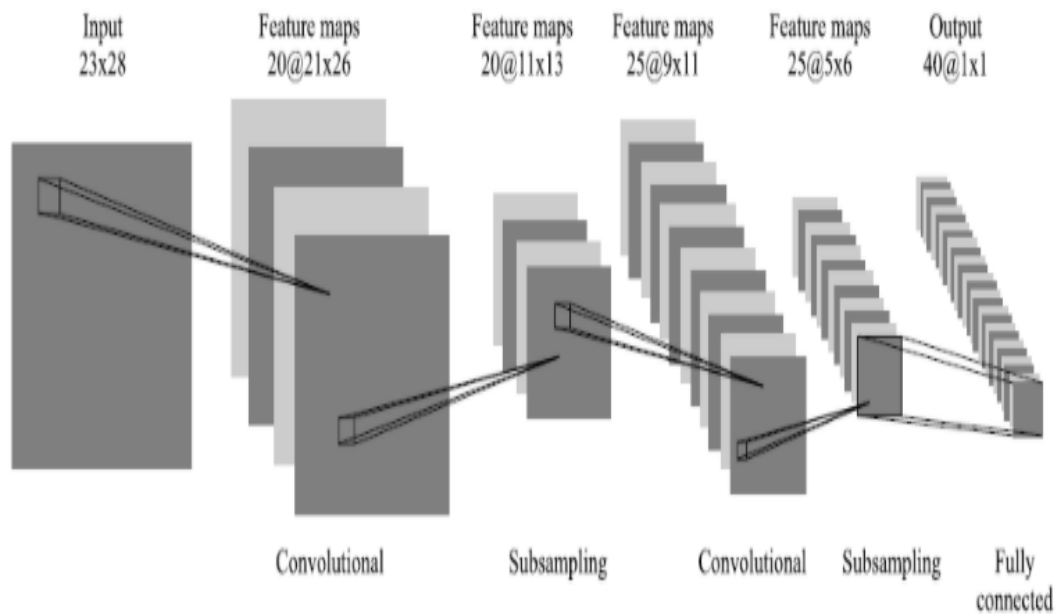


Рис. 1. Схематичне зображення архітектури згорткової нейронної мережі

Природно, в згортковій нейронній мережі набір ваг не один, а ціла гама, що кодує елементи зображення (наприклад лінії і дуги під різними кутами). При цьому такі ядра згортки не закладаються дослідником заздалегідь, а формуються самостійно шляхом навчання мережі класичним методом зворотного поширення помилки. Прохід кожним набором ваг формує свій власний примірник карти ознак, роблячи нейронну мережу багатоканальною (багато незалежних карт ознак на одному шарі). Також слід зазначити, що при переборі шару матрицею ваг її пересувають зазвичай не на повний крок (розмір цієї матриці), а на невелику відстань. Так, наприклад, при розмірності матриці ваг 5×5 її зрушують на один або два нейрона (пікселя) замість п'яти, щоб не «переступити» шукану ознаку.

Дана програма була реалізована на основі веб інтерфейсу в форматі html/php з використанням коду на мові JavaScript з підключенням бібліотеки Facearі. При створенні системи був використований редактор коду Visual Code.

В наступному розділі був проведений експеримент з порівнянням системи з системою бібліотеки комп'ютерного зору OpenCV.

Підсумовуючи результати експериментів можна сказати, що розроблена система на нейронних мережах показала себе краще ніж система на OpenCV. В експерименті на близькій дистанції обидві системи впорались з розпізнаванням. Проте в інших експериментах система на OpenCV мала певні проблеми з розпізнаванням.

Обсяг використовуваної пам'яті був більший на системі OpenCV (приблизно 300мб проти в середньому 80 мб на розробленій системі), а час розпізнавання був також більший на цій системі. Також варто зазначити, що в системі OpenCV спостерігались значні проблеми з FPS(кількість кадрів в секунду), коли на системі з нейронними мережами такої проблеми не було.

Якщо казати про переваги системи на OpenCV, то варто зазначити, що таку систему простіше розробляти, через простоту методів бібліотеки комп'ютерного зору.

– Спроекувати і програмно реалізувати систему розпізнавання.

В результаті були вирішені наступні задачі:

- система виявляє обличчя досить точно;
- система виявляє емоції людини;
- система виявляє рік та стать людини;
- система розпізнає саму людину;
- система працює навіть в поганих умовах.

Система розпізнавання обличчя необхідна користувачеві в першу чергу для забезпечення безпеки аутентифікації. Сьогодні приватний замовник в основному готовий платити за безпеку. Завдання розпізнавання осіб актуальна як в області інтелектуальних середовищ, так і в системах безпеки.

Другий важливий аспект – це інтуїтивно зрозумілий інтерфейс. Система була розроблена на мові JavaScript та середовище розробки Visual Code. Була взята архітектура згорткової нейронної мережі MobilNet та були використані деякі функції для покращення роботи системи розпізнавання обличчя.

Наукова новизна: Удосконалено та оптимізовано методи розпізнавання обличчя за допомогою нейронних мереж та засобів програмування JavaScript.

Висновки. В ході дослідження було виконано розробку комп'ютерної системи розпізнавання обличчя на основі згорткової нейронної мережі. Поставлені задачі були виконані, система досить добре впоралась з експериментами.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Face Recognition by Elastic Bunch Graph Matching: CRC Press, ISBN 0-8493-2055-0, Chapter 11, pp. 355-396, (1999) – режим доступу: <https://www.face-rec.org/algorithms/EBGM/WisFelKrue99 FaceRecognition-JainBook.pdf>.

2. Distortion Invariant Object Recognition in the Dynamic Link Architecture: IEEE TRANSACTIONS ON COMPUTERS, VOL. 42, NO 3, MARCH 1993 – режим доступу: https://www.researchgate.net/publication/3043085_Distortion_Invariant_Object_Recognition_in_the_Dynamic_Link_Architecture.

3. Face Recognition: A Convolutional Neural-Network Approach: IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 8, NO. 1, JANUARY 1997 – режим доступу: <https://pdfs.semanticscholar.org/ee1e/322b5f8f15ad3fcd17762fba3da209b0c484.pdf>.

ПРАКТИЧНА ЕФЕКТИВНІСТЬ ВИРІШЕННЯ ЗАДАЧІ ДОСЛІДЖЕННЯ ТЕМПЕРАТУРНИХ ПОЛІВ ПЛАСТИН МЕТОДОМ КІНЦЕВИХ РІЗНИЦЬ

Анотація. Розглянуто аналітичне обґрунтування програмному забезпеченню дослідження розповсюдження температурних полів в пластинах на основі використання методу кінцевих різниць.

Ключові слова: температурні поля, двухточечні крайові задачі, методу кінцевих різниць, дискретні апроксимації, ефективне стійке рішення.

Вступ. Точне рішення крайових задач викликає дуже великі труднощі. Звідси існує підвищений інтерес та велика різноманітність наближених методів оптимізації вирішення завдань. Метод кінцевих різниць є одним з таких методів і був обраний для вирішення поставленого завдання розрахунку температурних полів пластин.

Постановка задачі. Метою являється дослідження на практиці ефективності вирішення поставленої задачі методом кінцевих різниць. Розглядаються двухточечні крайові задачі для звичайних диференціальних рівнянь другого порядку в загальному вигляді.

Основний зміст роботи. Ідея методу кінцевих різниць (МКР) рішення крайових задач досить проста і видно вже з самої назви: замість похідних в диференціальному рівнянні використовуються їх кінцеворізницеві апроксимації. При побудові дискретних апроксимацій крайових диференціальних завдань потрібно прагнути пов'язати дві, можливо, суперечливі цілі: гарна якість апроксимації і ефективне стійке рішення утворюються у своїй алгебраїчних систем. Нехай дана функція, графік якої представлений на рис. 1.

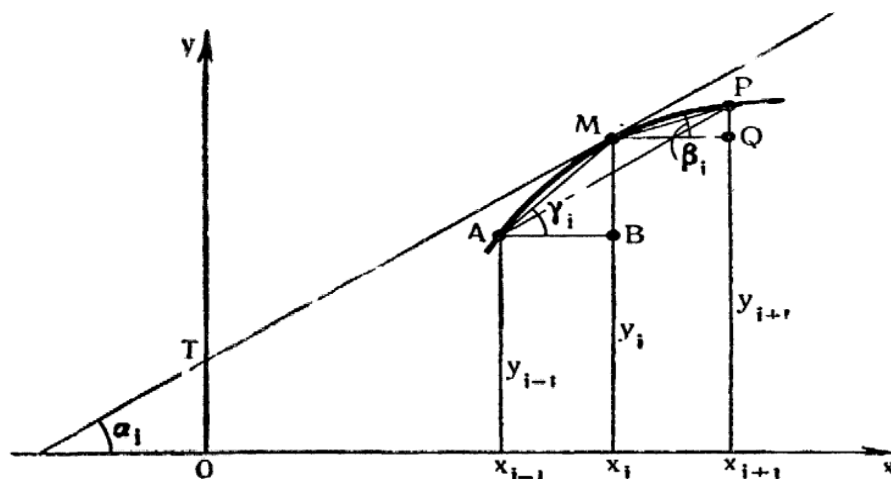


Рис. 1. Графік функції $y = f(x)$ та її похідної.

Якщо через α_i позначити кут, утворений з позитивним напрямком осі абсцис дотичної до кривої, проведеної в точці $M(x_i, y_i)$, то похідна функція при $x = x_i$ визначається за формулою

$$y'_i = \tan \alpha_i. \quad (1)$$

Візьмемо на кривій дві сусідні точки $A(x_{i-1}, y_{i-1})$ та $P(x_{i+1}, y_{i+1})$ так, щоб різниці $x_i - x_{i-1} = x_{i+1} - x_i = h$ були б досить малі, і наближено замінимо α_i на β_i чи γ_i (або розглянемо замість дотичної МТ одну з січних МР чи АМ). З цього виходить, що

$$y'_i \approx \operatorname{tg} \beta_i = \frac{QP}{MQ} = \frac{y_{i+1} - y_i}{h}, \quad (2)$$

або

$$y'_i \approx \operatorname{tg} \gamma_i = \frac{BM}{AB} = \frac{y_i - y_{i-1}}{h}. \quad (3)$$

Якщо ж кутовий коефіцієнт дотичної МТ наближено замінити кутовим коефіцієнтом січної АР, то

$$y'_i \approx \frac{y_{i+1} - y_{i-1}}{2h}. \quad (4)$$

Праві частини формул (2)-(3) називаються відповідно: різницеvim відношенням вперед, різницеvim відношенням назад і симетричним різницеvim відношенням.

Наближене значення другої похідної y''_i функції $y = f(x)$ при $x = x_i$ може бути отримано, якщо замінити криву на ділянці АР ламаною лінією АМР, що має в точці М два нахилу, тобто

$$y''_i \approx \frac{1}{h} \left(\frac{y_{i+1} - y_i}{h} - \frac{y_i - y_{i-1}}{h} \right) = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}. \quad (5)$$

Варто сказати, що наведені формули (2)-(5) для заміни похідних різницеvim відносинами не є єдино можливими. Часом потрібно проводити інші заміни, проте при чисельному інтегруванні рівнянь теплопровідності найбільш часто застосовують саме ці формули.

Метод кінцевих різниць дозволяє вирішувати системи диференціальних рівнянь теплопровідності як при постійних, так і при змінних коефіцієнтах, а також одновимірні, двох- і тривимірні задачі.

Проаналізуємо процес теплопереносу у пластині (рис. 2).

Для рішення даної задачі маємо пластину з довжиною L та висотою H . Горизонтальні границі є адіабатичними, а на вертикальних границях підтримуються постійні температури T_h та T_c , при цьому початкова температура області вирішення – T_0 .

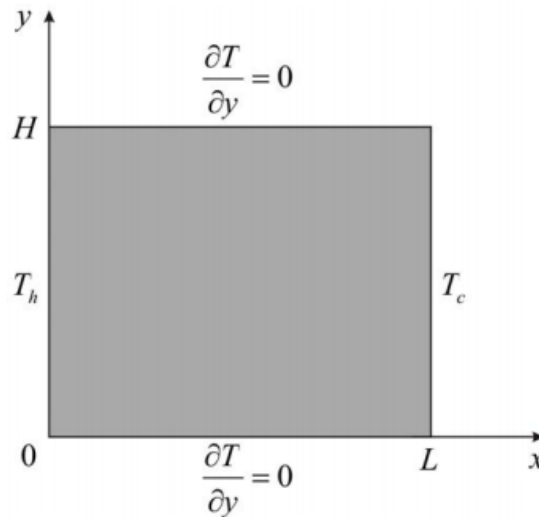


Рис. 2. Область вирішення.

Математична постановка задачі буде мати вигляд:

$$\rho c \frac{\partial T}{\partial t} = \lambda \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right), \quad \left| \begin{array}{l} 0 < x < L \\ 0 < y < H \end{array} \right. \quad (6)$$

Початкові і граничні умови запишуться в такий спосіб:

$$\begin{aligned} t = 0: T &= T_0, 0 \leq x \leq L, 0 \leq y \leq H \\ x = 0: T &= T_h, t > 0 \\ x = L: T &= T_c, t > 0 \\ y = 0: \frac{\partial T}{\partial y} &= 0, t > 0 \\ y = H: \frac{\partial T}{\partial y} &= 0, t > 0 \end{aligned} \quad (7)$$

Для апроксимації диференціального рівняння (6) різницеvim введемо просторово-часову сітку з координатами

$$x_i = (i - 1) \cdot h_x, \quad y_j = (j - 1) \cdot h_y, \quad t_n = n \cdot \tau,$$

де h_x, h_y - кроки сітки за координатами x, y відповідно; τ - крок за часом; $i = \overline{1, N_x}; j = \overline{1, N_y}; n = \overline{0, K}$. Тобто вся розрахункова область (рис. 2) покривається сіткою (рис. 3.).

Введемо наступне позначення: $T(x_i, y_j, t_n) = T_{i,j}^n$.

Дискретизацію рівняння (6) будемо проводити на основі локально одновимірної схеми А.А. Самарського, яка є абсолютно стійкою і має властивість сумарною апроксимації. Сутність цього підходу полягає в тому, що крок за часом реалізується в два етапи - на проміжному часовому кроці проводимо дискретизацію двовимірного рівняння (6) тільки в напрямку осі x і отримуємо одновимірний рівняння, після його рішення проводимо знову дискретизацію рівняння (6), але вже в напрямку осі y і, вирішуючи отримане

одномірне рівняння, визначаємо поле температури на цілому кроці по часу.

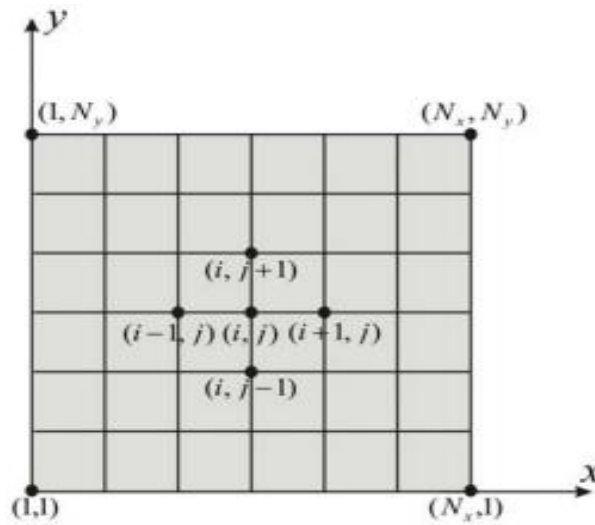


Рис. 3. Різницева сітка області рішення

Отже:

$$\rho \cdot c \cdot \frac{T_{i,j}^{n+\frac{1}{2}} - T_{i,j}^n}{\tau} = \lambda \cdot \left(\frac{T_{i+1,j}^{n+\frac{1}{2}} - 2 \cdot T_{i,j}^{n+\frac{1}{2}} + T_{i-1,j}^{n+\frac{1}{2}}}{h_x^2} \right), \quad (8)$$

$$\rho \cdot c \cdot \frac{T_{i,j}^{n+1} - T_{i,j}^{n+\frac{1}{2}}}{\tau} = \lambda \cdot \left(\frac{T_{i,j+1}^{n+1} - 2 \cdot T_{i,j}^{n+1} + T_{i,j-1}^{n+1}}{h_y^2} \right). \quad (9)$$

Різницеві рівняння (8), (9) зводяться до стандартного трехдіагонального узві і вирішуються послідовно методом прогонки. Спочатку для всієї області вирішується рівняння (8), після того як його рішення буде знайдено, переходять до вирішення рівняння (9).

Розглянемо рішення рівняння (8) методом прогонки. Наведемо це рівняння до виду $A_i T_{i+1,j}^{n+\frac{1}{2}} - B_i T_{i,j}^{n+\frac{1}{2}} + C_i T_{i-1,j}^{n+\frac{1}{2}}$. Тоді коефіцієнти A_i, B_i, C_i візьмуть вигляд:

$$A_i = C_i = \frac{\lambda}{h_x^2}, B_i = \frac{2 \cdot \lambda}{h_x^2} + \frac{\rho \cdot c}{\tau}, F_i = -\frac{\rho \cdot c \cdot T_{i,j}^n}{\tau}.$$

Для визначення прогоночних коефіцієнтів по співвідношенню

$$\alpha_i = \frac{A_i}{B_i - C_i \cdot \alpha_{i-1}}, \beta_i = \frac{C_i \cdot \beta_{i-1} - F_i}{B_i - C_i \cdot \alpha_{i-1}}.$$

необхідно знайти α_1 і β_1 з лівої граничної умови. Далі визначаючи значення $T_{N,j}^{n+\frac{1}{2}}$ з правої граничної умови, знаходять поле температури $T_{N,j}^{n+\frac{1}{2}}$ на

проміжному часовому шарі за формулою $T_i^{n+1} = \alpha_i \cdot T_{i+1}^{n+1} + \beta_i$. Після цього приступають до вирішення рівняння (9). Етапи рішення рівняння (9) аналогічні рішенню рівняння (8).

Висновки. Метод кінцевих різниць заснований на заміні похідних їх наближеним значенням, вираженим через різниці значень функції в окремих дискретних точках – вузлах сітки. В результаті таких перетворень диференціальне рівняння замінюється еквівалентним співвідношенням в кінцевих різницях, рішення якого зводиться до виконання нескладних алгебраїчних операцій. Таким чином, кінцевий результат рішення дається виразом, за яким значення «майбутнього» потенціалу (температури) в даній точці (вузлі) є функцією часу, її «справжнього» потенціалу та «справжнього» потенціалу суміжних вузлових точок.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Лыков А.В. Теория теплопроводности.– М.: Высшая школа, 1967.– 600 с.
2. Кузнецов Г.В., Шеремет М.А. Разностные методы решения задач теплопроводности: учебное пособие. / Г.В. Кузнецов, М.А. Шеремет. – Томск: Изд-во ТПУ, 2007. – 172 с.
3. Методы и алгоритмы моделирования тепловых полей в трехмерной сборке интегральных схем [Текст] / В.Ф. Барабанов, С.Л. Подвальный, А.В. Ачкасов, М.Н. Аралов // Радиотехника. – 2014. - №6 – С. 82-87.
4. Вержбицкий В.М. Основы численных методов. - М.: Высшая школа, 2002. – 840 с.
5. Райт мл. Р.С., Липчак Б. OpenGL. Суперкнига = OpenGL SuperBible. - 3 изд. - М.: Вильямс, 2006. - С. 1040. - ISBN 5-8459-0998-8.

УДК 004.415.3:681.6

Б.І. Мороз¹, І.Г. Гуліна¹, В.Ю. Сердюк¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

МЕТОДИ ТА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДЛЯ ВИВЧЕННЯ ПРАВИЛ ДОРОЖНЬОГО РУХУ ЗА ДОПОМОГОЮ ВПРОВАДЖЕННЯ ЕЛЕМЕНТІВ ІГРОФІКАЦІЇ

Анотація. Описано процес покращення користувацького досвіду при роботі з системами вивчення правил дорожнього руху. Покращення досягається за рахунок будування найбільш ефективної архітектури системи із впровадженням елементів ігрофікації.

Ключові слова: адаптивна система, архітектура систем, ігрофікація, Unity-додаток, C#, MonoBehaviour, вивчення правил дорожнього руху.

Вступ. В останній час все більше і більше світових компаній починають використовувати елементи ігрофікації в різних сферах. Адаптивність систем навчання стає не лише опціональною особливістю навчальної системи, а

обов'язковою в будь-якій системі спрямованій на навчання. Ігри існують з часів виникнення нашої цивілізації. Навіть відеоігри вже існують понад 40 років і представляють індустрію, яка приносить більше 70 мільярдів доларів прибутків на рік. Мільйони людей у всіх куточках Землі проводять по 2-3 години в день за іграми на консолі, персональних ком'ютерах та мобільних телефонах. Багато адаптивних навчальних середовищ показали ефективність при використанні, але вони були швидко покинуті користувачами із-за відсутності чіткої мотивації.

Незважаючи на високий інтерес до цієї індустрії в останній час ігрофікація є доволі молодою сферою досліджень. Термін «ігрофікація» був введений лише в 2002 році Ніком Пелінгом. Розробкою теоретичних підходів і практичних методів для вирішення даного кола проблем займалися такі вчені як Томас Малон, Річард Бартл, Стівен Драпер, Нік Пелінг, Кевін Вермахт та багато інших.

Постановка задачі: Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- Викласти принципи побудови адаптивних систем навчання;
- Дослідити особливості адаптивних систем навчання;
- Дослідити елементи ігрофікації;
- Побудувати інформаційну систему вивчення правил дорожнього руху;
- Впровадити елементи ігрофікації;

Основний зміст роботи. Під час роботи над даною роботою були використані наступні методи та інструменти.

- основи побудови адаптивних інформаційних систем;
- елементи ігрофікації;
- C#, .Net, Monobehaviour, Unity.

Адаптивні освітні системи відстежують важливі характеристики учня та вносять відповідні корективи в навчальне середовище для підтримки та вдосконалення навчання. Метою адаптивних освітніх систем, в контексті цього розділу, є створення навчально-обґрунтованого та життєздатного середовища, яке підтримує навчання для студентів з різними здібностями, вадами розвитку, інтересами, походженням та іншими характеристиками. Завдання досягнення цієї мети значною мірою залежить від точного визначення характеристик конкретного учня або групи учнів – таких як тип і рівень знань, навичок, рис особистості, афективних станів – а потім визначення способу старіння інформації для вдосконалення студента навчання.

Ігровий рушій – це архітектура, яку розробники використовують для запуску гри. Загалом ігровий рушій повинен надавати для розробника наступні елементи:

- Фізика;
- Способи введення та виведення інформації;
- Рендеринг елементів;
- Створення сценаріїв поведінки елементів;
- Виявлення зіткнень елементів;

- Штучний інтелект.

Ігрові рушії - це багаторазові компоненти, які розробники використовують для побудови фреймворку. Це дає їм більше часу, щоб зосередитись на таких унікальних елементах, як моделі персонажів, текстури, взаємодія об'єктів тощо. Однак, як і раніше, існує велика кількість великих компаній і навіть незалежних команд, які створюють свої двигуни. Для цього потрібно використовувати інтерфейси програм, такі як DirectX, OpenGL та XNA, а також комерційні та відкриті бібліотеки, що містять фізику, графіки сцен та бібліотеки графічного інтерфейсу.

Останні роки показали швидке розповсюдження масового програмного забезпечення, яке надихається відеоіграми. Зазвичай термін «ігрофікація» пов'язують із значним набором існуючих концепцій та досліджень взаємодії людини з комп'ютером та ігрових досліджень, таких як серйозні ігри, загальнодоступні ігри, ігри з альтернативною реальністю або грайливий дизайн. Однак не зрозуміло, як «ігрофікація» пов'язана з ними та чи позначає воно нове суттєве явище.

Ігрофікація – це застосування елементів ігрового дизайну та ігрових принципів у неігрових контекстах. Його також можна визначити як сукупність видів діяльності та процесів для вирішення проблем, використовуючи або застосовуючи характеристики ігрових елементів. У ігрофікації зазвичай використовуються елементи ігрового для поліпшення залучення користувачів, організаційна продуктивність, потік, навчання, краудсорсинг, збереження знань, набір та оцінка працівників, простота використання, корисність систем, фізичні вправи, порушення правил дорожнього руху, апатія виборців та інше. Збірник досліджень з питань ігрофікації показує, що більшість досліджень з питань ігрофікації виявляють позитивний вплив на людей. Однак існують індивідуальні та контекстуальні відмінності.

Методи ігрофікації мають на меті використати природні бажання людей до спілкування, навчання, оволодіння, змагання, досягнень, статусу, самовираження, альтруїзму чи закритості або просто їхньої реакції на формування ситуації як гри чи гри. Ранні стратегії ігрофікації використовують винагороду для гравців, які виконують бажані завдання або змагання, щоб залучити гравців. Типи винагород включають бали, значки або рівні досягнення, заповнення індикатора прогресу, або надання користувачеві віртуальної валюти. Визначення винагороди за виконання завдань видимим для інших гравців або надання дощок лідерів – це способи заохочення гравців до змагань.

Інший підхід до ігрофікації – зробити так, щоб існуючі завдання були більш схожими на ігри. Деякі прийоми, що використовуються у цьому підході, включають додавання значущого вибору, використання навчальних посібників, збільшення викликів та додавання розповіді.

Елементами ігрофікації є:

- Бали
- Досягнення;
- Значки;

- Графік ефективності;
- Сюжет;
- Зворотній відлік;
- Аватари;
- Соціалізація та створення персонажів всередині гри.

Деякі з потенційних переваг успішних ініціатив ігрофікації в процесі навчання включають:

- надання студентам права власності на їх навчання;
- можливості для ідентичності шляхом прийняття альтернативних «я»;
- свобода невдач і спробувати ще раз без негативних наслідків;
- шанси збільшити задоволення від навчання;
- можливості диференційованого навчання;
- зробити навчання видимим;
- забезпечення керованого набору підзадач та завдань надихаючи студентів відкрити внутрішні мотиватори для навчання;

- мотивація студентів з дислексією з низьким рівнем мотивації.

Крім усіх цих тенденцій, ігрофікація не є безпрограшною комбінацією і може призвести до деяких побічних ефектів. Поширена критика породила новий вираз, названий "pointification" (або "badgefication"), що описує акт створення винагород як бали чи досягнення (значки) без стратегії чи справжнього значення для гравців.

Програма складається з п'яти компонентів:

1. Модуль візуалізації задач правил дорожнього руху;
2. Адаптивний модуль;
3. Модуль, контролюючий хід програми та взаємодію всіх модулів між собою;
4. Модуль ігрофікаційних елементів;
5. Модуль який вимірює зацікавленість користувача.

Під час виконання кваліфікаційної роботи провели експеримент на не адаптивній системі без елементів ігрофікації. Групі користувачів А, ми встановили цю програму без модулю адаптивності та ігрофікації. Групі користувачів В ми встановили адаптивну систему. Групі користувачів С ми встановили повну версію з елементами ігрофікації. Після тижня з модулю, який вимірює зацікавленість користувачів ми зібрали данні та представили наступним чином (Рис 1.):

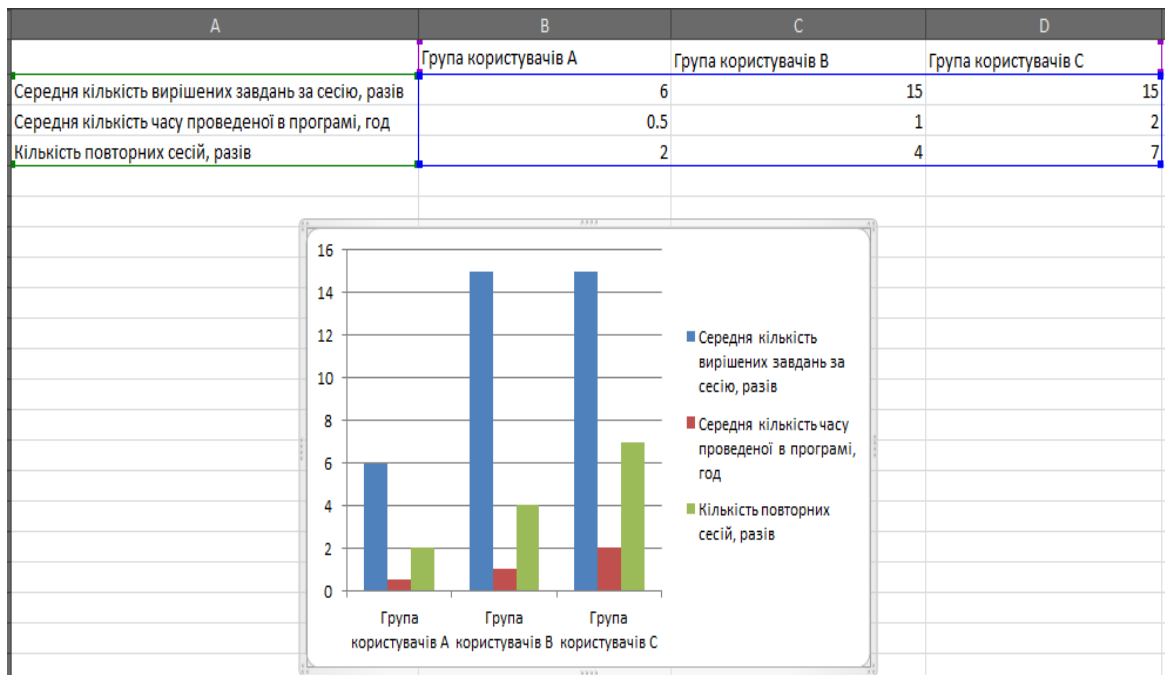


Рис. 1. Таблиця даних модулю зацікавленості користувача

З цих даних можна зробити наступний висновок:

1. В адаптивних системах кількість вирішених завдань за сесію значно більше ніж в не адаптивних системах;
2. Модуль ігрофікації значно поліпшив кількість повторних сесій, що доволі важливо для нашої системи;
3. Кількість часу показує, що чим більше система адаптується тим більше часу користувач готовий провести в програмі;
4. Елементи ігрофікації вплинули лише на кількість повторних сесій та кількість часу, але на кількості вирішення завдань за сесію не вплинули.

Наукова новизна результатів роботи полягає в удосконаленні методів та алгоритмів вивчення правил дорожнього руху за допомогою впровадження ігрофікації.

Висновки. В результаті виконання роботи було створено надійну та ефективну інформаційну технологію для вивчення правил дорожнього руху. У тому числі, було обрано найбільш важливі та доречні елементи ігрофікації та розглянуто теоретичні та практичні основи побудови адаптивних систем.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Heinze N., Nejd W., A Logical Characterization of Adaptive Educational Hypermedia. New Review of Hypermedia and Multimedia (NRHM) / Impact Factor – New York, 2004 – 114 с.
2. Brusilovsky, P. Methods and techniques of adaptive hypermedia. User Modeling and User-Adapted Interaction. – 1996. – № 6 (2-3). – 87-129 с.
3. Zichermann, G., Cunningham, C. Gamification by Design: Implementing game mechanics in web and mobile apps/ O'Reilly Media – Dallas, 2011 – 182 с.

4. Burke B. Gamify: How Gamification Motivates People to Do Extraordinary Things/ Routledge – New York, 2014 – 192 с.

5. Hocking J. Unity in action: Multiplatform game development in C#/ Manning – New York, 2015 – 334 с.

УДК 004.415.3:681.6

М.О. Алексєєв¹, О.С. Шевцова¹, Д.Є. Моссур¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ОПТИМІЗАЦІЇ, СИНХРОНІЗАЦІЇ, КЕРУВАННЯ ТА ВІДТВОРЕННЯ СТАНІВ СПИСКІВ З ПОСТОРІНКОВИМ ВІДТВОРЕННЯМ

Анотація. Описано процес створення максимально ефективного інструменту для вирішення типових задач з відтворення екранів які містять списки з посторінковими запитами. Покращення досягається за рахунок розробки відповідної програмної бібліотеки, яка містить скінченний автомат станів списку та оптимізує логіку з відображення станів та елементів списків.

Ключові слова: *список з посторінковим відтворенням, типи посторінкових запитів, машина станів, алгоритм Маєрса, мобільні додатки.*

Вступ. Розповсюдженість смартфонів і широкий спектр їх використання притягує увагу бізнесу, як ще один канал для збільшення аудиторії та покращення своїх сервісів. Бізнес створює попит для створення нових мобільних додатків. Вже існує багато компанії, які займаються створенням таких додатків.

Майже не можливо зараз знайти додаток, який не містив би хоча б один список. Бізнес, як і ринок, дуже змінливий, тому майже всі данні зберігаються не на пристроях користувачів, а на серверах. Списки – не виняток. Такі умови впливають на стани відображення, оскільки даних під час запуску додатку на пристроях немає, то поки він буде загрузатися користувачу потрібно відобразити відповідний стан. Якщо це буде просто пустий екран користувач подумає що додаток не відповідає і може без зволікань установити додаток конкурентів і бізнес втратить клієнта. Таких станів декілька і вони розповсюджуються майже на кожен список.

Таким чином, маємо частину бізнес логіки, яка повторюється і начебто можна написати абстрактну реалізацію таких станів. Але існуючі бібліотеки створюють цілий ряд обмежень для їх використання на ряду з неможливістю змінити їх код тільки сповільнює процес розробки додатка.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- провести узагальнений аналіз бізнес вимог до типових задач з відображення екранів зі списками;

- побудувати скінченний автомат для станів екрану зі списком з посторінковими запитами;
- дослідити та застосувати алгоритм Маєрса для оптимізації відображення елементів посторінкового списку;
- розробити крос-платформну бібліотеку;
- дослідити альтернативні вирішення;
- провести детальний аналіз та порівняння найкращого альтернативного вирішення з розробленою бібліотекою;
- зробити висновки щодо доцільності створення подібного інструменту;

Основний зміст роботи. Спираючись на проведений у роботі детальний аналіз вимог до екранів ПЗ, на яких використовуються списки з посторінковими запитами, можна побудувати бізнес модель станів у вигляді графу. Таку модель зображено на рис. 1.

Використовуючи в якості бази для подальшого аналізу граф станів приведений на рис. 1, можна стверджувати, що кількість станів для екрану відображення списку з посторінковими запитами скінченна. Кількість подій які можуть змінювати ці стани також скінченна. Тому можливим рішенням задачі буде скінченний автомат.

Скінченний автомат, або як ще його називають у англійській літературі, машина станів – це математична розрахункова модель, абстрактний автомат, який може знаходитись в одному зі скінченної кількості притаманних йому станів. Перехід з одного стану в інший називається транзакцією. Новий стан скінченного автомату буде обраний спираючись на вхідні дані, які викликали транзакцію та попередній його стан. [1]

Скінченні автомати поділяють на детерміновані та не детерміновані. В детермінованих скінченних автоматах кожен вхідний параметр відповідає одній транзакції. Для не детермінованих автоматів це правило не працює, і один визначений вхідний параметр може призвести до однієї, декількох транзакцій або до жодної [2].

Для вирішення поставленої задачі побудуємо скінченний не детермінований автомат для відображення станів списку з посторінковим завантаженням PFA (Pageable Final Automaton).

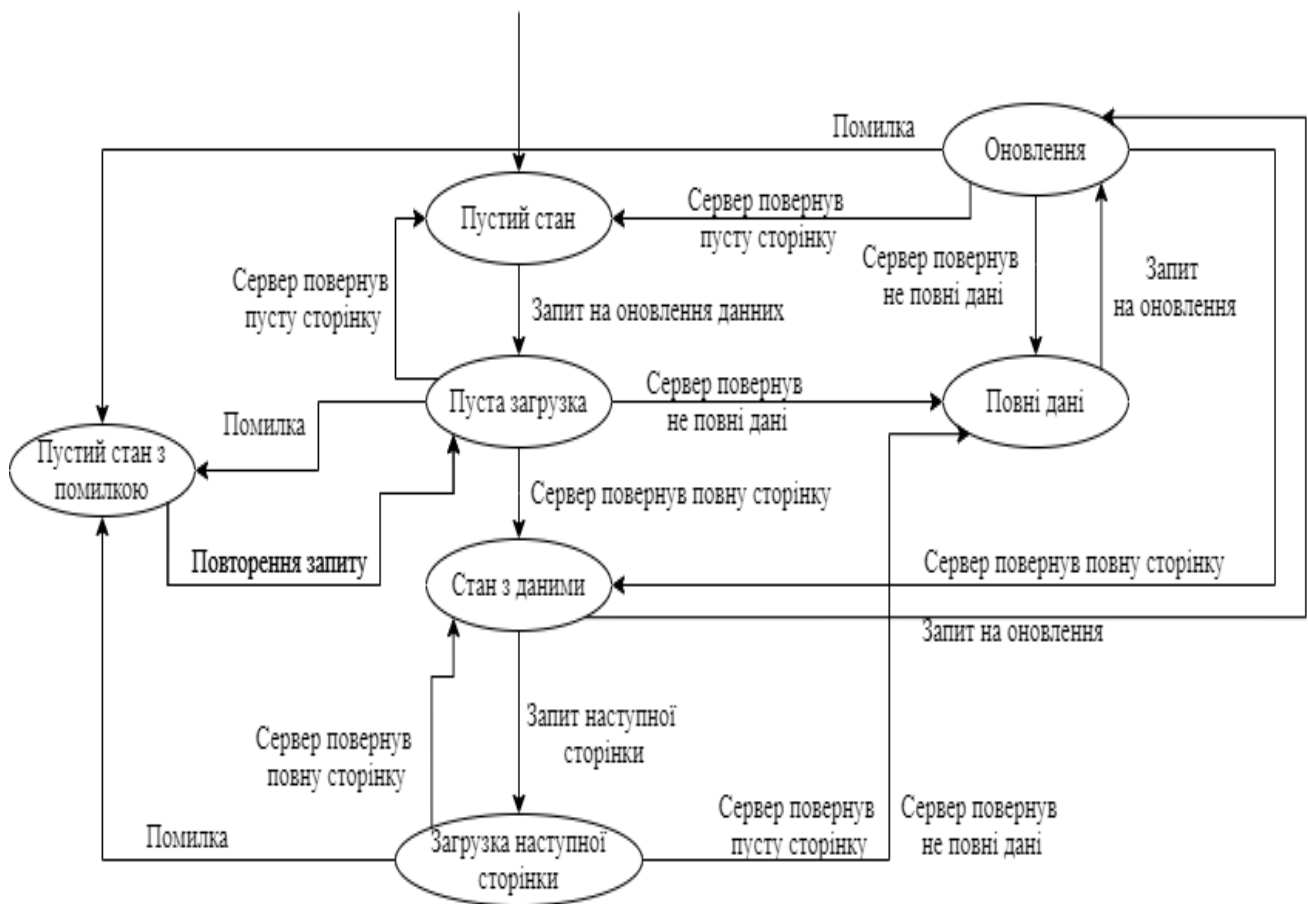


Рис. 1. Граф бізнес моделі станів екрану з посторінковим списком

Спираючись на аналіз області застосування можна побудувати множину станів автомату та записати її у вигляді формули 1.

$$Q = \{ \text{Empty, EmptyProgress, EmptyError, Data, Refresh, NewPageProgress, FullData} \} \quad ((1))$$

В якості вхідних параметрів скінченний автомат посторінкового відображення може отримувати події від користувача, через користувацький інтерфейс та події які сповіщають про результат запиту. Алфавіт автомату з умовними позначеннями подій відображений у формулі 2.

$$\Sigma = \{ \text{Restart, RefreshEvent, LoadMore, NewPage, PageError, Edit} \} \quad (2)$$

Функція переходів між станами (табл. 1) є повною, тож усі переходи є визначеними. Лівий стовбець таблиці описує дійсний стан автомату на момент додання нової події. Тип події вказано у першому рядку таблиці. Пусті клітинки вказують на те, що подія не буде оброблена та стан автомату залишиться незмінним.

**Таблиця функцій переходів скінченного автомата для екранів з
посторінковими списками**

	Restart	Refresh Event	Load More	New Page	Page Error	Edit
Empty	Empty Progress	Empty Progress				Data / Empty
Empty Progress				Data/ Empty	Empty Progress	NewPageProgress / EmptyProgress
Empty Error	Empty Progress	Empty Progress				
Data	Empty Progress		New Page Progress			Data / Empty
Refresh	Empty Progress	Refresh		Data/ Empty	Empty Progress	Refresh / EmptyProgress
New Page Progress	Empty Progress			Data/ Empty	Empty Progress	NewPageProgress / EmptyProgress
FullData	Empty Progress	Refresh				FullData / Empty

Початковим станом необхідно рахувати стан Empty, оскільки жодних даних не було ще отримано і пам'ять автомата пуста, хоча за кращими практиками користувацького інтерфейсу необхідно завантажувати дані одразу. Перше завантаження буде викликано додатком спираючись на його стан.

Оскільки користувач може перервати роботу не тільки автомата, а й всього додатку у будь-який момент, то будь-який стан з множини станів автомату можна рахувати кінцевим. Таким чином мова скінченного автомата приймає всі можливі слова, складені з алфавіту Σ (3).

$$L(PFA) = \{\omega * \Sigma^*\} \quad (3)$$

Використовуючи для подальшої роботи розроблені та побудовані модель станів та мову скінченного автомата, можна побудувати ERR-діаграму класів (рис. 2) для основної частини бібліотеки, метою якої є встановити базову логіку зміни станів та синхронізації відтворюваних списків.

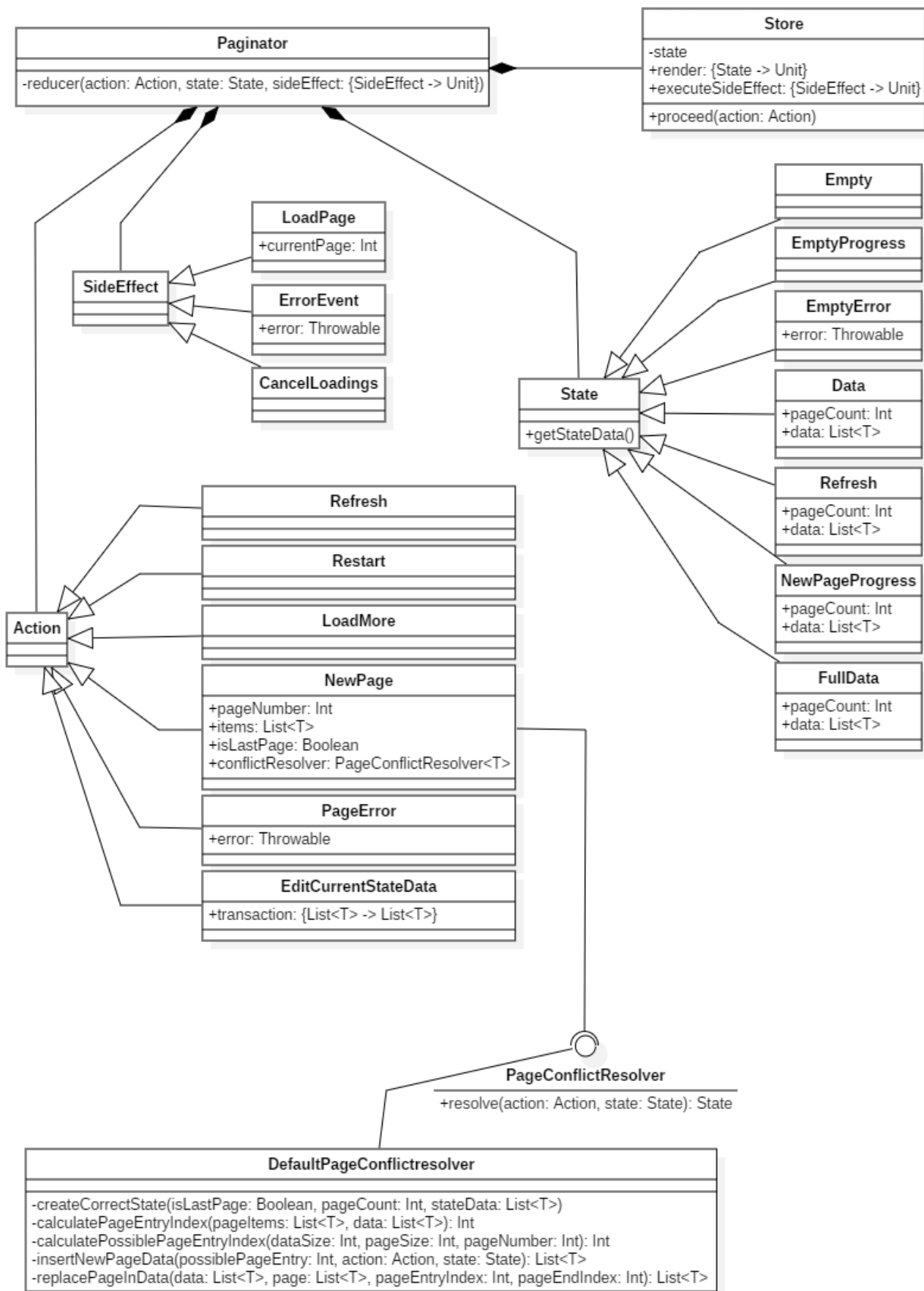


Рис. 2. EER-діаграма класів моделюючих скінченний автомат PFA

Основним класом буде Paginator основною задачею якого є агрегація всіх сутностей які мають відношення до скінченного автомату. Таким чином вкладені класи Action та State відповідають мові автомату та множині його станів відповідно, а метод reducer зберігає функцію переходів станів скінченного автомату [4].

Відповідно до діаграми класів, список спадкоємців класів State, Action та SideEffect обмежений. Для таких випадків у Kotlin існує модифікатор класу sealed, який дозволяє обмежити список спадкоємців у кодї для спрощення виразів перевірки на тип об'єкту та інкапсуляції можливості наслідування.

Описаний вище модифікатор вказується перед ключовим словом class.[3] Спадкоємців класів об'явлених як sealed class дозволено створювати виключно у тому ж документі, де знаходиться їх предок. Задача визначення класу досить поширена в контексті даної архітектури Для визначення класу об'єкту використовується оператор when., на об'єктах sealed класів не потребується створення додаткової вітки else, яка описує випадок для не зазначений у попередніх вітках. Оскільки список спадкоємців лімітований, то кожна вітка оператора when буде відповідати певному класу.

Виходячи із загального положення і користуючись в якості аргументу та логічного обґрунтування сукупністю раніше згаданих емпіричних знань саме функція переходів станів визначає коли необхідно завантажити наступну сторінку, обробити помилку чи відмінити дійсне завантаження даних, до класу Paginator необхідно додати утилітний клас SideEffect за допомогою якого можна сповістити оточення про необхідність обробки вище зазначених сформованих випадків.

Пам'ять скінченного автомату представлена об'єктом класу Store, який окрім стану зберігає функції для зворотного сповіщення оточення про зміну стану та необхідність обробки сайд ефекту.

Задля запобігання проблем касту об'єктів до певних типів, а також для більшої гнучкості до усіх методів та класів, які працюють зі списком даних було додано тип T, який обмежує можливі варіанти доданих даних. Такі типи також називають дженеріками або шаблонами.

Оптимізацію оновлення списку виконано за допомогою порівняння безпосередньо даних, які необхідно відобразити між собою. Логіку такого порівняння елементів списку інкапсулює в собі клас – DiffUtil. Зазначений клас використовує алгоритм Маєрса для пошуку елементів які змінилися та потребують оновлення відображених даних, далі по створеним алгоритмом графам DiffUtil визначає елементи які потребують видалення, додавання до відображених або зміни позиції.

Наукова новизна: отримав подальший метод синхронізації, керування та відтворення серій посторінкових запитів при паралельному виконанні; удосконалено процес побудови та відтворення списків з посторінковими запитами протягом створення ПЗ.

Висновки. Основною метою було створення максимально ефективного інструменту для вирішення типових задач з відтворення екранів які містять списки з посторінковими запитами. Як результат, в процесі роботи було

розроблено бібліотеку, яка містить скінченний автомат станів списку та оптимізує логіку з відображення станів та елементів списків. Функціональна частина бібліотеки розроблена на мові Kotlin, а скрипт модуля на Groovy для системи автоматичного збирання проектів Gradle, що дозволяє використовувати дане вирішення для крос-платформних проектів з використанням технології Kotlin Multiplatform Mobile. Для короткої документації проекту було застосовано мову Markdown.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Sakarovitch, Jacques (2009). Elements of automata theory. Translated from the French by Reuben Thomas. Cambridge University Press
2. Brutscheck, M., Berger, S., Franke, M., Schwarzbacher, A., Becker, S.: Structural Division Procedure for Efficient IC Analysis. IET Irish Signals and Systems Conference, (ISSC 2008), pp.18–23. Galway, Ireland, 18–19 June 2008
3. Dmitry Jemerov, Svetlana Isakova Kotlin in Action Version 11 New York City Manning Publications 2016 – 360 pages
4. Design Patterns Elements of Object-Oriented Software Erich Gamma and others Addison-Wesley 1994 – 416 pages
5. Eugene W. Mayers. An O(ND) Algorithm and It's Variations URL: <http://www.xmailserver.org/diff2.pdf>

УДК 004.415.3:681.6

Б.І. Мороз¹, А.А. Мартиненко¹, К.В.Додатко¹

¹Національний технічний університет "Дніпровська політехніка", Дніпро, Україна

МЕТОДИ, АЛГОРИТМИ ТА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДЛЯ ДОСЛІДЖЕННЯ ТА АНАЛІЗУ РИЗИКІВ ОПЕРАЦІЙ НА ФОНДОВОМУ РИНКУ

Анотація. Описано процес аналізу ризиків на фондовому ринку. Проаналізовані різні методи VaR-підходу та запропонована модифікація одного із методів. Створено програмне забезпечення для візуалізації VaR.

Ключові слова: біржова торгівля, ризик менеджмент, VaR, R Studio, MS Excel, Python, модифікація, волатильність, мінімізація, коваріація.

Вступ. Кожного дня на світових фондових біржах здійснюються мільярди операцій. Ціни фінансових інструментів формуються за допомогою сотень факторів, які змінюються у реальному часі. При будь-яких умовах існує можливість того, що на результат операції буде впливати один чи декілька несприятливих факторів. Для здобуття успіхів у виконанні таких операцій при наявності цих невизначених факторів потребує проведення ретельного аналізу та оцінки ризиків. Основними кількісними характеристиками ризику є – ступінь (ймовірність), рівень (величина втрат) та ресурс допустимого ризику (час функціонування системи). Після оцінки приведених вище характеристик

ризик, обирається підхід до роботи з ризиком, що, як правило, включає уникнення, передачу, пом'якшення та ін.

При оцінюванні основних кількісних характеристик ризику, найчастіше використовують VaR-підхід. Результатом застосування такого класичного підходу є впевненість, що впродовж деякого періоду часу втрати не перевищать деякого значення. Обчислене значення втрат формує резервний капітал, який буде доступним для покриття можливих збитків від подальших операцій на фондовому ринку. Збитки може отримати як сам виконавець операції, так і клієнт інвестиційного фонду, який довірив свої кошти під управління компанії. В іншому випадку своєчасне покривання збитків клієнта є питанням збереження репутації компанії.

При використанні VaR-підходу обов'язковим є оцінювання волатильності кожного фінансового інструмента, що входить в інвестиційний портфель. Окрім основного параметра, що оцінюється, – середньоквадратичного відхилення (волатильності), нормальний розподіл має несуттєвий параметр (математичне сподівання). Саме VaR-підхід є найпоширенішим для мінімізації ризиків, має різноманітні модифікації та використовується багатьма інвестиційними фондами.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- Дослідити історію фондового ринку, його інфраструктуру в наш час, класичні методи аналізу фондових ринків, та основні операції, що здійснюються на фондовому ринку.

- Класифікувати ризики операцій на фондовому ринку.

- Систематизувати інформацію про основні переваги та обмеження ризик-менеджменту, підходи до роботи з ризиками та кількісні характеристики ризиків.

- Дослідити застосування параметричного VaR-підходу для оцінювання фінансових ризиків.

- Дослідити метод історичного моделювання, метод параметричної оцінки, заснований на коваріаційній моделі, та метод оцінки Монте-Карло.

- Розробити стратегію оцінювання основних характеристик ризиків операцій на фондовому ринку з використанням програмного забезпечення R Studio та MS Excel.

- Провести порівняльний аналіз отриманих за різними методами оцінок.

- Створити програмне забезпечення для візуалізації результатів за допомогою мови програмування Python.

Основний зміст роботи. Під час роботи були використані наступні методи та інструменти: метод історичного моделювання; варіаційно-коваріаційна модель параметричної оцінки; R-Studio, MS Excel, Python.

Ризик – це поєднання ймовірності та наслідків настання несприятливих подій. Аналіз та управління ризиками мають на меті виявлення областей невизначеності, які можуть мати негативний вплив у майбутньому; проаналізувати ці невизначеності; скласти методи роботи з ризиками.

Ризик-менеджмент – це знаходження, аналіз і оцінка ризиків. На сьогоднішній день, у кожній великій компанії бізнес-аналітики розробляють плани уникнення, зменшення або модифікування ризиків і за необхідності використовують ці плани на практиці. Ризик-менеджмент відбувається у реальному часі, поки ще йде робота на проекті. Регулярні дослідження інтересів в потрібній області ринку надає змогу як визначити нові ризики, так і слідкувати за вже визначеними.

Аналіз ризику складається з розуміння ризику та оцінки його кількісних характеристик. Як правило, існують установлені стандарти роботи з ризиками, які враховуються під час аналізу. При аналізі можуть брати за основу стандартну, або базову шкалу впливу ризиків. Для того, щоб визначити потенційну вигоду і прийнятний рівень ризику таких категорій, як грошові витрати та витрачені зусилля, можна регулювати гранично допустимі значення приведених параметрів.

Якщо на фондовому ринку здійснюються будь-які операції, першим кроком повинна бути оцінка ризиків (очікуваних, та неочікуваних). Пізніше розробляється план, як саме затвердити резервний капітал. Також важливо визначити, протягом якого періоду часу і з якою ймовірністю величина втрат не буде виходити за межі резервного капіталу.

Після виконання основних кроків, можна з урахуванням ризиків оцінити реальну доходність операцій, а також встановити ліміт втрат, після якого необхідно виходити з позиції. Для визначення цих показників найчастіше використовують VaR-підхід (Value at Risk). VaR (англ. Value at Risk) – величина резервного капіталу, яка з деякою ймовірністю перекирив можливі втрати протягом деякого періоду.

В даний час основними, класичними підходами до оцінки VAR вважаються: метод історичного моделювання; метод параметричної оцінки, найбільш поширений в формі варіаційно-коваріаційної моделі; метод імітаційного моделювання, часто іменованій по основній застосовуваній в його рамках моделі методом Монте-Карло.

На більш ранніх етапах використання VAR найбільш поширені були історичне моделювання та варіаційно-коваріаційні моделі, з яких перевагу віддавали останнім в зв'язку з більш наукоподібним математичним апаратом, відповідно до домінуючим думкою про необхідність застосування складних математико-статистичних технологій. Обмеженням застосування імітаційного моделювання довгий час були комп'ютерні технології, розвиток яких дозволило створити прийнятні за швидкістю і швидкодії рішення лише на рубежі століть.

На сучасному етапі можна констатувати вихід процесу технічних інновацій в області аналізу ризиків на стадію мінімальної граничної корисності наукомістких досліджень, що на рівні громадської думки знайшло відображення в завершенні буму просунутого статистичного моделювання. При виборі учасниками ринку практичних програмних рішень зростає роль факторів простоти, наочності і ефективного поєднання витрат і результатів.

В цілому, в контексті перспектив розвитку методології управління ризиками VAR оцінюється як «базова технологія», і на сучасному етапі

зберігають актуальність все 3 зазначених методи оцінки VAR у різних модифікаціях. VaR-підхід дозволяє з деяким довірчим рівнем визначити верхню межу втрат, що виникають в результаті настання ризикових подій (1):

$$P\{Loss_t(k) < VaR_t(k)\} = \frac{100-\alpha}{100} \quad (1)$$

де $Loss_t$ – фактичні втрати на момент часу t за період k днів; $VaR_t(k)$ – прогнозовані втрати на момент часу t за період k днів; α – довірчий рівень.

Існує два основних підходи для обчислення VaR: параметричний (VaR оцінюється за параметрами, які характеризують ціновий ряд) та непараметричний (обчислюється за історичними значеннями доходності портфеля або за методом Монте-Карло). Формула для обчислення параметричної величини VaR (2):

$$VaR_t(k) = \alpha\sigma * VP * \sqrt{N} \quad (2)$$

де α – квантиль довірчого інтервалу; σ – волатильність (норма мінливості); VP – відкрита позиція; N – період прогнозу.

Для нормального розподілу були розроблені таблиці, за якими можна знайти відповідні квантилі. Волатильність найчастіше характеризується, як середньоквадратичне відхилення. VaR-підхід найчастіше використовується для аналізу розмірів резервного капіталу, оцінювання доходності операцій з урахуванням ризиків, моделювання сценаріїв розвитку, виявлення ситуацій, при яких необхідно закривати позицію.

Варіаційно-коваріаційний підхід до розрахунку VaR:

мінуси:

- Використовується припущення про нормальний закон розподілу доходностей.
- При розрахунку VaR більше ніж на один день передбачається що матриця коваріацій постійна.
- Неможливо використовувати для розрахунку VaR для опціонів і інструментів з рисами опціонів.

плюси:

- Швидкий розрахунок, не вимагає великих обчислювальних ресурсів.
- Якщо в портфелі тільки лінійні інструменти на зразок акцій, ф'ючерсів, валют, вигідно використовувати для розрахунку VaR.
- Коваріаційна матриця може бути отримана від вендора.

Історичний підхід до розрахунку VaR:

плюси:

- Використовується розподіл доходностей на ринку з «товстими хвостами» (fat-tailed distribution).

- Матриця коваріацій не розраховується, що знижує кількість помилок.

мінуси:

- Труднощі при розрахунку VaR для опціонів, так як опціонні треки короткі, також можуть виникнути проблеми зі зберіганням великої кількості опціонних треків.

- Інфраструктурні витрати припускають, що повинна бути база даних, де будуть зберігатися та оновлюватися історичні треки цін.

Наукова новизна: запропонована модифікація історичного методу VaR-підходу для оцінювання фінансових ризиків, створено програмне забезпечення для складання портфеля компаній та візуалізації методу розрахунку VaR.

Висновки. В результаті виконання дипломної роботи було проаналізовано та порівняно різні методи розрахунку VaR. Найбільш оптимальний метод було модифіковано та спроектовано програмне забезпечення для забезпечення візуалізації.

ПЕРЕЛІК ПОСИЛАНЬ

1. Максимов В. Основы успеха валютных спекуляций / В. Максимов. – М.: Издательский дом «Евро», 2003. – 312с.

2. Ф.Х. Найт Ризик, невизначеність і прибуток / Видавництво «Справа», 2003. – 341с.

3. Jorion P. VAR mapping // Value at Risk: The New Benchmark for Managing Financial Risk. / McGraw-Hill, 2006. - 602 p.

4. Dowd K. Measuring Market Risk / John Wiley & Sons Ltd, 2005. - 390 p.

5. Талейб Н. Н. Антихрупкость. Как извлечь выгоду из хаоса / пер. с англ. Н. Караева. / М.: Изд. КоЛибри, Азбука-Аттикус, 2014. - 768 с.

UDC 004.93

L. Kabak¹, D. Buslov¹

¹Dnipro University of Technology, Dnepr, Ukraine

USING COMPUTER TECHNOLOGY FOR BLIND PEOPLE

Annotation. The theses describe the problems that blind people face in the modern digital world. And proposed ways to solve them using digital technologies. The use of artificial intelligence and functioning accessories - glasses that will receive all the information from the outside world, which with the help of the voice of the AI will be transmitted to the person was proposed.

Keywords: *blind people, digital technologies, digital glasses, face_recognition, Google glasses, Apple's glasses.*

Problem statement: More and more people in the world are becoming blind. According to statistics, by 2020 there were about 75 million. Though not geometrically progressive, this number is increasing. Although many interesting and useful solutions have been invented to help blind people navigate in the street and at home, which are trying to evolve with the course of technology, but they are not as spectacular as we would like them to be.[1]

Analysis of recent publications and research: In 2019, an app from Microsoft was created that allowed blind people to "see" their surroundings, which was voiced by the program in words. The app is called "Seeing AI." It opens new possibilities for people, allowing them to live almost the same life as seeing people [2].

Blind people have incredible echolocation abilities. Blind people can use it to navigate in space. Fortunately, they don't have to make ultrasonic sounds like bats, just click their tongue. The experiment was described in the journal Proceedings of the Royal Society B. This innate or acquired ability is great for them to learn new technologies that become a complement to their echolocation.

And there is the possible use of advanced artificial intelligence technology, which Tesla is developing, and accessories from Google and Apple. Tesla has made a breakthrough in AI that could be applied to Google and Apple's glasses. There will be similarities to the "Seeing AI" app - information about what will surround a person will be communicated through voiceover. To drive a car, you need a neural network performance of more than 50 trillion operations per second. If we take a human, this number of operations would be optimal, even more. The size of the package determines the number of elements you work with at the same time. For example, Google's TPU has a packet size = 256. You have to wait until you have 256 elements to process, only then can you start working. The developers did it differently - they did it with batch size = 1. As soon as the image appears, the packet is processed immediately to minimize latency and increase security [3, 4].

Regarding their system: they adopted such a law as "power supply redundancy". The first chip runs on one power source and the second on another. The cameras have the same principle - one half runs on one power supply, the other half runs on the second. Both chips receive all the video and process it independently of each other. Here we also use invented imaging technology from Google Glasses, which we upgrade with solutions from Tesla.

As supervisor **Pete** said, "I've never worked on a project where the performance improvement was more than 3. So that was pretty fun. If you compare it to the nvidia Drive Xavier solution, the chip there provides 21 trillion operations, whereas our FSDC with two chips is 144 trillion operations." (FSDC- Full Self-Driving Computer)

Navigation. In this context, the most relevant is the use of Google Maps, which allows to get to the desired location by the most optimal way. This development by Google has useful functions, such as: traffic congestion and how to get to the destination. If the first function is clear - it tells the way, which road to take to get to the destination faster, the second talks about the transport. For example - the number of the minibus. You say the number of the bus, for example "40". The system guides you to the bus stop and with the help of object recognition (in the next chapter) - distance and number - lets you get on it and arrive at the designated stop.

Object recognition. Our main task in helping the blind using AI is pattern recognition. For seeing people, this is a very simple task. We look at an object and see a table, a chair, a streetlight, a traffic light. Very easy and simple for us, but not for the computer. The problem is that the image of these objects for the computer is

just an array of pixels, where each pixel is the brightness value at that point. That is, when we "look through" these images, the computer gets about a million numbers in the array.

We show our neural network an object, such as our traffic light, and the network will make a prediction of what it sees. Neural networks are initialized randomly, so it may be that the network thinks it's a pole. Then during the test, we find out that it's a traffic light, and we say we want the probability of a traffic light for that object to increase, and the probability of everything else to decrease. Then the error back propagation method is turned on. Random gradient descent, which allows us to propagate the signal through the links and update their weights. If we update the link weights just a little bit, the probability of a traffic light for that object will increase slightly, and the probability of the other choices will decrease. Naturally, this is not done with one image, but with millions. And these results are recorded in memory, which allows the network to learn. In the next tests, the results will be more accurate and faster.

Distance to object. As we know, neural networks are an insanely powerful visual recognition tool. If we want them to recognize distance, we need to markup distances, and then the network will learn to do that.

We can use forward-facing radar. This radar measures and marks the distance to objects that the neural network sees. This system could also be put into glasses, which would allow the blind to navigate in space without a cane. How will they feel it? Through sensors on the body, which will give pulses of different strengths. Thus, depending on the location of this sensor and the strength of the impulse, the blind person will know whether to stop or go further. Also these sensors will warn of dangers such as: a curb, a descent, a pole, etc.

Our own research: I did an experiment that can easily be done by anyone with basic programming skills. Using the Python programming language, I created a program that identifies a face in real time using the camera. For this I used the "face_recognition" library from Ageitgey. This program identifies a person's face and displays their name on the screen in a square. This is a great opportunity to experiment and create a prototype that allows the blind to record their name and face image in a database when they meet people. The next time they meet, their interlocutors will be pleasantly surprised to be recognized.

Conclusion: This work has described methods of improving the lives of blind people using computer technology. The use of artificial intelligence and functioning accessories - glasses that will receive all the information from the outside world, which with the help of the voice of the AI will be transmitted to the person was proposed. No similar studies have been found, so I propose this method of providing "vision" to blind people.

REFERENCES:

1. Engineering & Transportation (2017) «Numsense! Data Science for the Layman - Annalyn Ng, Kenneth Soo»[Text] / Annalyn Ng, Kenneth Soo // -p. 1-147. Available at [<https://leanpub.com/numsense>]

2. Literature on Deaf and Sign Language (2015) «Miller, VF Actual problems of social adaptation of the visually impaired and some ways to solve them» [Text] / VF Miller // Education and training of children with developmental disorders. - No. 6. - p.56-62. Available at [<https://elibrary.ru/item.asp?id=24245365>]

3. Tesla Autonomy Day is on Mon, April 22nd [Electronic resource](2021) –“Tesla Autonomy Day” [<https://www.youtube.com/watch?v=Ucp0TTmvqOE&t=5549s>]

4. Journal Proceedings of the Royal Society B [Electronic resource] (2001)- «Dispersed activation in the left temporal cortex for speech-reading in congenitally deaf people» [<https://royalsocietypublishing.org/doi/10.1098/rspb.2000.0393>]

УДК 004.415.3:681.6

О.І. Сироткіна¹, К.С. Родна, А.Г. Мединський¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПРОГНОЗУВАННЯ ARIMA ДЛЯ ЧАСОВИХ РЯДІВ

Анотація: Представлено аналіз ринку електроенергії. Побудовано моделі прогнозування ARIMA та ANN для спотових цін. Модель застосовується до часових рядів, що складаються з цін на електроенергію на день наперед від біржі енергії EPEX.

Ключові слова: прогнозування цін, електроенергія, прогностична модель, ACF, ARIMA, NNAR.

Вступ. Електроенергія є унікальним товаром на ринку, яким через його природу важко управляти. Його зберігання є дорогим, непродуктивним, і вимагає величезної кількості об'єктів. Отже, на відміну від інших видів торгової продукції, неможливо тримати її на складі і мати її доступною за запитом. Попит і пропозиція постійно змінюються і не завжди легко передбачувані. Такі якості електроенергії як товару визначають суть ринку електроенергії.

Головним викликом ринку електроенергії є забезпечення синхронізованості генерації та споживання та забезпечення достатньої потужності навіть під час піків попиту. Крім того, нещодавнє впровадження розумних мереж та інтеграція відновлюваних джерел енергії підвищує складність ринку та невизначеність майбутньої пропозиції та попиту. Саме тому прогнозування цін на електроенергію все більше набуває значення для всіх учасників ринку. Прогнозування цін на електроенергію зосереджене на прогнозуванні спотових і форвардних цін з різними часовими горизонтами для оптових ринків електроенергії.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені наступні завдання: проаналізувати основні правила роботи ринку електроенергії та стратегії його учасників; проаналізувати ціни на електроенергію за минулий період та зробити висновки щодо її природи;

попередньо обробити набір даних для наступного застосування прогностичних моделей; застосувати моделі ARIMA та ANN та проаналізувати результати; оцінити та проаналізувати можливість створення гібридних прогностичних моделей; зробити висновки щодо отриманих результатів.

Основний зміст роботи: Під час роботи над даним завданням були використані наступні методи та інструменти:

- Статистична модель ARIMA (авторегресивне інтегроване ковзке середнє);
- Авторегресивна нейронна мережа на основі нейронної мережи прямого поширення;
- Усі обчислення виконані в MATLAB.

Оптовий ринок електроенергії пропонує генераторам продавати свою електроенергію роздрібним торговцям. Специфічні характеристики електроенергії як активу створюють певні обмеження на енергоринку. Як звичайний товарний або фінансовий актив, її можна купити, продати або проміняти. Однак її природа створює наступні обмеження:

- Електрику не можна легко зберігати. Одним з варіантів є зберігання її в іншій формі енергії шляхом перекачування води в сховища, проте можливості зберігання обмежені (має бути достатньо місця для водойм). Іншим варіантом є використання акумуляторів, які неймовірно дорогі і екологічно недружні. Це означає, що електроенергія повинна споживатися відразу після її виробництва;

- Передача електроенергії передбачає збитки;
- Частота і відхилення напруги можуть призвести до великих втрат.

У глобальному масштабі ціна електроенергії залежить від різних факторів:

- Урядові нормативно-правові акти;
- Сусідні конкурентні ринки;
- Світові ринки;
- Зміни у виробництві;
- Імпорт та експорт;
- Фінансові спекуляції;
- Погодні умови;
- Технологічні інновації;
- Рівні попиту;
- Доступність постачання.

Всі ці фактори відіграють значну роль у процесі визначення ціни. Деякі з них можна передбачити тільки в короткостроковій перспективі, як погода, інші як в короткостроковій, так і в довгостроковій перспективі з різною точністю (пропозиція, попит). Інші або не можуть бути достатньо спрощені, щоб виступати в якості вхідних даних в моделі прогнозування, або майже непередбачувані за своєю природою.

Прогнозування цін на електроенергію зосереджене на прогнозування стопові та форвардні ціни на оптових ринках електроенергії. Всі гравці ринку використовують прогнози цін як фундаментальний внесок у механізми прийняття рішень на корпоративному рівні. Оскільки ціна вкрай нестабільна,

компаніям доводиться хеджувати не тільки обсяг (скільки потрібно електроенергії), але і її ціну. Витрати на помилки в купівлі або продажу стратегій і, як наслідок, купівля неправильної кількості електроенергії або в невідповідний час можуть призвести до величезних фінансових втрат або банкрутства. Більш точний прогноз полягає в тому, що генератор, роздрібний продавець або споживач може коригувати свою стратегію торгів або графік виробництва / споживання, щоб зменшити ризик або максимізувати прибуток. Важко недооцінити необхідність нових методів і моделей в прогнозі цін на електроенергію, оскільки ці прогнози є наріжним каменем на всьому ринку електроенергії.

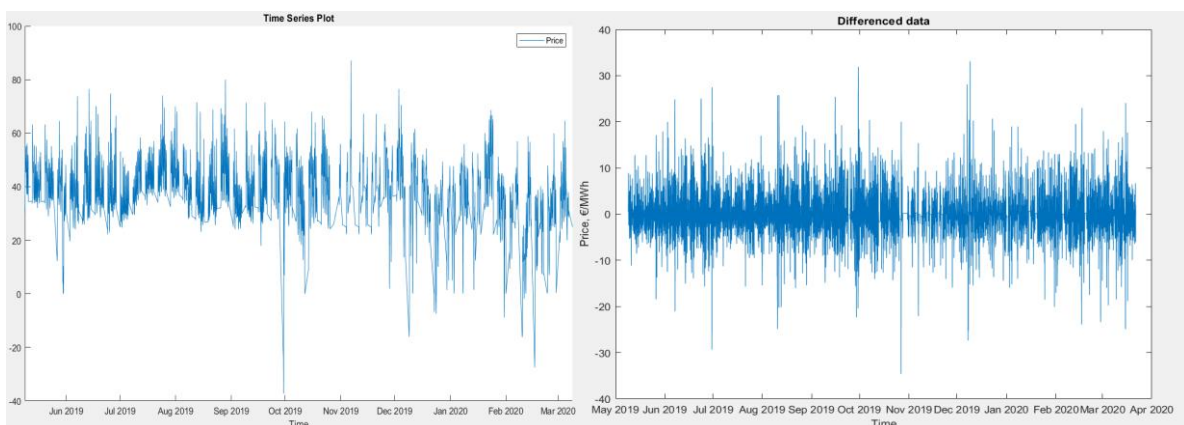
Прогноз робиться для Європейської енергетичної біржі (EPEX) з використанням історичних значень для ринку електроенергії на добу вперед.

Моделі ARIMA забезпечують статистичний підхід до прогнозування часових рядів. Для прогнозування майбутніх значень вони мають на меті описати автокореляції в історичних даних. ARIMA означає авторегресивне інтегроване ковзке середнє. Це поєднання двох різних моделей: авторегресивної моделі і моделі ковзкого середнього. Авторегресивна модель використовує минулі значення змінної для прогнозування її майбутніх значень. Ковзне середнє значення, зі свого боку, є середнім арифметичним набору попередніх значень протягом вказаної кількості кроків часу в минулому.

Щоб модель ARIMA була застосована, часовий ряд має бути стаціонарним. Стаціонарний часовий ряд - це той, чий властивості, такі як середнє значення, дисперсія, автокореляція, не залежать від часу, коли ряд спостерігається [1]. Таким чином, якщо часові ряди мають тренд або сезонність, то вони не є стаціонарними. Щоб позбутися від тренду та/або сезонності, використовується диференціація (рис 1). Після цього передбачається, що статистичні властивості будуть такими ж в майбутньому, як і в минулому.

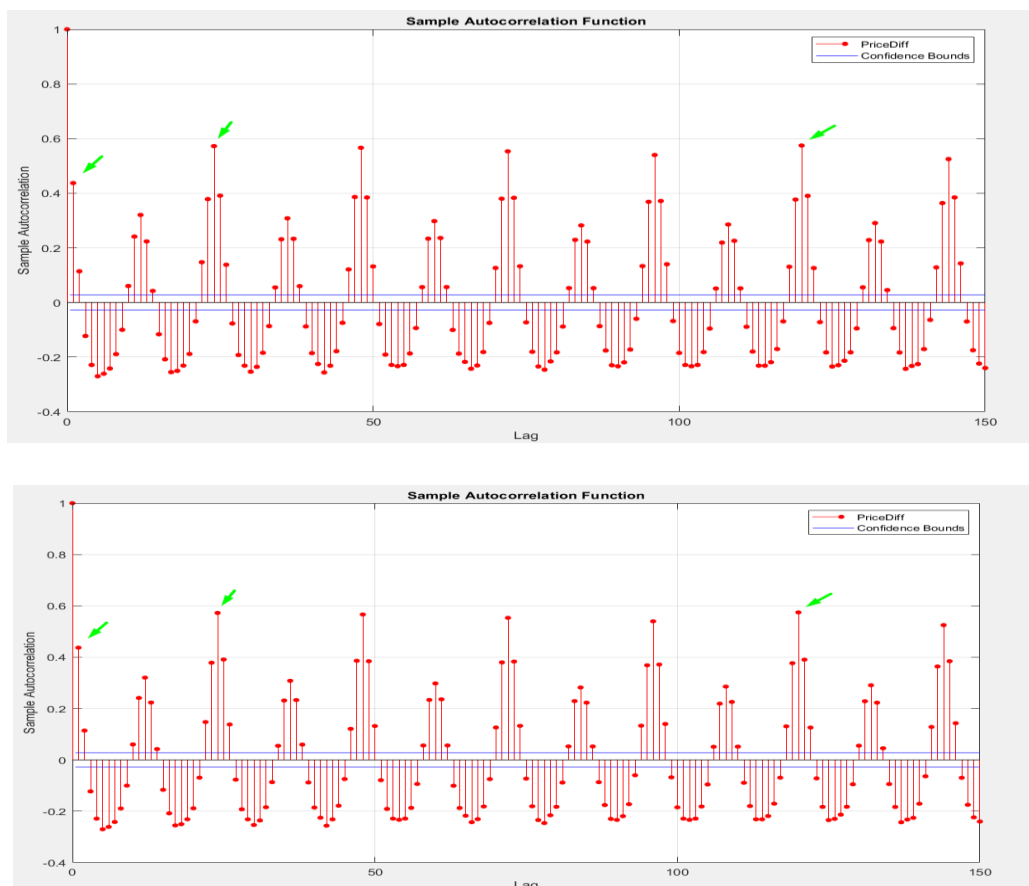
Модель ARIMA характеризується 3 термінами: p , d , q : де p – це порядок автоматичного регресії, q – це порядок ковзкого середнього, а d – порядок інтегрування, необхідний для того, щоб зробити часовий ряд стаціонарним.

Ця ділянка являє собою погодинні ціни на електроенергію на робочі дні на період з 09 травня 2019 року по 08 березня 2020 року.



1. Часовий ряд (ліворуч), диференційований ряд(праворуч)

Оскільки модель ARIMA використовує лише значення часового ряду, необхідно визначити кореляцію між ними. Для оцінки кореляції використовуються дві функції: Функція автоматичної кореляції (ACF) і функція часткової автоматичної кореляції (PACF) (рис 2).



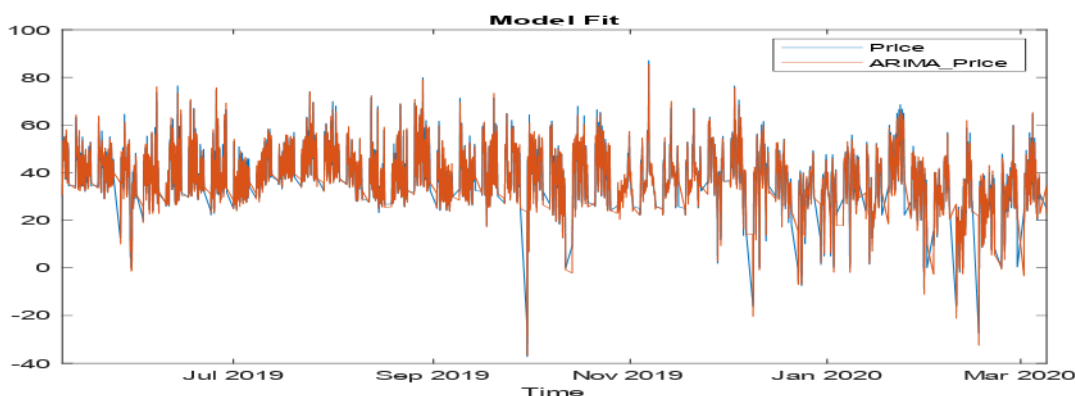
2. Автокореляція та часткове автокореляція

Розглядалися моделі ARIMA з різними параметрами AR і MA. На основі значень від ACF і PACF наступна модель ARIMA, як виявилось, є найкращою.

1

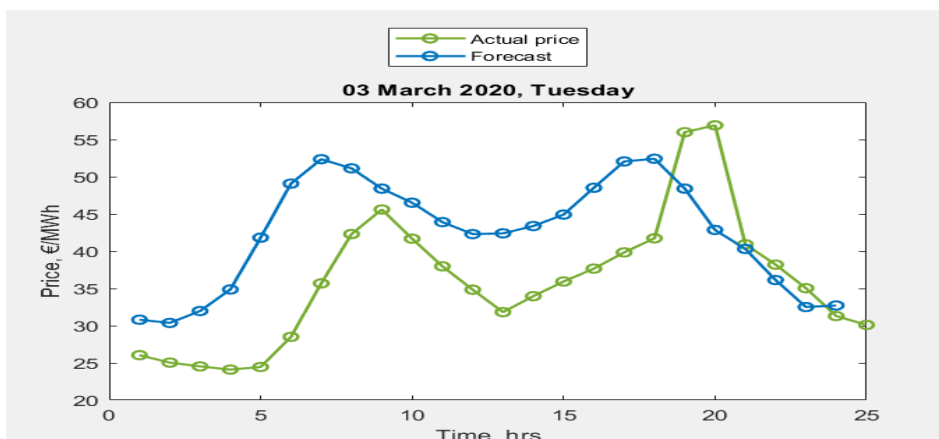
$$(1 - \phi_1 L - \phi_{24} L^{24} - \phi_{48} L^{48} - \phi_{120} L^{120})(1 - L)y_t = c + (1 + \theta_1 L + \theta_2 L^2)\varepsilon_t \quad (1)$$

де ϕ – параметр авторегресії; θ – переміщення середнього параметра; L – значення змінної; c – константа; ε – білий шум;

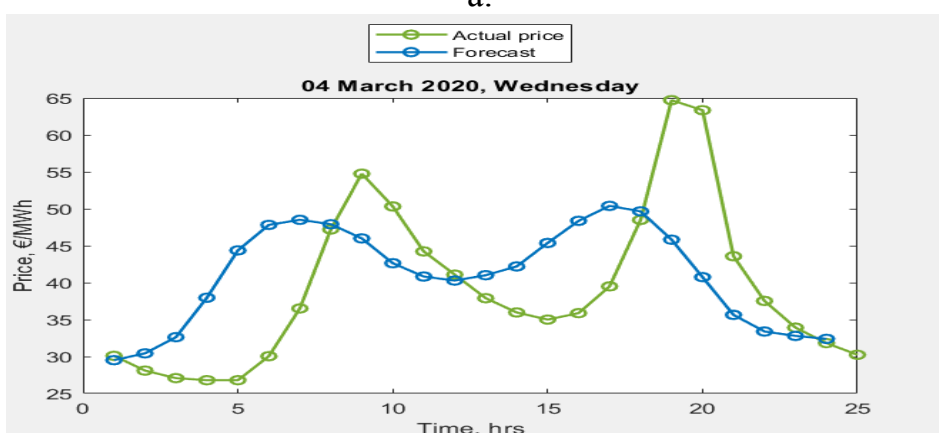


3. Отримана в результаті модель

На рис.4 знаходяться графіки фактичної та прогнозованої ціни



а.



б.

4. Порівняння фактичних (а) та прогнозованих (б) цін

Для пошуку нелінійних залежностей між цінами застосована нейромережа прямого поширення.

Вона приймає 24 минутих значення як вхідні дані і має 1 прихований шар з 12 нейронами (рис. 5).

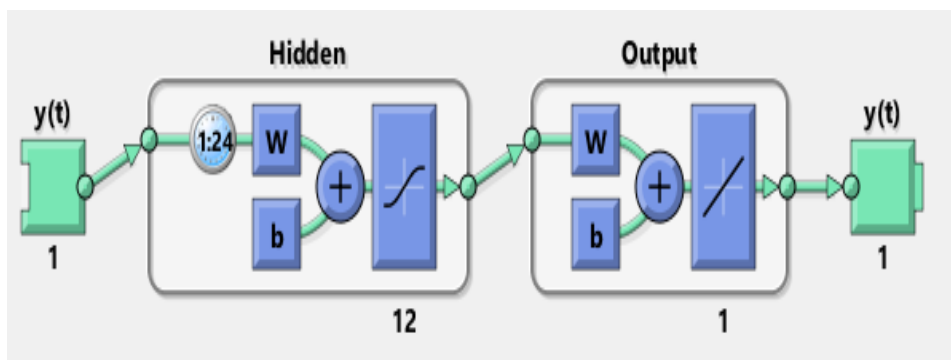
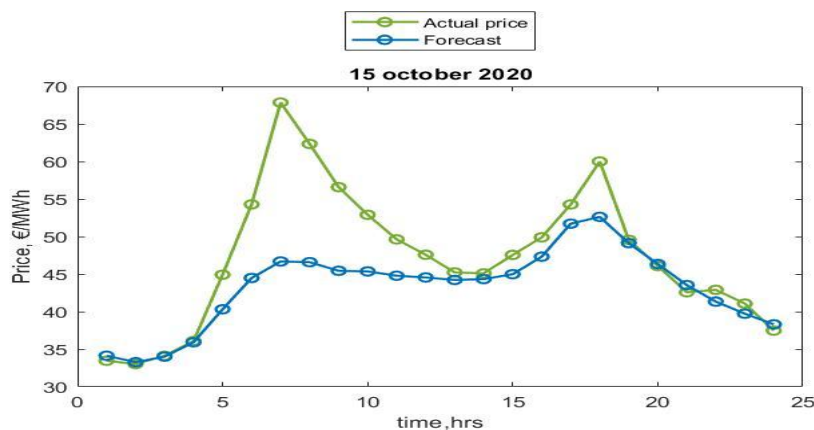


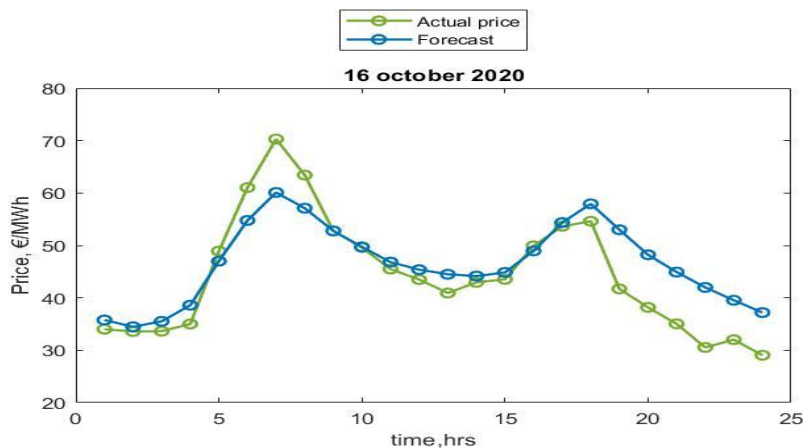
Рис. 5. Графічне відображення нейронної мережі

По-перше, модель навчається у відкритому циклі. Це означає, що вона використовує цільові значення як вхідні дані для підвищення продуктивності моделі. Потім вона перемикається на замкнутий цикл для отримання прогнозу

на час, який не включений у набір даних. Модель навчається за методом Байєсової регуляризації. Результати прогнозу моделі (рис. 6).



а.



б.

Рис. 6. Порівняння фактичних (а) та прогнозованих (б) цін

Наукова новизна результатів роботи полягає в удосконаленні методів прогнозування цін на електроенергію.

Висновки. Проаналізована структура та основні характеристики дерегульованого ринку електроенергії а також властивості та природу формування ціни. Створено та порівняно прогностичні моделі на основі статистичних методів та штучних нейронних мереж.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Contreras J., Espínola R., Member S., Nogales F., “ARIMA Models to Predict Next-Day Electricity Prices,” vol. 18, no. 3, pp. 1014–1020, 2003.
2. Weron R., “Electricity price forecasting: A review of the state-of-the-art with a look into the future,” Int. J. Forecast., vol. 30, no. 4, pp. 1030–1081, 2014.
3. “Electricity Market representation.” [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0169207014001083>.

РОЗДІЛ 3

ПРОГРАМНІ ЗАСОБИ УПРАВЛІННЯ, ЗБОРУ, ОБРОБКИ І ПЕРЕДАЧІ ІНФОРМАЦІЇ

УДК 004.415.3:681.6

І.М. Удовик¹, С.М. Мацюк¹ В.О. Ігнашев¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ОБРОБКИ ЗАПИТІВ ПОШУКОВИХ СИСТЕМ ТА ЇХ ФУНКЦІЇ У КОРПОРАТИВНІЙ РОЗВІДЦІ

Анотація. Описано процес покращення користувацького досвіду при роботі з пошуковими системами. Покращення досягається за рахунок аналізу обробки пошуковими системами запитів користувача та визначення точності та швидкості результатів.

Ключові слова: пошукова система, корпоративна розвідка, оператори розширеного пошуку, зображення, Яндекс, Google, Bing, TinEye.

Вступ. В останні десятиліття в світі бурхливо розвивається напрямок SEO – оптимізації та просування сайтів, що створює тісний зв'язок матеріалів та сторінок інтернет порталів із пошуковими системами, що, у свою чергу, робить пошукові системи незамінним інструментом корпоративної розвідки.

Актуальність досліджень в цьому напрямку підтверджується кількістю різних застосувань пошукових систем. Це є інструментом пошуку нової пісні лише за мотивом та ритмом, пошуку давніх друзів та знайомих, актуальних новин, нового будинку або нової машини, ресурсів для навчання, розважального контенту та безліч інших напрямків.

Широке коло завдань, що вирішується пошуковими системами, не дозволяє в даний час створювати універсальні та водночас точні запити, змушуючи розробляти текст запиту в залежності від цілі пошуку та очікуваних результатів.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- визначити поняття корпоративної розвідки;
- визначити алгоритми роботи пошукових систем;
- провести аналіз процесу обробки зображень пошуковими системи;
- спроектувати на основі проаналізованих матеріалів базу даних;
- зробити висновки щодо доцільності створення подібної бази.

Основний зміст роботи. Під час роботи над даною роботою були використані наступні методи та інструменти: пошукові системи Google, Bing, TinEye, Яндекс.

Пошукова система – це онлайн-служба (апаратно-програмний комплекс з веб інтерфейсом), що надає можливість пошуку інформації в Інтернеті. У просторіччі під пошуковою системою розуміють веб сайт, на котрому розміщено інтерфейс системи. Програмною частиною пошукової системи є пошукова машина (пошуковий рушій) – комплекс програм, що забезпечує функціональність пошукової системи. Більшість пошукових систем шукають інформацію на сайтах Інтернету, але існують також системи, здатні шукати файли на ftp-серверах, товари в інтернет-магазинах, та ін. Класифікація пошукових систем відбувається за метою пошуку, типу даних, за якими здійснюється пошук, масиві даних, у якому здійснюється пошук та геолокації. За такими параметрами пошукові системи можна поділити на такі категорії:

- загальні;
- локальні;
- спеціалізовані.

Загальні пошукові системи. До загальних пошукових систем відносяться ті системи, що виконують пошук за індексованими у мережі Інтернет без прив'язки до певних локальних кордонів (таких як країна, частина світу, континент) та мови запиту, можуть індексувати різні типи файлів - від тексту до файлів з певним розширенням (.doc, .pdf, .rar та ін.). Загальними пошуковими системи є найпопулярніші у світі системи, такі як Google, Bing, Duckduckgo, Yahoo.

До локальних пошукових систем відносяться системи, що орієнтовані на запити на певній мові або у певній геолокації. Вони також можуть індексувати різні типи файлів - від тексту до файлів з певним розширенням (.doc, .pdf, .rar та ін.), але показують гірший від загальних пошукових систем результат при пошуку на іншій мові або пошуку певної інформації за межами заданої геолокації. До локальних пошукових систем можна віднести німецьку систему Wseeker, українську UKR.NET, французьку Voilà та багато інших.

До спеціалізованих пошукових систем можна віднести системи, що були створені для певного типу задач та найкращий результат показують при пошуку лише певних запитів, таких як пошук медіафайлів, пошук наукової інформації, пошук зображень та пошук даних у неіндексованих сторінках мережі Інтернет.

Представниками цієї групи пошукових систем є система для пошуку файлів GlobalFileSearch, система для пошуку зображень Tineye, системи для пошуку наукової інформації ScienceDirect та Google Scholar, а також системи для пошуку серед неіндексованих сторінок у браузері Tor - Torch та NotEvil.

Корпоративна розвідка на основі відкритих даних, або OSINT – дисципліна розвідки, що складається з пошуку та аналізу інформації, здобутої у відкритих джерелах, таких як: публікації, статті, соціальні мережі, інтерв'ю, резюме, публічні реєстри, звіти, декларації та інші джерела, що є загальнодоступними та відкритими.

Методами корпоративної розвідки є:

- формулювання запитів у пошукових системах;
- аналіз інформації у соціальних мережах;

- моніторинг інтернет порталів;
- зіставлення фактів;
- пошук інформації у відкритих реєстрах.

Формулювання коректних та доцільних запитів у пошукових системах є основною складовою у корпоративній розвідці. Завдяки великій популярності просування сайтів та Інтернет-порталів у пошукових системах, правильно складений запит може вирішити поставлене завдання ще на етапі формулювання. Завдяки моніторингу і аналізу соціальних мереж, можна дізнатися про особисті переваги, смаки та хобі людини, дізнатися її коло спілкування, близьких людей та родичів, роботу та геопозицію. Якщо ж мова йде про юридичну особу, наприклад, компанію, проаналізувавши акаунти можна дізнатись штат, поточні проекти, зміни у топ-менеджменті та навіть фінансові показники. Моніторинг інтернет-ресурсів спрямований на спостереження та аналіз змін на сторінках порталу, що може визначити зміну власника порталу, отримати контактні дані або компрометуючий матеріал. Особливо корисним є аналіз сторінки у минулому, тобто проаналізувати, як виглядала сторінка у минулому. Для цього використовується або кеш пошукових систем, або портал InternetArchive. Зіставлення фактів - основна частина роботи у корпоративній розвідці. Завдяки цьому формуються певні висновки щодо поставленої задачі. Реєстри є невід'ємною складовою пошуку інформації, адже вони містять точну та достовірну інформацію. У більшості випадків використовується інформація з судових реєстрів, аби дізнатися про судові справи, адміністративні та кримінальні правопорушення об'єкта та реєстри з інформацією по юридичним особам, котрі можуть містити важливу інформацію про власників, бенефіціарів, засновників та пов'язаних з об'єктом інших осіб.

Корпоративна розвідка є сферою, у котрій неможливо скласти однаково успішний алгоритм дій для досягнення результатів у будь-яких завданнях, на це впливає значна варіативність завдань та доступної у конкретному випадку інформації, але можна виділити загальний план, на який можна опиратися при роботі, а саме:

- формулювання технічного завдання;
- пошук необхідної інформації;
- аналіз отриманих даних;
- проєкція результату на технічне завдання;
- висновок.

Важливою особливістю корпоративною розвідки є увага до деталей, бо саме зіставлення непомітних, на перший погляд, деталей є ключовим фактором до результативної роботи. Також, завдяки гнучкості сфери, існує безліч методик пошуку та аналізу інформації. У якості базових методик, що є універсальними, є прогресивний джипег, або ітеративний метод, та метод дерев. У основі ітеративного методу є збір максимальної кількості інформації за короткий проміжок часу, після чого увесь знайдений матеріал аналізується та сортується, переглядається, та вибірково дошукується у логічних та смислових

розбіжностях. Метод дерев передбачає пошук безперервний пошук інформації по заціпкам - від однієї до іншої, таким чином, пошук інформації нагадує дерево з гілками, на кожній гілці якого ще по декілька гілок.

У корпоративній розвідці багато роботи з різноманітними типами даних: звіти, код програми, виписки з реєстрів, і т.д. Одним з типів даних, на який рідко звертають увагу, є зображення, такі як зображення профіля у соціальних мережах та форумах, фотографії у небезпечному програмному забезпеченні, логотипи компаній та інші. Виконуючи пошук за зображенням, з'являється можливість отримати багато інформації про об'єкт досліджень. Для деяких випадків може бути достатньо пошуку зображень за допомогою звичайної пошукової системи, такої як Google або Яндекс. Наприклад, якщо у роботі трапився незнайомий логотип, виконавши простий пошук зображення можна отримати інформацію про походження логотипу, інформацію про бренд або геолокацію. Іноді, при спробі виконати пошук зображення може бути корисним змінити початкове зображення для досягнення найкращих результатів. Наприклад, іноді зображення може бути опубліковано та заявлено, що воно є оригіналом, коли в дійсності це відзеркалена або перевернута версія вже існуючого фото. Повертаючи або відзеркалюючи фотографію та виконуючи пошук зміненого зображення, є ймовірність знайти додаткові результати, які, можливо, не були отримані при пошуку початкової версії. Також, клопітливе обрізання фотографії може дати кращі результати, так як інші об'єкти на зображенні можуть заважати пошуковій системі сфокусуватися на основному об'єкті. Хоча зображення та фотографії не являються найпоширенішим джерелом інформації, цей метод може бути корисним в деяких випадках, коли необхідна ідентифікація людини або логотипу.

Наукова новизна: удосконалено метод визначення ефективності оброблення запитів пошуковими системами та їх застосування у корпоративній розвідці.

Висновки. В результаті виконання роботи було спроектовано надійну та ефективну базу знань для роботи із пошуковими системами. У тому числі, було обрано найбільш важливі та поширені критерії класифікації об'єктів зображень, такі як геолокація, обличчя, деталі, будівля, лого та ресайз.

ПЕРЕЛІК ПОСИЛАНЬ:

1. An Introduction to Computer Security: The NIST Handbook. Draft. National Institute of Standards and Technology, Technology Administration, U. S. Department of Commerce, 1994. - 310 p.
2. Cheswick W. R., Bellovin S. M. Firewalls and Internet Security: Repelling the Wily Hacker. - Addison-Wesley, 1994. - 275 с.
3. Браун С. Мозаика и Всемирная паутина для доступа к Internet: Пер. с англ. М.: Мир, Малиц, СК Пресс, 1996. - 167с.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗШИРЕННЯ ФУНКЦІЙ САПР МОДЕЛЕЙ РЕАЛЬНИХ ОБ'ЄКТІВ В РЕЖИМІ ДОПОВНЕНОЇ РЕАЛЬНОСТІ.

Анотація. Описано процес поліпшення методів та алгоритмів, які дозволяють демонструвати тривимірні об'єкти, будь-якої складності, створені в САПР Autodesk Inventor в режимі доповненої реальності. Розширення функціоналу САПР досягається завдяки розробці плагіна, який відповідає за конвертацію моделі і її відображення в поле сприйняття користувача

Ключові слова: *плагін, САПР, доповнена реальність, валідація, конвертація.*

Вступ. Зростання продуктивності інтегрованих графічних процесорів, а також досягнення в областях комп'ютерного зору і комп'ютерної графіки стали причиною появи безлічі нових способів сприйняття оточуючого нас світу. Одним їх таких способів є доповнена реальність яка є варіацією більш об'ємного поняття – віртуальна реальність.

Технології віртуальної реальності повністю занурюють користувача всередину синтетичного оточення, згенерованого комп'ютером. Перебуваючи в віртуальному оточенні, користувач не може бачити навколишній реальний світ. Навпаки, доповнена реальність дозволяє користувачеві бачити навколишній світ з віртуальними об'єктами, накладеними на реально існуючі предмети. Отже, доповнена реальність доповнює навколишній світ, а не повністю його заміщає. У самому ідеальному варіанті користувач буде не в змозі відрізнити реально існуючий об'єкт від віртуального об'єкта введеного в його поле зору.

В даний момент технології доповненої реальності знаходяться на піку своєї популярності так як стрімко зростає кількість додатків які використовують технологію доповненої реальності. Це обумовлено багатьма факторами, одним з яких є, як це вже згадувалося раніше, зростання продуктивності апаратних платформ, на яких ці самі додатки будуть виконуватися а також появи безлічі засобів розробки, які дозволяють фахівцям з програмного забезпечення з легкістю створювати додатки, що використовують доповнену реальність

Незважаючи на всі досягнення, які відбулися в останні роки, все ще існує безліч проблем, після вирішення яких фінальною метою буде генерація віртуальних об'єктів, які будуть настільки реалістичними, що вони будуть не відрізняються від навколишнього середовища

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- дослідити поняття та характеристики доповненої реальності;

- провести аналіз існуючих наборів засобів розробки для роботи з доповненою реальністю;
- визначити вимоги для розробки плагіна для САПР Autodesk Inventor;
- спроектувати архітектуру програми, яка буде відображати моделі в режимі доповненої реальності;
- зробити висновки щодо доцільності створення подібної системи.

Основний зміст роботи. Під час роботи над даною роботою були використані наступні методи та інструменти:

- методи відстеження зображень, для використання їх в якості початку координат;
- алгоритми комп'ютерного зору;
- C#, Net Framework, Unity Engine, Visual Studio, Batch, EasyAR, Inventor API.

Доповнена реальність - це інтерактивний досвід реального середовища, де об'єкти, що перебувають у реальному світі, доповнюються генеровану комп'ютером інформацією, іноді через різні сенсорні модальності, включаючи зорові, слухові, тактильні, сомато-сенсорні та нюхові. Доповнену реальність можна визначити як систему, яка виконує три основні функції: поєднання реального та віртуального світів, взаємодію в реальному часі та точну 3D-реєстрацію віртуальних та реальних об'єктів. Накладена сенсорна інформація може бути конструктивною (додавати інформацію до природного середовища) або деструктивною (тобто маскувати природне середовище). Цей досвід безперешкодно переплітається з фізичним світом так, що сприймається як захоплюючий аспект реального середовища. Таким чином, доповнена реальність змінює постійне сприйняття реального середовища, тоді як віртуальна реальність повністю замінює реальне середовище користувача на модельоване. Доповнена реальність пов'язана з двома в основному синонімічними термінами: змішаною реальністю та комп'ютерно - опосередованною реальністю.

У віртуальній реальності сприйняття реальності користувачами повністю базується на віртуальній інформації. У доповненій реальності користувачеві надається додаткова інформація, що генерується комп'ютером, та покращує їх сприйняття реальності. Наприклад, в архітектурі VR можна використовувати для створення прохідного моделювання внутрішньої частини нової будівлі; а AR можна використовувати для відображення структур та систем будівлі, накладених на реальний погляд. Інший приклад - використання службових програм. Деякі додатки AR, такі як Augment, дозволяють користувачам застосовувати цифрові об'єкти в реальних середовищах, дозволяючи компаніям використовувати пристрої доповненої реальності як спосіб попереднього перегляду своїх продуктів у реальному світі. Подібним чином, доповнену реальність також можна використовувати для демонстрації того, як можуть виглядати товари в середовищі для споживачів, як це демонструють такі компанії, як Mountain Equipment Co-op або Lowe, які використовують

доповнену реальність, щоб дозволити клієнтам переглянути, як можуть виглядати їхні товари вдома завдяки використанню 3D-моделей.

Доповнена реальність відрізняється від віртуальної реальності тим, що в доповненій реальності частина навколишнього середовища насправді є «реальною» і просто додає прошарок віртуальних об'єктів до реального середовища. З іншого боку, в віртуальній реальності навколишнє середовище є абсолютно віртуальним. Демонстрація того, як AR-шари об'єктів у реальному світі можна побачити за допомогою ігор з доповненою реальністю. WallaMe - це програма для доповненої реальності, яка дозволяє користувачам приховувати повідомлення в реальному середовищі, використовуючи технологію геолокації, щоб дозволити користувачам приховувати повідомлення, де б вони не побажали у світі. Такі програми мають багато застосувань у світі, в тому числі для активізму та художньої виразності.

Основним проєктним рішенням у побудові системи AR є те, як здійснити поєднання реального та віртуального. Доступні два основні варіанти: оптична та відеотехніка. Кожен має певні переваги та недоліки. Цей розділ порівнює ці два і зазначає компроміси. Прозорий HMD – це один пристрій, що використовується для поєднання реального та віртуального. Стандартні HMD із закритим видом не дозволяють мати прямий погляд на реальний світ. На відміну від цього, пронизаний HMD дозволяє користувачеві бачити реальний світ за допомогою віртуальних об'єктів, накладених за допомогою оптичних або відеотехнічних засобів.

Оптичні прозорі HMD працюють, розміщуючи оптичні суматори перед очима користувача. Ці комбайнери є частково пропускаючими, так що користувач може дивитись прямо через них, щоб побачити реальний світ. Оптичні суматори також частково відображають, так що користувач бачить, як віртуальні зображення відбиваються від комбайнерів від моніторів, встановлених на голові. Цей підхід за своєю суттю схожий з головними дисплеями (HUD), які зазвичай використовуються у військових літаках, за винятком того, що комбайнери прикріплені до голови. Таким чином, оптичні прозорі HMD іноді описуються як «HUD на голові». На рис. 1 представлена концептуальна схема оптичного прорізу HMD. На рис. 2 показані два оптичні прозорі HMD, виготовлені X'юзом Електроніка.

Оптичні комбайнери зазвичай зменшують кількість світла, яке користувач бачить із реального світу. Оскільки комбайнери діють як напівсріблені дзеркала, вони пропускають лише частину світла з реального світу, щоб вони могли відображати частину світла від моніторів в очі користувача. Наприклад, HMD, описаний в, передає близько 30% надходить світла з реального світу. Вибір рівня змішування - проблема дизайну. Більш складні комбайнери можуть змінювати рівень внесків залежно від довжини хвилі світла. Наприклад, такий комбайн може бути встановлений для відображення всього світла певної довжини хвилі і жодної на будь-якій іншій довжині хвилі. Це було б ідеально з монохромним монітором. Практично все світло від монітора буде відбиватися в очах користувача, тоді як майже все світло з реального світу (за винятком певної довжини хвилі) досягне очей користувача. Однак більшість існуючих

оптичних прозорих HMD дійсно зменшуються кількість світла з реального світу, тому вони відірвані від сонячних окулярів, коли відключено живлення.

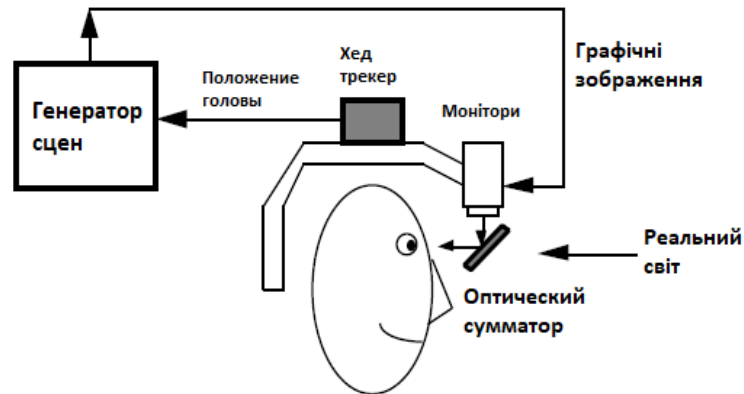


Рис. 1. Концептуальна схема оптичного прозорого наголовного дисплея

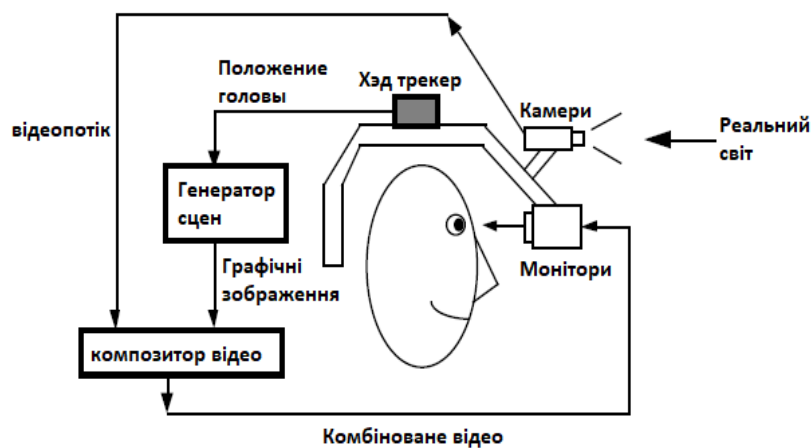


Рис. 2. Концептуальна схема наголовного відео дисплея

На відміну від цього, відеопрозорі HMD працюють, поєднуючи HMD із закритим видом з однією або двома відеокамерами, встановленими на голові. Відеокамери надають користувачеві погляд на реальний світ. Відео з цих камер поєднується з графікою зображення, створені генератором сцен, поєднуючи реальне та віртуальне. Результат є надсилається на монітори перед очима користувача в HMD із закритим видом. Рисунок 1 показує концептуальну схему відеопрозорого HMD. На рис. 2 показано фактичний відеопрозорий HMD, з двома відеокамерами, встановленими поверх льотного шолома.

Композиція відео може бути виконана більш ніж одним способом. Простий спосіб - використовувати кольорову маніпуляцію: техніку, що використовується у багатьох відео спецефектах. Тло графічних зображень комп'ютера встановлено в певний колір, скажімо зелений, який не використовує жоден з віртуальних об'єктів. Потім крок комбінування замінює всі зелені зони на відповідні частини з відео реального світу. Це має наслідком накладання віртуальних об'єктів на реальний світ. Більш досконала композиція використовує інформацію про глибину. Якби система мала інформацію про глибину на кожному пікселі для зображень у реальному світі, вона могла б

поєднувати реальні та віртуальні зображення за допомогою піксельного порівняння глибини. Це дозволило б реальним об'єктам покривати віртуальні об'єкти і навпаки.

Наукова новизна: були вдосконалені методи, що дозволяють розглядати тривимірні моделі створені в САПР Autodesk Inventor.

Висновки. В результаті виконання роботи була підвищена ефективність роботи САПР Autodesk Inventor для демонстрації тривимірних об'єктів будь-якої складності в режимі доповненої реальності. Збільшення ефективності стало можливим завдяки розробленому плагіну, який інтегрується з вищезгаданою САПР, та надає користувачу доступ до функціонала який раніше не існував.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Wloka, Matthias M. and Brian G. Anderson. Resolving Occlusion in Augmented Reality. Proceedings of 1995 Symposium on Interactive 3D Graphics (Monterey, CA, 9-12 April 1995), 5-12.

2. State, Andrei, Mark A. Livingston, Gentaro Hirota, William F. Garrett, Mary C. Whitton, Henry Fuchs and Etta D. Pisano. Techniques for Augmented-Reality Systems: Realizing Ultrasound-Guided Needle Biopsies. Proceedings of SIGGRAPH '96 (New Orleans, LA, 4-9 August 1996), 439-446.

3. Meyer, Kenneth, Hugh L. Applewhite, and Frank A. Biocca. A Survey of Position-Trackers. Presence: Teleoperators and Virtual Environments 1, 2 (Spring 1992), 173-200.

2. Rose, Eric, David Breen, Klaus Ahlers, Chris Crampton, Mihran Tuceryan, Ross Whitaker, and Douglas Greer. Annotating Real-World Objects Using Augmented Reality. Proceedings of Computer Graphics International '95 (Leeds, UK, 25-30 June 1995), 357-370.

3. Hiller, Jonathan D.; Lipson, Hod (2009). "STL 2.0: A Proposal for a Universal Multi-Material Additive Manufacturing File Format" (PDF). Cornell University. Retrieved 5 May 2017.

УДК 677.055.621

І.В. Новицький¹, В.В. Слесарев¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ПРОГРАМНА РЕАЛІЗАЦІЯ ОЦІНКИ НАДІЙНОСТІ СИСТЕМИ

Анотація. Запропонована модель для оцінки надійності технічної системи з урахуванням ймовірнісних характеристик окремих її станів.

Ключові слова: програмна реалізація, апарат марковських процесів, ймовірності безвідмовної роботи, технічне обслуговування, проектуванні технічних об'єктів.

Вступ. У статті розглянута програмна реалізація одного з методів розрахунку показника якості системи технічного обслуговування (ТО). Як показник якості обраний функціонал, що характеризує відносний час перебування об'єкта в

працездатному стані. Основними факторами, що впливають на ефективність функціонування системи ТО, є показники надійності об'єкта, а також час пошуку (індикації) відмов і тривалості планових і аварійно-відновлювальних робіт. Метод дозволяє при проектуванні технічних об'єктів заздалегідь оцінити достатність передбачених обсягу і номенклатури регламентних робіт для отримання оптимальної стратегії ТО.

Постановка задачі. Технічне обслуговування, наприклад, енергетичних об'єктів, таких як газотурбінні установки (ГТУ), котлоагрегати і т. п., є важливим компонентом системи експлуатації та включає комплекс робіт для підтримання справності або тільки працездатності установок при їх підготовці до використання за призначенням при зберіганні і транспортуванні.

Основний зміст роботи. Відомо, що необхідність ТО викликається головним чином тим, що в конструкції складних технічних об'єктів практично неможливо реалізувати принцип рівної міцності всіх вузлів і деталей. Тому існують елементи з меншим ресурсом, ніж загальний призначений ресурс. Крім того, внаслідок дії на об'єкт комплексу факторів зовнішнього середовища інтенсивність відмов деталей, вузлів, систем буде різною, в тому числі зростаючою зі збільшенням напруження. Встановлено, що в ГТУ такими «вузькими місцями» є лопатковий апарат і інші елементи проточної частини, які піддаються дії високих температур, а також промислового пилу, піску і інших агресивних включень в повітрі, що рухається по циклу.

При розробці нових енергетичних об'єктів, при модернізації існуючих, а також при продовженні призначеного ресурсу виникає завдання створення або вдосконалення системи технічного обслуговування. Створення ефективної системи ТО забезпечить на практиці повну реалізацію призначеного ресурсу об'єкта, скоротить час простою через відмови, зменшить витрати на експлуатацію. Розробці такої системи ТО (інакше, оптимальної стратегії технічного обслуговування) повинне передувати дослідження її можливостей при певних обмеженнях на матеріальні витрати і на тимчасові ресурси.

Відомо, що технічне обслуговування як комплекс робіт включає організаційні і технічні заходи, які в загальному випадку можуть розглядатися як система з певною структурою і відповідними зв'язками між елементами. Вона має чітко визначені цілі функціонування і реалізується в практиці експлуатації об'єктів через певні моделі [1].

Головним у виборі моделі технічного обслуговування і тим самим у формуванні структури системи є наявність вихідної інформації - значень показників надійності об'єкта експлуатації, а також можливість їх оперативної оцінки. Наявність або відсутність інформації про показники надійності об'єкта, в свою чергу, визначається внутрішніми властивостями об'єкта і організаційною структурою системи експлуатації. Ефективність конкретної моделі ТО оцінюється показниками, які розробляються з використанням заздалегідь встановлених показників надійності об'єкта.

Отримано досить точні рішення задачі оптимізації термінів ТО об'єктів (простих і структурно-складних) з різною природою відмов – поступових і раптових, а також при різних законах їх надійності. При цьому розглядаються пристосованість об'єктів для

індикації відмов (вбудований контроль) і певні характеристики їх ремонтпридатності для випадків планового і аварійного відновлення працездатності.

На основі аналізу і узагальнення існуючих методів створена пропонується методика оцінки ефективності системи ТО.

У загальному випадку процес функціонування об'єкта розглядається як випадковий процес у часі з обмеженою кількістю можливих станів. Тому траєкторія процесу $x(\tau)$ буде ступінчастою. Вона задається кількістю переходів m , моментами переходів і набором станів, в яких об'єкт перебував між моментами переходу. Як показник, за яким можна судити про оптимальність обраної стратегії ТО, можна прийняти функціонал, визначений на множині можливих траєкторій випадкового процесу $x(\tau)$. Для обчислення показника ефективності слід розрахувати час знаходження об'єкта в кожному з виділених станів. При вирішенні даного завдання доцільно використовувати апарат марковських процесів [1].

Стає доцільним розгляд питання про моделювання цього процесу. Тоді метою цього моделювання буде збереження або перетворення інформації по спеціальній структурі, що вносить порядок в даний об'єкт, елементи об'єкта і їх відносини.

Було розроблено програмне забезпечення для обчислення ймовірності знаходження системи в тому чи іншому стані. В якості вхідних специфікацій використовується матриця станів. Вихідною специфікацією є система рівнянь Колмогорова [2].

Коментування програми :

У створеному програмному забезпеченні було застосовано два модуля і наступні процедури:

- **TForm 1** – в цьому модулі вводяться і обробляються дані описують безперервний ланцюг Маркова.

- procedure Edit1Change – процедура обмежень на введення кількості станів.

- procedure Button1Click – по змінній з процедури Edit1Change заповнює матрицю станів первинними даними.

- procedure Button2Click – по матриці станів розраховує рівняння Колмогорова і викликає модуль TForm 2.

- **TForm 2** – в цьому модулі виводяться дані на екран.

- procedure Button1Click – процедура виклику модуля TForm 1 для повторного застосування програми.

- procedure Button2Click – процедура закриття програми по закінченню всіх робіт.

При запуску програми перед користувачем з'являється наступне вікно (рис. 1):

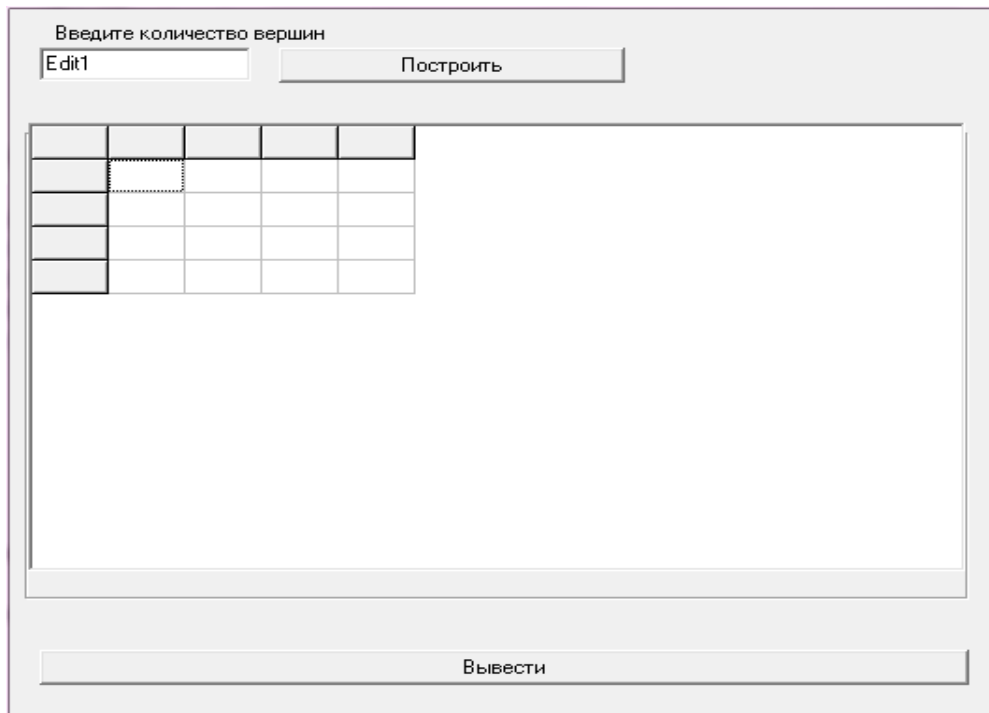


Рис. 1. Форма пуску програмного забезпечення

У вікно Edit1 вводиться кількість вершин. При натисканні на кнопку «Побудувати» формується матриця станів з уже заданим користувачем кількістю вершин. Далі користувач повинен заповнити матрицю станів, тобто змінювати значення комірок з «0» на «1». Слід змінювати тільки осередки зі значенням «0». Натискання кнопки «вивести» відкриває другу форму і в вікно виводить отриманий результат. Кнопка «Запустити знову» дозволяє повторити операцію з початку. Кнопка «Вихід» завершує роботу з програмою.

Рішення системи – результат роботи програми, виконується функцією $f(p, a, b, n, D)$, в якій параметр p ($1: q$) являє собою вектор початкових значень ймовірностей станів об'єкта (початкові умови інтегрування); a, b - ліва і права межі інтегрування; n - число інтервалів розбиття області інтегрування; D - матриця, складена з правих частин рівнянь системи. Ймовірність працездатного стану $p_1(t)$ відповідає значенню коефіцієнта готовності об'єкта $K_g(t)$.

Інтеграл від цієї ймовірності в інтервалі від нуля до часу, відповідного її стаціонарному значенню, визначає середній час безвідмовної роботи об'єкта з урахуванням можливих переходів в інші стани [3]:

$$T_{CP} = \int_0^t p_1 \epsilon d\tau \quad (1)$$

Якщо обчислити аналогічні інтеграли від інших значень ймовірностей станів, то можна отримати середній час перебування об'єкта в кожному з виділених станів. Тоді частка від ділення часу працездатного стану T_{CP} на суму всіх інших дасть функціонал, що характеризує якість функціонування системи технічного обслуговування об'єкта

$$J = \left(\int_0^t p_1 \langle d\tau \rangle \right) / \left(\sum_{i=2}^n \int_0^t p_i \langle d\tau \rangle \right) \quad (2)$$

де n - кількість станів ГТУ.

Наукова новизна: При обчисленні по рівняннях (1), (2) інтегрування ймовірностей станів об'єкта в запропонованому алгоритмі виконується після того, як вектори значень ймовірностей апроксимуються кубічними сплайнами.

Висновки. Підвищити значення функціоналу J можна за рахунок збільшення ймовірності безвідмовної роботи (забезпечення надійності об'єкта).

ПЕРЕЛІК ПОСИЛАНЬ

1. Вопросы математической теории надежности / Барзилович Е. Ю., Беляев Ю. К., Каштанов В. А. и др.; Под ред. Гнеденко Б. В.— М.: Радио и связь, 1983-184 с., ил.
2. Барзилович Е. Ю. Модели технического обслуживания сложных систем: Учебное пособие. - М.: Высш. школа, 1982.- 231 с., ил.
3. Байхельт Ф., Франкен П. Надежность и техническое обслуживание. Математический подход: пер. с нем - М.: Радио и связь, 1988.- 392 с., ил.

УДК 004.415.3:681.6

В.І. Корнієнко¹, С.М. Мацюк¹, В.В. Рудь¹

¹Національний технічний університет "Дніпровська політехніка", Дніпро, Україна

МЕТОДИ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ БІОНІЧНОГО ХАПАННЯ З УНИКНЕННЯМ ПЕРЕШКОД ЗА ДОПОМОГОЮ ДЕРЕВ ОКТАНТІВ ТА ГЛИБОКОГО НАВЧАННЯ

Анотація. Описано процес розробки системи керування роботоманіпулятором. Система є комплексним рішенням для розпізнання об'єктів, визначення поз хапання, розпізнання перешкод, побудова шляху до об'єкту з уникненням перешкод, хапання з перевіркою проковзування.

Ключові слова: *робот, маніпулятор, ROS, MoveIt, YOLO, Fin-Ray ефект, хапання, розпізнання об'єктів, дерево октантів, хмара точок, 3D-друк.*

Вступ. Важливим напрямом сучасних технологій є робототехніка. Вона стрімко розвивається у напрямках автоматизації виробничих процесів, взаємодій з об'єктами тощо.

Управління рукою робота - одна з найважливіших частин програмного забезпечення для робототехніки. Областями керування маніпулятором є: розпізнання об'єктів та перешкод, визначення об'єктів для хапання, визначення надійних позицій пристроєм хапання, планування руху маніпулятора до визначених позицій з уникненням перешкод та розпізнання проковзування або визначення надійного хапання.

Існують рішення для використання у зазначених областях, таких як керування рукою, визначення позицій для хапання, тощо. Але поєднання всіх існуючих рішень та підходів може бути більш складним завданням. Остаточне рішення повинно поєднувати найбільш відповідне рішення у кожній області та об'єднувати їх як цілісну систему.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- викласти принципи управління роботом маніпулятором;
- визначити основні рішення у кожній області;
- обрати найбільш доцільні рішення;
- об'єднати рішення у цілісну систему шляхом розробки взаємодії елементів системи;
- протестувати розроблену систему
- зробити висновки щодо доцільності створення подібної системи.

Основний зміст роботи. У маніпулюванні об'єктами є кілька сфер: захоплення, планування руху, виявлення ковзання. Спілкування між ними не встановлено. Аналіз останніх досліджень та публікацій. Досліджувана проблема актуальна при розробці роботизованої руки. Встановлено, що деякі дослідження та література згадують заявлену проблему. Але майже всі вони зосереджені на незалежних рішеннях, тоді як проблема полягає в інтеграції малих цільових рішень у цілісну систему.

Існує кілька основних підходів до управління рукою. У тому числі MoveIt! [2], хмарна хмара для захоплення [11].

Автори [10] описують відповідний алгоритм для уникнення самостійного зіткнення руки. Ця стаття демонструє повний алгоритм, але не демонструє інтеграцію з уникненням зіткнення навколишнього середовища.

Автори [7] описують повне алгоритмічне рішення, але без реалізації.

Хапання. Однією з найбільш значущих сфер взаємодії робота і навколишнього середовища є маніпулювання предметами. Маніпуляція може бути частиною багатьох надзвичайно корисних процесів: складання, сортування, переміщення, замовлення, завантаження вантажу тощо. Ці процеси використовуються у промислових та випадкових роботах.

Роботизовані маніпулятори повинні мати кінцевий ефектор, також відомий як Кінцевий інструмент. Кінцеві ефекти роботів поділяються на наступні типи:

- Неефективний: фізично схопити, приклавши безпосередню силу до об'єкта.
- Проникнення: проникає на поверхню предмета, щоб підняти його.
- Звужувальні: застосовуються сили притягання, такі як вакуум або магнетизм.
- Контигутивний: вимагає прямого контакту для зчеплення.

Розглянута проблема пов'язана з ударним типом захоплювача.

Fin-Ray ефект – це ефект, який описує гнучку конструкцію, яка нахилиється навколо об'єкта при застосуванні сили до гнучкої конструкції. Така конструкція дає додаткову площу контакту, яка забезпечує додаткове

тертя. Промені рибних плавників надихнули структуру. Плавники риби мають будову з двома кістками, які прикріплені один до одного еластичною тканиною. Хвостовий плавник – основна точка прикладання сили для руху. Олово складається з декількох основних конструкцій, складених одна над одною. Конструкція повинна бути легкою, але досить міцною, оскільки надмірна вага призведе до непотрібних втрат енергії. Конструкція, яка копіює рибний плавник, складається з двох прикріплених поздовжніх волокон. Серед поздовжніх волокон є поперечні волокна, які зберігають всю конструкцію після складання. Поперечні та поздовжні волокна з'єднані гнучко, що забезпечує необхідний рух між ними [5].

Матеріал пальців. Дослідницька група постала перед вибором способу виготовлення пальців для захоплення. 3D-друк було обрано як найпростіший спосіб, який відповідає усім вимогам. Пальці виконуються за допомогою FDM 3D-друку на матеріалі TPU. Інші частини корпусу виготовлені з алюмінію та пластику з ABS.



Рис. 1. Ефект Фін-Рей

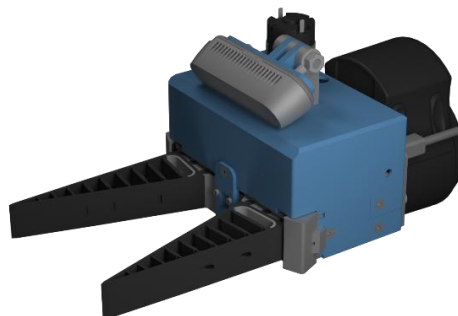


Рис. 2. 3D-модель використаного захоплення

Сприйняття навколишнього середовища. Далі ми повинні виявляти та зберігати інформацію про навколишнє середовище. Система повинна сканувати навколишнє середовище за допомогою пристроїв введення сприйняття, представляти його як структуру даних та обробляти, щоб уникнути перешкод під час планування шляху.

Оскільки робота-рука може рухатися в тривимірному просторі, вона повинна бути повною об'ємним виявленням перешкод. Зазвичай це робиться за допомогою бінокулярного або монокулярного 3D-сканування за допомогою SLAM [2; 3].

SLAM – одночасна локалізація та картографування – це підхід до обробки декількох кадрів та позицій датчиків для побудови об'ємної моделі середовища.

Візуальний шлем може виконуватися навіть за допомогою однієї (монокулярної) камери. Це дешево і просто. Глибина не доступна повністю з одного зображення; натомість нам потрібні декілька зображень, щоб відповідати їм і обчислити диспропорцію.

З двома камерами це ще простіше - ми маємо фіксовану відстань між камерами. Маючи невідповідність, ми можемо встановити відстань від камери для кожної точки і, виходячи з неї, побудувати 3D-зображення зображення для поточного кадру.

Камера глибини Intel RealSense D435 – це стерео рішення, що пропонує якісну глибину для різних застосувань. Це широке поле зору, яке ідеально підходить для таких програм, як робототехніка або доповнена та віртуальна реальність, де найважливішою частиною є бачення якомога більше. Глибинна камера Intel RealSense D435 має більш широке поле зору приблизно в полі зору 85°. Камера глибини D435 має глобальний затвор. Камери з рухомим затвором реєструють всі пікселі в сцені, швидко скануючи ліворуч або праворуч або вертикально. Зазвичай це відбувається для кількох кадрів, але дані зберігаються як один кадр. Глобальні камери затвора працюють по-різному, оскільки знімають всю сцену в одному кадрі, тому кожен піксель знімається одночасно.

Весь підхід SLAM передбачає створення повної карти. SLAM може використовувати декілька різних типів датчиків, а потужності та обмеження різних типів датчиків стали важливим рушієм нових алгоритмів. Статистична незалежність необхідна для того, щоб впоратися з метричним зміщенням та шумом у вимірах. Дані різних типів датчиків можуть оброблятися різними алгоритмами SLAM, підходи яких більш сумісні з датчиками [6].

У нашому випадку маємо SLAM з одометрією, лазери, а тепер і октри з хмари точок.

Програмна платформа. Для організації проекту була використана ROS – операційна система робота. Це великий фреймворк, навіть схожий на операційну систему, як її називали, до того ж це не операційна система. ROS також надає деякі вбудовані пакети для навігації, моделювання, локалізації тощо [1].

Для тестування та налагодження чого-небудь, не працюючи з реальним роботом, можна використовувати альтанку, яка імітує робота у віртуальному просторі.

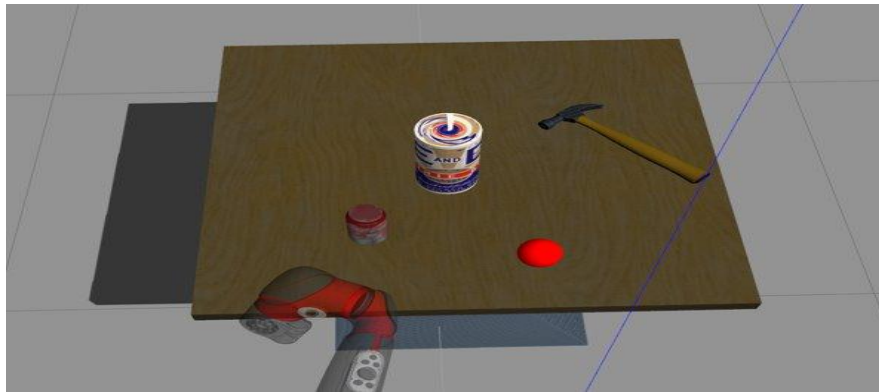


Рис. 3. Моделювання в Gazebo

Хмара точок. Після того, як робот отримав будь-які дані про глибину із зображення, він повинен зберігати ці дані. Одним із способів є хмара точок – сукупність точок у просторі. Коли камера (або вона генерується з кількома кадрами) робить глибинне зображення, кожен піксель глибинного зображення може бути перетворений у точку в просторі щодо положення камери. Тоді ці дані можуть бути додані до раніше створених хмар точок з попередніх кадрів.

Але якщо робот лише захоплює та додає дані, дані стануть більшими та більшими. Навіть коли камера бачить той самий об'єкт, вона додаватиме все більше і більше очок. Це непотрібний великий обсяг даних.

Вокселі та дерево октантів. Одним із способів дискретного представлення тривимірного середовища є вокселі. Воксель – це об'ємний піксель. В основному, воксель – це значення в звичайній сітці. Зазвичай значенням є прапор, якщо він зайнятий; крім того, він може містити будь-які дані, наприклад, колір. У типовому випадку, якщо ми маємо воксельну сітку, то воксель не представляє реальної позиції в тривимірному просторі; натомість він містить лише свою позицію. Такий підхід може бути ефективнішим, ніж звичайніша полігональна графіка, але іноді він може бути обчислювально дорожчим. Але це також можна оптимізувати – за допомогою дерева октантів. Дерево октантів – це деревоподібна структура даних, подібна до вокселів. Це можна розуміти як вокселі змінного розміру.

Ідея полягає в тому, щоб зробити один октант розміром сцени, а потім відокремлено розділити його на вісім октантів, якщо не весь простір всередині октанта зайнятий.

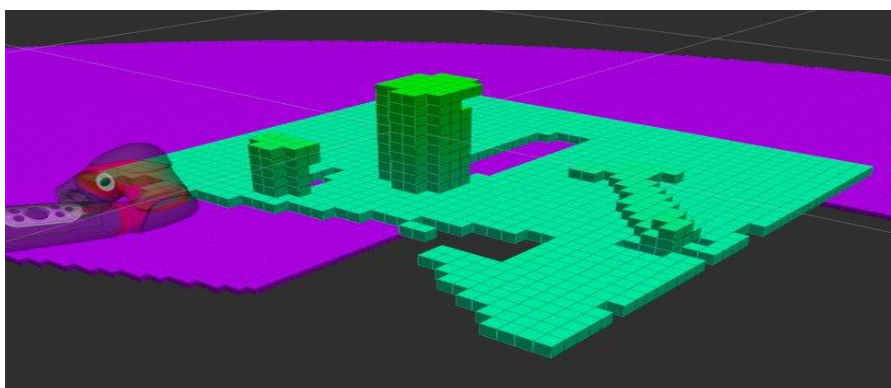


Рис. 4. Дерево октантів

Бібліотека OctoMap реалізує підхід 3D відображення сітки, забезпечуючи структури даних та алгоритми картографування на C ++, особливо підходящі для робототехніки. У цьому дослідженні я використовую OctoMap як частину MoveIt! рамки.

Виявлення пози для захоплення. Перш ніж схопити будь-який предмет, його слід впізнати та вибрати. Вирішити проблему в управлінні роботою робототехніки. Підходи до сприйняття можна розділити на такі шляхи.

– З відомими моделями. На основі розпізнавання заздалегідь визначених завантажених моделей. Метод намагається знайти подібні об'єкти серед наперед визначених моделей САПР. Це може бути точніше щодо відомих об'єктів, але зазвичай не працює, якщо об'єкт не розпізнається.

– Без відомих моделей. Коли система дозволяє знайти позицію захоплення об'єкта, навіть якщо це невідомий об'єкт, але в той же час, вона також може підтримувати виявлення відомих об'єктів.

Існує реалізація другого підходу – пакет виявлення пози (GPD). Він забезпечує 6-DOF пози для захоплення предметів. Це означає, що предмет можна захопити з будь-якої сторони з будь-якою орієнтацією, якщо рука може забезпечити таку позу. На даний момент він підтримує лише паралельні захоплювачі з двома пальцями, що підходить для захоплювача, який ми використовуємо. Крім того, захоплювач - це не паралельний щелепний захват, а ножичний тип [11].

В якості вхідних даних GPD використовує дані хмарних точок та виводить доступні пози для захоплення.

Основними сильними сторонами GPD є:

- роботи для нових об'єктів (для виявлення не потрібні моделі САПР),
- працює в щільній метушні, і,
- виводить 6-DOF пози для захоплення (дозволяючи не лише захоплення зверху вниз).

Управління рукою за допомогою MoveIt. MoveIt – це проста у використанні платформа для роботи з відкритим вихідним кодом для алгоритмів розробки комерційних додатків, створення прототипів та тестування [9]. MoveIt має інтерфейс плагіна для роботи з планувальниками руху. Це дозволяє MoveIt використовувати та взаємодіяти з різними планувальниками руху, які можуть надаватися багатьма бібліотеками. Це робить MoveIt легко розширюваним [8].

Для представлення фактичного світу навколо робота та збереження його та стану робота в рамках використовується сцена планування. Це підтримується монітором сцени планування всередині вузла групи переміщення. Монітор сцени планування підписується на:

- Інформація про державу: за темою спільних держав
- Інформація про датчик: використовуйте монітор світової геометрії, описаний нижче
- Інформація про геометрію світу: отримує дані від введення користувачем на тему планування сцена (як різниця сцени планування).

Монітор карти заповнення розробляється за допомогою Octomap для представлення карти заповнення навколишнього середовища. Octomap може кодувати ймовірнісну інформацію про окремі комірки, але ця інформація не використовується в поточній версії MoveIt. Вихідні дані Octomap як ostree можна безпосередньо передати у Бібліотеку гнучких зіткнень - бібліотеку перевірки зіткнень, яку використовує MoveIt.

Перевірка зіткнень у MoveIt налаштована всередині сцени планування за допомогою об'єкта CollisionWorld. На щастя, MoveIt налаштовано таким чином, що кінцевим користувачам ніколи не доведеться турбуватися про те, як внутрішня перевірка зіткнень. Перевірка зіткнень у MoveIt в основному обробляється за допомогою пакету FCL –основної бібліотеки перевірки зіткнень MoveIt.

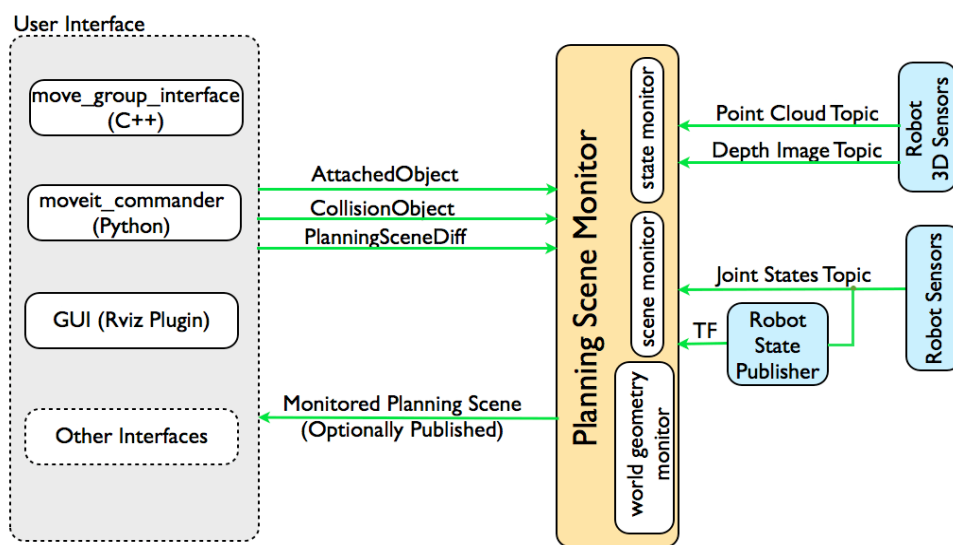


Рис. 5. Структура сцени планування

MoveIt підтримує перевірку зіткнень для різних типів об'єктів, включаючи:

- Полігональну сітку.
- Первісні форми – напр. паралелепіпед, циліндри, конуси, сфери та площини.
- Octomap – вікно, створене Octomap, може бути безпосередньо використано для перевірки зіткнень.

Виявлення ковзання. Найпростіший спосіб досягнення тактильного сприйняття - це використання датчика прискорення. Це може бути використано для реєстрації вібрацій на захоплюючому пристрої в напрямках X, Y та Z. Датчик прискорення повинен бути встановлений якомога далі перед ефектором, щоб уникнути згасання вібрації еластичним TPU.

Крім того, властивість демпфування може таким чином бути використано для зменшення вібрацій іззовні [7].

Інша можливість полягає у реєстрації зусиль на приєднувальному фланці захоплювача. Це також дозволяє реєструвати вібрації на захоплюючому пристрої в напрямках X, Y та Z. Для цього до фланця захоплювача приклеюють тензодатчики. Завдяки змінам опору вібрації, що діють на фланець, можуть

бути обчислені назад. Крім того, можна зробити висновки щодо ваги захопленого предмета. Еластичний TPU може настільки приглушити вібрацію на пальці, що вона зникає в результаті шуму.

Безпосереднє вимірювання захопленого об'єкта можливо за допомогою оптичного датчика миші або трекбола / трекпада, встановленого в поглибленні. Це дало б можливість виявити об'єкт, що вислизає безпосередньо. Об'їзд через трудомістку та обчислювальну оцінку може бути опущений.

Висновки. В результаті роботи було розроблено повне рішення. Це рішення забезпечує повний конвеєр для виявлення, планування руху та захоплення. Під час дослідження розглядалися існуючі алгоритми та рішення. Після повного аналізу проблеми та доступних рішень була представлена повна система, яка вирішує дану проблему.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Koubaa, Anis (2016) Robot Operating System (ROS) (Volume 1). Springer International Publishing Switzerland
2. Koubaa, Anis (2017) Robot Operating System (ROS) (Volume 2). Springer International Publishing Switzerland
3. Koubaa, Anis (2019) Robot Operating System (ROS) (Volume 3). Springer International Publishing Switzerland
4. Carol Fairchild, Dr. Thomas L. Harman (2017) ROS Robotics By Example - Second Edition: Learning. to control wheeled, limbed, and flying robots using ROS Kinetic Kame 2nd Revised edition Packt Publishing
5. Pfaff O., Simeonov S., Cirovic I& Stano, P. (2011) Application of finray effect approach for production process automation. Vienna, Austria: DAAAM International
6. С. О. Кошель, Ю. Ковалёв, О. П. Манойленко (2019) Проектування промислових роботів та маніпуляторів. Центр навчальної літератури
7. Н.Д., Платонов А.К., Соколов С.М., Трифонов О.В. (2017) Алгоритмы управления движением схвата манипулятора. Москва: ИПМ им. М.В.Келдыша РАН
8. Mohammad Safeea, Pedro Neto, Richard Bearee (2019) On-line collision avoidance for collaborative robot manipulators by adjusting off-line generated paths: An industrial use case. Robotics and Autonomous Systems, Elsevier, 119, pp.278-288. hal-02362167
9. Kivelä, Tuomo & Mattila, Jouni & Puura, Jussi & Launis, Sirpa. (2017). Redundant Robotic Manipulator Path Planning for Real-Time Obstacle and Self-Collision Avoidance. 10.1007/978-3-319-61276-8_24.
10. Agostino De Santis, Alin Albu-Schäffer, Christian Ott, Bruno Siciliano, and Gerd Hirzinger (2007) The skeleton algorithm for self-collision avoidance of a humanoid manipulator The skeleton algorithm for self-collision avoidance of a humanoid manipulator. 1 - 6. 10.1109/AIM.2007.4412606.
11. ten Pas, Andreasland Gualtieri, Marcusland Saenko, Kate and Platt, Robert1 (2017) Grasp Pose Detection in Point Clouds. The International Journal of Robotics Research, Vol 36, Issue 13-14, pp. 1455-1473.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ФРЕЙМВОРКА ПІДМІНИ ДАНИХ ДЛЯ ТЕСТУВАННЯ FLUTTER ДОДАТКІВ

Анотація. Описано процес збільшення швидкості виконання модульних тестів для мобільних Flutter додатків. Приріст швидкості здійснюється за рахунок використання фреймворка підміни даних у тих ділянках коду, що потребують додаткового часу на виконання.

Ключові слова: *тестування, Flutter, мобільні додатки, модульне тестування, mock фреймворк, фреймворк підміни даних.*

Вступ. Останні декілька років у мобільній розробці набуває популярності крос платформне рішення Flutter. Flutter – це набір інструментів користувальницького інтерфейсу для створення гарних, нативно компільованих застосунків для мобільних, веб та десктопних платформ з єдиною кодовою базою та використанням мови програмування Dart.

Під час розробки програмних систем та додатків необхідно переконатися, що створений програмний продукт працює належним чином. Цей процес називається контролем якості або просто тестуванням програмного забезпечення.

Одним із видів тестування є автоматизоване модульне тестування або юніт тестування (unit testing). Модульне тестування представляє собою метод тестування програмного забезпечення, який включає в себе тестування індивідуальних елементів вихідного коду - наборів одного або кількох програмних модулів разом із асоційованим набором контрольних даних, використовуваних та операційних процедур, з метою перевірки їх на придатність до використання.

Перевагою такого методу тестування є надзвичайно велика швидкість та можливість автоматизованого запуску об'ємної кількості тестів без участі людини. Інколи при юніт тестуванні виникає проблема зовнішньої залежності, через яку тест неможливо виконати у тестовому середовищі. Також деякі об'єкти мають нестабільну поведінку або займають багато часу при перевірці, через що значно зменшується швидкість виконання модульних тестів. Одним із способів вирішення цих проблем є використання мок-об'єктів (mock objects) за допомогою фреймворків підміни даних (mock frameworks), які імітують поведінку реальних об'єктів, необхідних при роботі тесту, але недоступні через обмеження тестових систем або таких об'єктів, що значно сповільнюють роботу тестів.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- проаналізувати особливості та переваги Flutter SDK;
- визначити головні цілі тестування програмних продуктів;

- виявити основні проблеми, що виникають під час модульних тестів;
- дослідити ефективність застосування фреймворка підміни даних у вирішенні проблем модульних тестів;
- на основі отриманих знань створити власний інструмент для вирішення однієї з проблем модульних тестів.

Основний зміст роботи. Під час роботи були використані наступні інструменти: Flutter SDK, фреймворк для підміни даних mockito, бібліотека для відправки HTTP запитів http.Client.

Для економії часу та ресурсів у процесі створення мобільних застосунків було розроблено крос платформні рішення, що дозволяють писати код одночасно під дві платформи. Одними із перших були рішення, засновані на JavaScript, такі як веб віджети (Web Views) та Реакт Нейтів (ReactNative). Це доволі непогані рішення, що дозволяють позбутися проблем, що виникають у ході крос платформної розробки. Але вони мають значний недолік, так званий JavaScript міст. Суть проблеми полягає у тому, що коли крос платформному коду потрібно спілкуватися з нативним кодом, йому потрібно перейти JavaScript міст, що в свою чергу потребує додаткового часу. Flutter SDK позбавлений такої проблеми, оскільки він використовує власний движок відрисовки, Skia, так що потреба у конвертації Dart коду у нативний відпадає.

Окрім відсутності моста JavaScript, Flutter має продуктивний інструмент відрисовки елементів користувальницького інтерфейсу, які називаються віджетами. При описанні програмістом користувальницького інтерфейсу застосунку використовуються віджети, які відповідають загальноприйнятим елементам взаємодії користувача з програмою, такі як кнопки, тестові поля вводу даних, списки, зображення та інші, які відповідно представлені елементами Button, TextField, ListView, Image. Для кастомізації віджетів у Flutter застосовується компонування замість спадкування. Самим простим прикладом є додавання відступу до кнопки та встановлення її тексту, для цього потрібно використати елемент `Padding(child: Button(child: Text("текст кнопки")))`. Хоча тут використано три віджета, `Padding`, `Button` та `Text`, вони є лише метаданими та при фактичній побудові користувальницького інтерфейсу це не створить додаткових витрат на відрисовку, оскільки віджети будуть перетворені в єдиний цілісний елемент. Flutter є чудовим інструментом, який бере на себе багато важкої роботи по оптимізації виконання коду та досягненні високої продуктивності. Яким би розумним фреймворк не був, він не може застерегти від помилок бізнес логіки. Для знаходження та вирішення таких проблем використовується процес тестування програмного продукту.

Головною метою тестування програмного продукту є зниження ризику виникнення проблем під час роботи релізної версії програми на пристрої кінцевого користувача до прийнятного рівня. Акцент робиться на "прийнятному рівні", оскільки неможливо знизити ризик виникнення проблем до нульової вірогідності. Спроба довести, що програмне забезпечення працює на сто відсотків належним не може бути невдалою, оскільки достатньо лише одного невдало пройденого тесту, щоб доказати зворотне. Мету, що ставить перед собою завдання довести, що програмне забезпечення не працює

належним чином, доволі легко досягти. В такому разі з'являється нескінченний процес пошуку та виправлення проблем, через який програмне забезпечення виходить на ринок із значним запізненням, або не виходить взагалі. Тому доволі важливо дотримуватися балансу у процесі тестування та розуміти, що тестування не ставить перед собою мету охопити весь функціонал застосунку та довести, що він вільний від помилок, а знижує ризик їх виникнення до прийняттого рівня.

Модульне тестування (unit testing) є доволі швидким да дозволяє перевіряти працездатність великої кількості функціоналу програмного забезпечення у лічені хвилини. Цей процес можна порівняти з компіляцією програмного забезпечення. При компіляції застосунку компілятор аналізує код, та перевіряє, чи задовольняє він правилам написання, таким як наявність точки з комою у кінці програмної конструкції, використання неіснуючої змінної, коректність опису класів та використання операторів та керуючих структур. Компілятори допомагають знайти помилки у використанні мови програмування, тим самим значно економлять час програмістів. У ході модульного тестування описуються схожі правила, які відносяться до бізнес логіки застосунку або його окремих модулів, дозволяючи швидко переконуватися у працездатності коду та у тому, що нові зміни у бізнес-логіці не порушують стару бізнес-логіку.

Деякі програмні модулі через свою природу утворюють складнощі під час модульного тестування, такі як:

- збільшення часу проходження тесту;
- нестабільність роботи, що призводить до нестабільних результатів тестів;
- неможливість тестування через відсутність необхідних залежностей у тестовому оточенні.

Прикладом уповільнення роботи та нестабільних результатів можуть бути мережеві запити, які не тільки потребують додаткового часу на з'єднання з мережею, але й значно залежать від якості Інтернет з'єднання. Чим нижче швидкість з'єднання, тим повільніше будуть виконуватися тести, що задіють таку роботу. Швидкість виконання простого HTTP запиту на отримання даних може коливатися від 200 мс до цілих 30 секунд в залежності від якості з'єднання. При поганому з'єднанні запит може взагалі завершитися невдало через перевищення максимально дозволеного часу очікування, приводячи до нестабільної роботи тесту. І нарешті модулі, що виконують взаємодію із локальним сховищем, яке зберігається на фізичному мобільному пристрої, взагалі неможливо протестувати в початковому вигляді, оскільки у тестовому середовищі таке сховище відсутнє.

Один із засобів вирішення даних проблем є використання фреймворків для підміни даних. Такі фреймворки надають можливість створення мок-об'єктів, або тестових двійників, які імітують поведінку реальних об'єктів, але насправді ними не являються. При цьому система вважає, що вона працює з реальними об'єктами. Наприклад можна імітувати звернення до бази даних, яка насправді відсутня у тестовому середовищі, або повертати заздалегідь підготовлену

відповідь від серверу, що позбавить затримки на комунікацію та дозволить досягти стабільності.

Існує п'ять видів тестових двійників: холості, заглушки, шпигуни, мок-об'єкти та фальшивки. Незважаючи на таку різноманітність, вони можуть бути розділені на два типи: моки і заглушки. Відмінність між даними двома категоріями полягає в наступному. Мок об'єкти допомагають емулювати і досліджувати вихідну поведінку та вихідні взаємодії. Під час таких взаємодій тестова система звертається до її залежностей для зміни їх стану. Заглушки ж допомагають емулювати вхідні взаємодії. Під час таких взаємодій тестована система звертається до своїх залежностей для отримання вхідних даних.

Інструментом для створення мок-об'єктів на Flutter є офіційний фреймворк `mockito`, який має можливості типового мок-фреймворку, а саме описання логіки взаємодії з об'єктом та верифікація виклику методів. Для створення мок-об'єкту інструментами `mockito` необхідно створити клас, що розширює клас `Mock` та реалізує інтерфейс необхідно класу – `class MockMyClass extends Mock implements MyClass {}`, та настроїти його за допомогою методів `when()` та `thenAnswer()`.

Метод `when()` приймає виклик методу, який необхідно підмінювати. Під час створення конфігурації можна також вказати правила аналізу параметрів методу. Найпростіший варіант - це просто передати сире значення у метод, звичайний рядок, на яке фреймворк буде орієнтуватися під час перевірки параметрів, наприклад `myMethod("my_value")`. Також існують спеціальні об'єкти - матчери, які описують правила співставлення об'єктів. За допомогою функції `argThat(matcher())` можна комбінувати різні правила співставлення параметрів, наприклад:

- `startsWith("123")` та `endsWith("123")` – параметр повинен починатися або завершуватися значенням "123";
- `isNull/NotNull` – повинен дорівнювати або не дорівнювати значенню `null`;
- `hasLength` – параметр повинен мати властивість `length`.
- Окрім встановлених матчерів, можна створювати свої, розширюючи клас `Matcher` та реалізуючи метод `matches()`.
- Після конфігурування методу `when()` потрібно задати відповідь за допомогою одного із наступних параметрів:
 - `thenReturn` – повертає передане у нього значення, використовується у більшості випадків;
 - `thenAnswer` – повертає значення, що обгорнуто в об'єкт `Future`. Використовується для асинхронних операцій;
 - `thenThrow` – ініціює виклик виключення. Потрібен для імітації помилок.

Для перевірки ефективності фреймворку для підміни даних `mockito` було обрано програмний модуль, який потребує комунікації з сервером через мережу Інтернет для отримання даних. Тестування було проведено при різних типах мереж, які відрізняються максимальною швидкістю та пропускнуою

спроможністю, тим самим імітуючи зміну якості з'єднання, такі як Wi-Fi, 2G, 3G, 4G та з використанням мок-об'єкту. Результати затраченого часу були отримані із взяттям середнього значення після п'яти запусків тесту для кожного типу мережі, та зображені на рисунку 1.

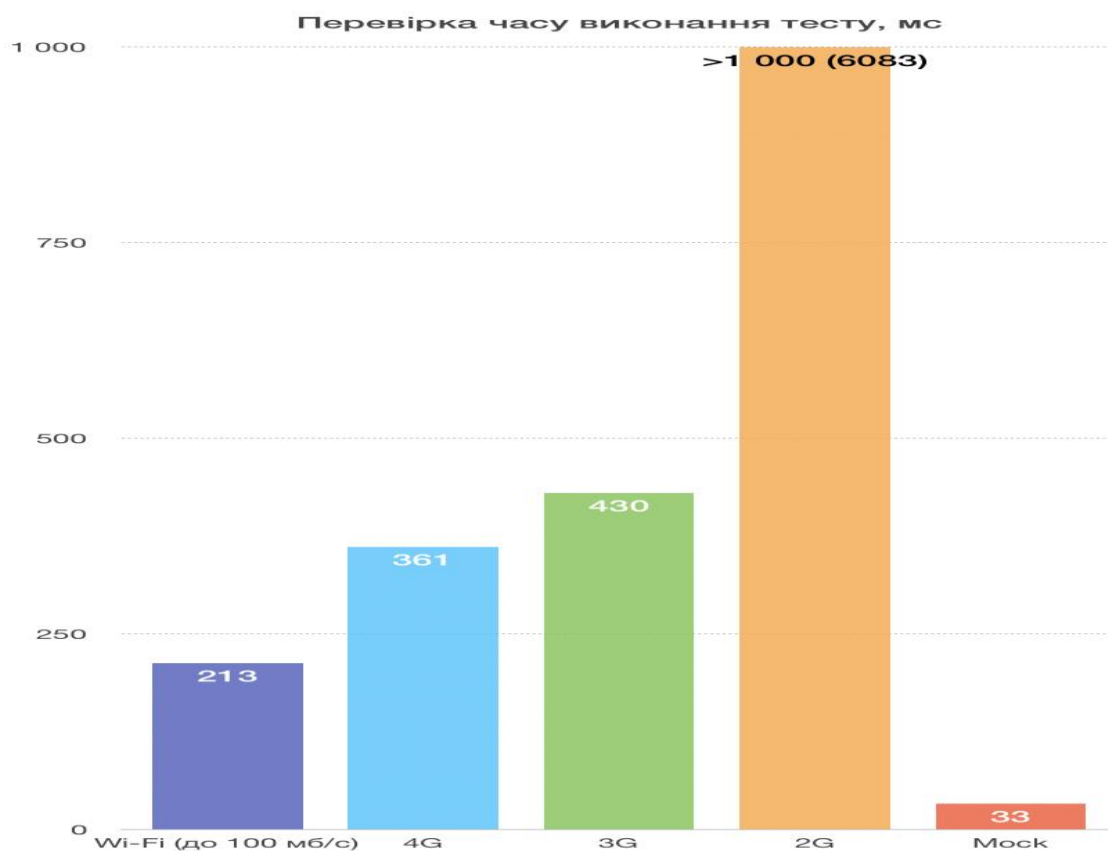


Рис. 1. Заміри затраченого часу на виконання модульного тесту з використанням різних типів мереж та мок-об'єкта

Середня швидкість виконання тесту склала: 213 мс для Wi-Fi мережі, 361 мс для мережі 4G, 430 мс для мережі 3G, 6083 мс для мережі 2G та лише 33 мс при використанні мок-об'єкту. З отриманих результатів видно, що використання мок-об'єкту дозволяє збільшити швидкість у 6,5 разів у порівнянні із Wi-Fi мережею та у 184 рази у порівнянні з мережею 2G.

В ході дослідження ефективності фреймворку для підміни даних `mockito` було виявлено, що головним вузьким горлечком у швидкості модульних тестів є виконання Інтернет запитів, тому було розроблено спеціальний вузькоспрямований інструмент, який дозволяє підмінювати дані для таких комунікацій. Даний інструмент має більш лаконічний декларативний синтаксис, який спрощує процес підміни даних для модулів, що мають комунікацію з мережею Інтернет, а також виконує справжні запити.

Даний інструмент дозволяє визначати логіку імітації роботи мережевих запитів для класу `NetworkApi`, що виконує запити. Він має методи для відправки `post` та `get` запитів та поле з конфігурацією `NetworkConfiguration`, яка реалізує відправку запитів. `NetworkConfiguration` є абстрактним класом що містить базові методи `sendGet()` та `sendPost()` але не реалізує їх.

HttpNetworkConfiguration розширює клас NetworkConfiguration та виконує HTTP запити за заданими параметрами. MockNetworkConfiguration також розширює базовий клас NetworkConfiguration, але замість відправки реальних запитів на сервер бере дані з об'єкту MockData.

На даний момент для конфігурації передбачено методи - response() та stubs(). Метод response() є доволі простим у використанні та приймає шлях і відповідь, яку необхідно повернути.

Метод stubs() дозволяє встановити більш широкую конфігурацію. Він також приймає параметр path, шлях, для якого потрібно встановити заглушки та параметр groupScenarioBuilder, який приймає список сценаріїв. На даний момент існує три типи відповідей:

- success – успішна відповідь, приймає json рядок;
- error – відповідь з помилкою, приймає об'єкт HttpExceptionWithStatus;
- successObject – такий же як success, але замість рядка приймає об'єкт.

Приклад використання зображено на рис. 2.

```
mock.stubs(  
  path: "users/1/update",  
  groupScenarioBuilder: GroupScenarioBuilder.scenarios([  
    Scenario.success(  
      when: (it) =>  
        it.bodyJson["id"] == 1 &&  
        it.bodyJson["name"] == "Taras Shevchenko",  
      response: """"{"status":"success"}""", // Scenario.success  
    Scenario.error(  
      when: (it) => it.header["Auth"] == "Auth Value",  
      exception: HttpExceptionWithStatus(  
        400, "CustomMessage bad request"), // HttpExceptionWithStatus, Scenario.error  
    Scenario.successObject(  
      when: (it) =>  
        it.query["type"] == 1 &&  
        it.query["action"] == "Update",  
      object: UserNetworkModel(  
        id: 3,  
        name: "Ivan Siabro",  
        username: "african_grey_parrot")) // UserNetworkModel, Scenario.successObject  
  ])); // GroupScenarioBuilder.scenarios
```

Рис. 2. Використання сценаріїв success, error та successObject.

Наукова новизна: удосконалено існуючі методи модульного тестування Flutter-додатків з використанням спеціального фреймворку для підміни даних; розроблено спеціальний інструмент для підміни даних мережевих запитів з лаконічним та декларативним синтаксисом.

Висновки. Було досліджено та збільшено швидкість виконання модульних тестів Flutter додатків за допомогою використання фреймворку для підміни даних. Завдяки знанням, що були отримані у ході дослідження, було розроблено спеціальний інструмент для підміни даних мережевих запитів з лаконічним та декларативним синтаксисом.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Dabit N. React Native in Action / N. Dabit – NY: Manning Publications, 2019. – 8 с.
2. Windmill E. Flutter in Action / E. Windmill – NY: Manning Publications, 2019. – 25 с.
3. Marco L. N. Beginning Flutter: A Hands On Guide to App Development / L. N. Marco – Birmingham: Wrox, 2019. – 77 с.
4. Beizer B. Software Testing Techniques, Second Edition / B. Beizer – NY: Van Nostrand Reinhold Co, 1990. – 19 с.
5. Khorikov V. Unit Testing: Principles, Practices, and Patterns / V.s Khorikov – NY: Manning Publications, 2019, – 105 с.

УДК 004.415.3:681.6

О.В. Іванченко¹, О.О. Рябічев²

¹Університет митної справи та фінансів, Дніпро, Україна

²Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ПРОЄКТУВАННЯ ХМАРНОГО АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ 3D-ДРУКУ

Анотація. Описано процес покращення користувацького досвіду при роботі з системами 3D-друку. Покращення досягається за рахунок будівництва найбільш ефективної архітектури системи із застосуванням математичного методу багатокритеріальних зважених оцінок.

Ключові слова: багатокритеріальні зважені оцінки, архітектура систем, 3D-друк, 3D-моделі, 3D-принтери, хмарні обчислення, web-додаток, *nodejs*, *docker*, *kubernetes*, *curaengine*.

Вступ. За останнє десятиліття спостерігається стійка тенденція з розвитку технологічних компаній в напрямку створення своєї закритої екосистеми, або ж апаратно-програмного комплексу. Як показує практика, подібний підхід дозволяє надавати користувачу найкращий користувацький досвід, що в цілому сприяє зростанню доходів компанії, та формуванню так званої "лояльності" з боку споживачів.

Так саме, як і з АПК, в останні декілька років можна спостерігати зріст популярності 3D-друку та впровадження його в абсолютно різні сфери життя людини – від друкування різного роду запчастин та прототипів до створення протезів та продуктів харчування. Цей зріст створює необхідність спрощення

взаємодії користувачів з пристроями для 3D-друку – 3D-принтерів. На жаль, наразі входження в сферу 3D-друку потребує від користувача глибоких знань та експертизи з проектування моделей, роботи з 3D-принтерами та їх обслуговування, підготовки моделей до друку та їх подальшої обробки, та іноді, навіть програмування. Наразі в сфері 3D-друку не існує готового комплексного рішення, що мало достатньо низький рівень входження для середньо статичного користувача 3D-принтеру.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- викласти принципи проектування та оптимізації АПК у сфері 3D-друку;
- визначити основні, мінімально необхідні компоненти системи;
- спроектувати апаратну частину АПК, шляхом аналізу напрацювань у цій сфері та вибору найбільш підходящих елементів;
- спроектувати та розробити програмну частину АПК, шляхом аналізу напрацювань у цій сфері та вибору найбільш підходящих технологій;
- спроектувати взаємодію ПЗ між собою та з АЗ;
- зробити висновки щодо доцільності створення подібної системи.

Основний зміст роботи. Під час роботи над даною роботою були використані наступні методи та інструменти.

- математичний метод багатокритеріальних зважених оцінок;
- математична модель обробки користувачьких команд прошивкою;
- TypeScript, NodeJS, CuraEngine, Three.js, Auth0, VSCode, Angular.

Апаратно-програмний комплекс – це сукупність апаратних та програмних засобів, що працюють сумісно для виконання однієї або декількох задач. Зазвичай, глибока інтеграція програмної та апаратної частини дозволяє досягти не тільки високого рівня продуктивності та надійності, а й найкращого користувачького досвіду від використання, як комплексу в цілому так й окремих його частин. В таких комплексах, виробники мають можливість забезпечити найбільш ефективну взаємодію частин систем завдяки тому, що вони мають контроль над кожною окремою частиною комплексу.

До подібних систем висуваються надзвичайно високі вимоги з точки зору надійності, продуктивності та зручності користування, як всією системою в цілому так і кожного окремого її елементу. Також, зазвичай, від таких систем вимагають унікального функціоналу при взаємодії з декількома елементами системи.

Успішне функціонування апаратно-програмного комплексу (АПК) залежить від багатьох факторів, одним з найважливіших є коректність роботи керуючого програмного забезпечення, яке, в свою чергу, визначається рівнем організації процесів в системі.

Наразі існує велика кількість стандартів, що регламентують всі етапи життєвого циклу АПК, що мають служити основою для створення надійних та продуктивних апаратно-програмних засобів. Під час створення таких комплексів виникають проблеми, обумовленні як специфікою проєктованих

систем так і особливостями процесів їх розробки. Розробка програмної та апаратної частин комплексу ведеться паралельно на усіх етапах життєвого циклу – від ескізного проектування до попередніх випробувань, та характеризується необхідністю постійних узгоджень, зазвичай суперечливих вимог, що висуваються до функціональності системи.

При проектуванні АПК, розробники мають враховувати вимоги до кожного елементу системи, а також забезпечити максимальну гучність у розширенні функціоналу та впровадженні нових елементів системи та засобів їх взаємодії. Особливо критично прогалини при початковому проектуванні системи дають про себе знати при додаванні нових елементів до існуючого АПК, адже кожен новий елемент системи може мати негативний ефект на конкретні елементи системи, або на роботу усїєї системи в цілому.

Щоб мінімізувати ризики при розширенні системи зазвичай використовують гнучкі принципи будування систем на модульній архітектурі. Модульні системи, як програмні так і апаратні визначаються високим рівнем гнучкості та масштабування.

Адитивні технології (3D-друк) – одна з форм технологій адитивного виробництва, де тривимірний об'єкт створюється шляхом накладання послідовних шарів матеріалу (друку, вирощування) за даними цифрової моделі. Друк здійснюється спеціальним пристроєм – 3D-принтером, який забезпечує створення фізичного об'єкта шляхом послідовного накладання пластичного матеріалу на основі віртуальної 3D-моделі. 3D-принтери, як правило, швидші, більш доступні та простіші у використанні, ніж інші технології адитивного виробництва. 3D-принтери пропонують розробникам продуктів можливість друку деталей і механізмів з декількох матеріалів та з різними механічними та фізичними властивостями за один процес складання.

3D-принтерами називають спеціальні станки з програмним керуванням, що будують деталь адитивним способом. Такі станки можуть працювати з різними матеріалами, від різноманітних видів пластику: PLA, ABS, PVA, Nylone, HIPS до пастоподібних продуктів харчування, таких як, сир, шоколад, паштети та інші.

Під час друку 3D-принтер зчитує файл у спеціальному форматі, зазвичай GCode, що містить набір примітивних послідовних команд, та наносить послідовність слоїв спеціального матеріалу, будуючи 3D-модель з серії поперечних перерізів. Ці слої, що відповідають віртуальним поперечним перерізам CAD-моделі, об'єднуються або сплавляються разом для створення об'єкту заданої форми. Основною перевагою такого підходу є можливість створення геометричних форм майже необмеженої складності.

Наразі існує широкий вибір методів адитивного виробництва. Основні відмінності полягають в методі нанесення слоїв та матеріалів, що можуть бути використані для друку. Деякі методи спираються на плавці або розм'ягченні матеріалів для створення слоїв, до них можна віднести вибіркоче лазерне спікання (SLS), вибіркоче лазерна плавка (SLM), пряме лазерне спікання матеріалів (DMLS), друк методом пошарового наплавлення (FDM або FFF).

Іншим вектором стало виробництво твердих моделей за рахунок полімеризації рідких матеріалів, більш відоме, як стереолитографія (SLA).

В цілому, всі 3D-принтери можна поділити на дві основні категорії: промислові станки та побутові прилади. Промислові станки використовують, зазвичай, на великих виробництвах при створенні високоточних прототипів та складних деталей. Розвиток промислового адитивного виробництва, останнім часом, йде високими темпами. Наприклад, сумісна американо-ізраїльська компанія Stratasys поставляє устаткування для адитивного виробництва вартістю від 2 000 до 500 000 доларів, а компанія General Electric використовує пристрої високого класу для виробництва частин газових труб. Побудові прилади, у свою чергу створюються з прицілом на середньостатистичного користувача, для вирішення його повсякденних задач. Розробка таких приладів ведеться приватними компаніями та ентузіастами, число яких постійно збільшується. Більша частини роботи виконується силами любителів для власних та суспільних потреб, за допомогою з боку академічного співтовариства та хакерів.

Серед побутових 3D-принтерів слід виділити дві основні категорії: виробничі принтери та харчові. Перші використовують різноманітні види пластику та металу – найбільш популярні серед інженерів та компаній, які в своїй роботі потребують розроблення прототипів (наприклад виробники побутової техніки, автомобільних запчастин, різноманітного устаткування та інші). В той час, як харчові 3D-принтери, виходячи з назви, використовують різні пастоподібні матеріали для створення харчових прикрас та десертів, що вражають уяву та приваблюють клієнтів. Зазвичай користувачі побутових 3D-принтерів не мають спеціалізованих знань у сферах 3D-друку, проектування, електроніці та хімії. Це накладає додаткові вимоги в області зручності та простоти використання таких приладів. Хоча за останні роки індустрія побутових 3D-принтерів і зробила великий крок на зустріч середньостатистичному користувачеві, нажаль, це все ще досить складне обладнання з не завжди зручним та зрозумілим ПЗ та зазвичай високим цінником, що, у свою чергу, відштовхує побутових користувачів та сповільнює темпи розвитку, популяризацію та, як наслідок, доступність обладнання для 3D-друку.

Наразі усі наявні на ринку системи 3D-друку, так чи інакше, мають наступні проблеми:

- відсутність повноцінного апаратно-програмного комплексу, який би відповідав усім сучасним вимогам а покривав усі автоматизовані стадії 3D друку;
- висока ціна на пристрої з повноцінним функціоналом. Проте досить просунуті користувачі можуть зібрати такий пристрій самостійно, використовуючи один з наявних на ринку пакетів для самостійної збірки;
- проте, найбільшою проблемою можна вважати надзвичайно високі вимоги до користувача для користування 3D-принтером.

Ці проблеми помітно сповільнюють швидкість розвитку та розповсюдження технологій 3D друку.

Для побудування надійної та швидкої системи для 3D-друку необхідно спроектувати апаратну та програмну частини.

Апаратною частиною в роботі виступатиме 3D-принтер. Існують такі основні компоненти 3D-принтеру, як: плата керування, плата обробки взаємодії з користувачем, екструдер, контролери руху, нагрівальна платформа, сенсорний дисплей. Звичайно, повноцінний 3D-принтер складається з набагато більшої кількості різних за важливістю елементів, проте ця робота фокусується лише на найбільш важливих компонентах. Екструдер, контролери руху, нагрівальна платформа та сенсорний дисплей є більш-менш стандартизованими на ринку. Проте, для зваженого вибору плати керування та плати обробки взаємодії з користувачем необхідно було проаналізувати наявні на ринку варіанти.

Для такого аналізу було вирішено використати математичний метод багатокритеріальних зважених оцінок.

Суть методу полягає в:

- формуванні критеріїв для вибору;
- підбору ваги для кожного з критеріїв;
- проставлені і нормалізації оцінок для кожної опції по кожному з критеріїв;
- розрахунку зваженої оцінки для кожної опції по кожному з критеріїв;
- розрахунку остаточної зваженої оцінки для кожної опції;
- вибору найкращої опції.

Вага кожного з критеріїв вираховується за формулою (1):

$$\sigma_i = \frac{r_i}{\sum_i^m r_i} \quad (1)$$

Зважена оцінка для кожної опції по кожному з критеріїв можна вирахувати за формулою (2):

$$K_{ij \text{ weighted}} = K_{ij} * \sigma_i \quad (2)$$

Остаточна оцінка для кожної з опції вираховується за формулою:

$$S_i = \sum_{j=i}^m K_{ij \text{ weighted}} \quad (3)$$

За результатами аналізу було вирішено використовувати плату Arduino Mega у якості плати керування та Raspberry Pi3 у якості плати для обробки взаємодії з користувачем.

Створення повноцінного апаратно-програмного комплексу для роботи з 3D-друком потребує розробки спеціального ПЗ, що покривало б усі потреби середньостатистичного користувача.

Мінімальний набір подібного ПЗ включає:

- прошивки – низькорівнева оболонка, що відповідає за взаємодію з контрольованими елементами 3D-принтеру;
- інтерфейс для локального управління принтером (з використання сенсорного дисплея);
- інтерфейс для віддаленого керування процесом друку;
- проксі-сервер для забезпечення інфраструктури для віддаленого підключення до 3D-принтера
- слайсер – спеціальне ПЗ для підготовки 3D-моделі до друку на конкретному 3D-принтері;
- хмарне сховище – для зберігання оригінальних та підготовлених моделей.

Подібний набір програмного забезпечення повинен повністю покрити вимоги середньостатистичного користувача до системи для 3D-друку. Математична модель обробки користувацьких команд прошивкою виглядає наступним чином (рис. 1).

X – множина вхідних параметрів (користувацькі команди у формі сигналів);

Y – множина вихідних параметрів (команди у формі сигналів, що відправляються для обробки на контрольовані компоненти принтера);

Z – множина збуджуючих ефектів середовища (різної роботи перешкоди для сигналу, втрата підключення та інші);

A – оператор математичної моделі, що є множиною алгоритмів та функцій.

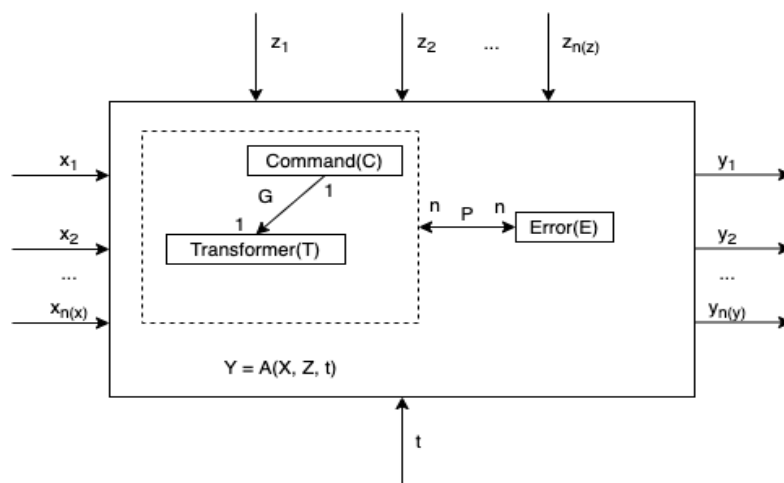


Рис. 1. Математична модель обробки користувацьких команд прошивкою

Нижче наведено основні терміни та визначення, що подані як сутності об'єктно-класифікаційної моделі та зв'язки між ними.

Команда (Comand) – команда у форми набору сигналі, отриманих від клієнта (у даному випадку Raspberry Pi). Множина команд складає:

$$C = \{c_1, c_2, \dots, c_{n(C)}\}, \quad (1)$$

де, $n(C) = |C|$ – потужність множини C , що відповідає кількості команд, що підтримується прошивкою.

Перетворювач (Transformer) – частини програмного коду прошивки, що відповідає за інтерпретацію отриманих команд у команди, що можуть бути відправлені до конкретного елементу принтера. Множина перетворювачів:

$$T = \{t_1, t_2, \dots, t_{n(T)}\}, \quad (2)$$

де, $n(T) = |T|$ - потужність множини T , що відповідає кількості трансформації, що підтримуються прошивкою.

Помилка (Error) – можливі помилки, що виникають в прошивці в процесі інтерпретації команди. Вони можуть бути викликані некоректними вхідними параметрам від клієнта, так само, як й зовнішніми збудниками. Множина помилок:

$$E = \{e_1, e, \dots, e_{n(E)}\}, \quad (3)$$

де, $(E) = |E|$ - потужність множини E , що відповідає кількості помилок, що можуть бути оброблені прошивкою

Зв'язки між класами об'єктно-класифікаційної моделі наступні: G – зв'язок між отриманою командою і перетворювачем, що повинен обробити її;

P – зв'язок між перетворенням та набор помилок, що можуть виникнути.

Наукова новизна: удосконалено математичну модель процесів обробки команд, що надходять на керуючу плату принтера; отримав подальший розвиток метод багатокритеріальних зважених оцінок для вибору найоптимальнішої архітектури системи.

Висновки. В результаті виконання роботи було спроектовану надійний та ефективний апаратно-програмний комплекс для 3D-друку. У тому числі, було обрано найбільш важливі елементи апаратного забезпечення з використанням методу багатокритеріальних зважених оцінок, а також удосконалено математичну модель процесів обробки команд, що надходять на керуючу плату 3D-принтера.

Перелік посилань

1. Edward A. Bender An Introduction to Mathematical Modeling (Dover Books on Computer Science) / Dover Publications (Educa Books) – Mineola, 200 – 272 с.
2. Bertier Luyt, Samuel N. Bernier, Tatiana Reinhard Design for 3D Printing: Scanning, Creating, Editing, Remixing, and Making in Three Dimensions / Make Community, LLC – New York, 2015. – 162 с.
3. Neil Rosenberg Designing 3D Printers: Essential Knowledge / 3D Hubs – Amsterdam, 2020 – 197 с.
4. Dr. Dan-Andrei Marinescu How to build a 3D Printer: DIY project: “EASY CoreXY 3D Printer Model 350” / LM Publishing House – Volendam, 2019 – 87 с.
5. William Shotts The Linux Command Line, 2nd Edition: A Complete Introduction / No Starch Press – San Francisco, California, 2019 – 502 с.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЗАСОБУ FLUTTER ПРИ РОЗРОБЦІ МОБІЛЬНИХ ДОДАТКІВ

Анотація. Описано процес ефективною розробки програмного забезпечення для мобільних пристроїв за допомогою засобу Flutter. Було розроблено бібліотеку для засобу Flutter, яка дозволяє досліджувати швидкодію роботи мобільних додатків під iOS та Android, розроблених засобами Flutter.

Ключові слова: мобільний додаток, фреймворк, Flutter, Android, iOS, React Native, Cordova, Xamarin, UI, візуальний інтерфейс, кросплатформа.

Вступ. В сучасному світі кількість мобільних девайсів, смартфонів та планшетів, перевищила кількість звичайних персональних комп'ютерів, і кожного року ця різниця зростає ще більше на користь мобільних девайсів.

На сьогодні існує дві основні мобільні операційні системи – Android та iOS, які займають ~74% та ~25% світового ринку відповідно.

Для бізнесу все важливіше надавати своїм клієнтам швидкий доступ до своїх послуг та сервісів за допомогою мобільних додатків. А деякі підприємства навіть стають mobile-first, де мобільні додатки стають основним інструментом взаємодії з клієнтами.

Тому мати свій мобільний додаток – вже частіше стає необхідністю для багатьох підприємств. Але розробляти та підтримувати одразу два незалежні один від іншого додатки під кожен популярну мобільну платформу двома різними командами розробників – часами буває занадто дорого та довго для бізнесу.

Для вирішення цієї проблеми існує перелік різноманітних кросплатформних рішень, на кшталт Cordova або React Native, але вони мають ряд вад, які роблять розробку ще більш складнішою, а досвід користування додатком – стає неприємним для користувача. Однією з причин цього є, наприклад, використання мосту з інтерпретатором JavaScript, або рендеринг браузерних сторінок всередині додатку.

Щоб вирішити цю проблему та мати єдину кодову базу для додатків на різних платформах, компанія Google випустила фреймворк Flutter, який сам відтворює на екрані кожний піксель інтерфейсу користувача без використання браузера або посередників з іншими віртуальними машинами.

Але розробникам бракує засобів за допомогою яких можна також дослідити ефективність застосування засобу Flutter для розробки мобільних додатків.

Мета і задачі дослідження. Метою роботи є вдосконалення та пришвидшення процесу розробки якісних мобільних додатків на замовлення бізнесу під популярні мобільні операційні системи за допомогою засобу Flutter.

Основний зміст роботи. Для досягнення мети дослідження необхідно розв'язати наступні задачі:

1. Здійснити аналіз існуючих методів та інструментів, які дозволяють займатися одночасною розробкою під Android та iOS.

2. Проаналізувати нове рішення від Google – фреймворк Flutter як засіб для розробки мобільних додатків.

3. Здійснити порівняння та побудувати структурні моделі усіх можливих рішень для розробки програмного забезпечення для мобільних операційних систем.

4. Розробити плагін для засобу Flutter для визначення ефективності програмного забезпечення для мобільних пристроїв.

5. Зробити висновки щодо найкращих методів та інструментів для розробки мобільного програмного забезпечення.

Було здійснено дослідження фреймворку Flutter та його порівняння з іншими засобами розробки кросплатформених додатків на мобільні пристрої.

На сьогодні існує декілька основних засобів для розробки мобільних додатків одразу на декілька платформ одночасно, використовуючи єдину кодову базу. Серед таких можна назвати React Native, Cordova, Xamarin та нову розробку в цьому напрямку – Flutter.

Більшість цих засобів використовують одну з двох технологій для реалізації роботи на багатьох платформах – відтворення вебсторінок за допомогою браузерного рушія, вбудованого у мобільний додаток, як це робить Cordova, або відтворення нативних візуальних елементів операційної системи за допомогою команд, які передаються операційній системі через спеціальний міст («bridge») з віртуальної машини іншої мови програмування, яка виконується у фоновому потоці разом з головним потоком мобільного додатка, як це працює, наприклад, у React Native.

Обидва ці способи мають цілий ряд вад, які нівелюють більшість переваг цих підходів. Так, більше усього при застосуванні цих засобів страждає швидкодія програмного забезпечення на кінцевих девайсах користувачів. Це провокує зниження рівня задоволення від використання мобільного додатку, зменшує час, який проводять в ньому користувачі та негативно впливає на оцінку у AppStore та Google Play.

Крім того, підтримувати мобільні додатки, які написані на фреймворках на кшталт React Native та Xamarin стає навіть складніше, ніж підтримувати рідні, нативні, додатки для мобільних платформ, що згодом заважає продовжувати швидко та якісну підтримку та оновлення програмного забезпечення.

Новий засіб у вигляді фреймворку Flutter був створений, щоб вирішити ці проблеми і тому, згідно з аналізом, він був побудований з урахуванням попередніх помилок інших засобів та створює дещо інший підхід до кросплатформеності.

А саме, замість використання браузерних рушіїв або мостів для відтворення візуального інтерфейсу, він використовує свій власний рушій для відтворення інтерфейсу, який компілюється у нативні бінарні бібліотеки arm/x86 під час побудови додатку.

Тобто, Flutter сам вимальовує кожен піксель на екрані та маніпулює з потоком вводу-виводу, без використання мостів, фонових потоків, браузерних рушіїв тощо.

Цей підхід дозволяє писати програмне забезпечення одразу на велику кількість платформ (на поточний час підтримується Android, iOS, Windows, Linux, macOS та створення вебсайтів) з однією кодовою базою, яка буде мати мінімальні залежності від платформи.

В теорії Flutter можна використовувати в більшості цифрової техніки, де є дисплей для відображення інтерфейсів. Так, наприклад, компанія Sony використовує Flutter у своїх embedded-розробках.

Разом з цим, Flutter використовує підходи декларативного написання візуальної частини інтерфейсу всередині програмного коду, що дозволяє вносити зміни та оновлювати візуальний інтерфейс додатків в легший спосіб, ніж це імплементовано у нативній розробці у Android та iOS. Ці платформи тільки зараз починають впроваджувати подібні декларативні підходи до побудови UI подібні до Flutter, за допомогою SwiftUI та Jetpack Compose.

Також був розроблений прототип спеціального плагіну для Flutter, який працює з Android і iOS, який допомагає досліджувати швидкодію програмного забезпечення для мобільних пристроїв, яке використовує засіб Flutter.

Використані методи та підходи можуть застосовуватися при розробці програмного забезпечення під мобільні пристрої на замовлення бізнесу, щоб розробка мобільних додатків проходила швидшими та ефективнішими методами. Бібліотеку можна використовувати в будь-яких проєктах, що використовують Flutter.

Наукова новизна полягає у тому, що вдосконалено методи визначення ефективності мобільних додатків з використанням засобу Flutter, отримали подальший розвиток процеси ефективної розробки мобільних додатків за допомогою засобу Flutter та декларативного підходу до побудови мобільних інтерфейсів.

Висновки. Було здійснено дослідження фреймворку Flutter і його аналіз та порівняння з іншими засобами розробки кросплатформенних додатків на мобільні пристрої. За проведеним аналізом можна зробити висновки, що технології та принципи, які використовує Flutter, є більш перспективними, надійними, зручними та швидкими до використання при розробці кросплатформенних мобільних додатків, ніж ті, що використовує його найближчий та найпопулярніший аналог – React Native.

Фреймворк Flutter випереджає React Native у розробці мобільних додатків за наступними параметрами:

- Продуктивність та швидкодія створеного програмного забезпечення.
- Простота проєктування та розробки користувацького інтерфейсу (UI).
- Час, потрібний для розробки продукту.
- Надійність та краща підтримка від розробників та товариством розробників.

- Поточний тренд зацікавленості у технології стрімко зростає та вже випередив React Native.

- Фреймворк має можливість компіляції під десктопні операційні системи (Windows, macOS, Linux) та збірку під вебдодаток, який відтворюється у браузері.

Також Flutter є доброю альтернативною більш дорожчою та довшою нативної розробки під кожен платформу окремо, якщо розроблюваний мобільний додаток не використовує 3D графіку, та не взаємодіє на низькому рівні з hardware-частиною мобільних пристроїв. Тобто, якщо додаток більше зав'язаний на взаємодію з віддаленим сервером.

В результаті роботи, було створено прототип плагіну для Flutter, який дозволяє визначати швидкодію розроблених кросплатформених мобільних додатків засобом Flutter під операційними системами iOS та Android. Використані методи та підходи можуть застосовуватися при розробці програмного забезпечення під мобільні пристрої на замовлення бізнесу, щоб розробка мобільних додатків проходила швидшими та ефективнішими методами. Бібліотеку можна використовувати в будь-яких проектах, що використовують Flutter.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Gabriel Peal (2018), *Sunsetting React Native*, Medium
2. Hidenori Matsubayashi (2020), *Graphical User Interface Using Flutter in Embedded Systems*, Embedded Linux Conference Europe.
3. Nate Ebel (2018), *React Native: A retrospective from the mobile-engineering team at Udacity*, Udacity Engineering.
4. Marco L. Napoli (2019), *Beginning Flutter: A Hands On Guide to App Development*, 4-10.
5. Barry Burd (2020), *Flutter For Dummies*, 25-28.

УДК 004.415.3:681.6

І.М. Удовик¹, В.С. Горобець, О.Б. Степанов¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

МЕТОДИ, АЛГОРИТМИ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ФРЕЙМВОРКІВ ЗГІДНО З ПОТРЕБАМИ ПРОЕКТУ

Анотація. Описано процес вибору веб технології за рахунок зіставлення характеристик проекту та фреймворків. Стратегія вибору базується на розрахованих показниках ефективності та статистичних даних тенденцій розвитку трьох актуальних фреймворків.

Ключові слова: *front-end, фреймворк, компонент, інтерфейс, бібліотека, стан, життєвий цикл, Virtual DOM, React, Angular, Vue, JavaScript, MVC, веб-додаток.*

Вступ. Значна кількість сучасних бізнес-процесів створення веб-додатків зосереджена на компонентно-орієнтованому підході, який прийшов з мобільної розробки і успішно увійшов у веб за допомогою таких компаній, як Facebook і Google. Вони створили реактивні фреймворки, які повністю змінили підхід до звичної веб-розробки, зробивши її більш продуктивною і полегшивши процес масштабування, що дозволяє створювати додатки будь-якої складності за допомогою веб технологій.

На сьогоднішній день проблема продуктивності реактивних фреймворків стоїть досить гостро, адже продуктивність впливає не тільки на відсоток відмов користувачів, але і на просування в пошукових системах, а рішення даної проблеми має бути комплексним, зберігаючи всі зручності компонентного підходу. Використання фреймворків дозволяє розробляти повністю динамічні додатки, де більше 90% компонентів – реактивні, тобто вони мають свій стан і працюють у фоновому режимі, чекаючи оновлення даних, щоб змінити інтерфейс, що дозволяє підвищити швидкодію веб-сторінки та зменшити час її завантаження.

Постановка задачі. Для досягнення поставленої мети були виконані наступні завдання:

- описати сутність поняття фреймворку;
- визначити принципи застосування фреймворків;
- розкрити можливості та області використання React;
- розкрити можливості та області використання Angular;
- розкрити можливості та області використання Vue;
- проаналізувати потреби проектів;
- дослідити ефективність фреймворків;

Основний зміст роботи. Фреймворк (англ. framework – структура, каркас) – сукупність рішень по архітектурі, структурі і способам об'єднання компонентів системи, які можуть бути застосовані для деякої безлічі однотипних завдань. Основною характеристикою фреймворка є так звана інверсія управління. Зазвичай фреймворк грає головну роль і викликає код самого додатка. Потік управління інвертується тут – фреймворк викликає додаток, а не навпаки.

Головним показником корисності фреймворка є те, що він допомагає розробнику підвищити продуктивність і якість коду. Сучасні технології часто надають інструменти для кодогенерації або готові каркаси типових додатків. Так само добре продуманий фреймворк задіє всі заходи безпеки, допомагаючи розробнику писати більш захищені додатки. Великою перевагою використання фреймворка у великих командах розробників є те, що він допомагає слідувати єдиним стандартам. У кожного фреймворка є загальноприйняті рішення типових задач, стиль написання коду і готові реалізації часто використовуваних інструментів. Це дозволяє зберегти єдиний стиль програмування у всіх частинах програми.

Серед важливих принципів використання фреймворків варто виділити наступні:

- простота – зведення до мінімуму кількості обмежень, які нав'язує фреймворк;
- монолітність – забезпечення підтримки повного стека завдань для розробки веб-додатків;
- послідовність – використання однакових підходів в дизайні архітектури, іменування, організації коду;
- очевидність – написання такого коду, що є документацією сам для себе і уникає незрозумілостей;
- угоди вище конфігурації – використання значень за замовчуванням, при цьому дозволяючи змінити їх через конфігурацію, що дозволяє зменшити кількість прийнятих рішень.

Серед існуючих на даний час фреймворків для розробки користувацького веб-інтерфейсу найбільше розповсюдження здобули React, Vue та Angular.

React.js – один з кращих і найбільш поширених на сьогоднішній день способів створити односторінковий додаток (Single Page Application). Це гнучкий і простий у використанні фреймворк, що використовує компонентний підхід, тобто увесь додаток має бути розділений на компоненти, які зазвичай є представленням конкретної візуальної сутності. React має вбудовані механізми для підвищення продуктивності (Virtual DOM), надає такі можливості як методи життєвого циклу компонентів та управління локальним станом.

В основі AngularJS полягає принцип, що декларативний стиль краще підходить для створення інтерфейсу користувача (User Interface), а імперативний для написання бізнес-логіки. AngularJS являє собою комплексний фреймворк. У стандартному вигляді він надає наступні можливості: все, що потрібно для створення CRUD-додатків: data binding, базові директиви для шаблонів, валідація форм, роутинг, deep linking, dependency injection, інструменти для взаємодії з серверними (RESTful) джерелами даних, засоби для модульного та end-to-end тестування, шаблон типової програми, що включає в себе структуру каталогів і тестові скрипти.

Vue.js є досить компактним, легким у використанні, проте відносно потужним засобом для створення інтерфейсу. Він є поступово адаптованим, що дозволяє організувати структуру індивідуально до потреб проекту та надає усі необхідні інструменти: вбудовані переходи, директиви, модульність, життєвий цикл компонентів.

Виділення критеріїв проекту допомагає заздалегідь оцінити можливі ризики. Тому для початку потрібно визначити основні фактори ризику для проекту. Це можна зробити різними способами: аналіз минулих аналогічних проектів, підготовка всієї проектної команди до загальної наради, на якій кожен озвучує ризики, які він бачить в проекті, і використання стандартних списків ризиків, як правило, наявних в кожному проекті (специфічних для галузі в цілому або для компанії).

Визначимо наступні показники: розмір і тип проекту; складність проекту; швидкість розробки; вартість спеціалістів; доступність спеціалістів; наявність готових рішень; наявність широкої підтримки спільноти; гнучкість рішень; вимоги до швидкодії.

Вибір технології виходячи з таких критеріїв вносить набагато більше об'єктивності і тим самим заощаджує час та гроші на розробку. Ці метрики можна умовно розділити на дві групи: для першої можна визначити фреймворк спираючись на статистичні дані щодо його використання, проте для другої необхідно провести практичний аналіз вихідного коду фреймворку.

Для тесту продуктивності використані мініфіковані версії вихідних кодів фреймворків, в той час як для оцінки якості і валідності коду використані версії для розробки.

Показники були обчислені за допомогою прикладного програмного забезпечення для статичного аналізу коду Scitools Understand 4.0, а також бібліотек plato і complex-report, що можуть використовуватись як окремо, так і бути інтегровані в проект для забезпечення постійного моніторингу ефективності.

Розрахуємо такі показники:

- Кількість рядків у кодї;
- Кількість виразів;
- Кількість рядків коментарів;
- Співвідношення кількості рядків коментарів та рядків коду;
- Цикломатична складність;
- Глибина вкладеності;
- Кількість попереджень;
- Середня важкість функцій згідно методу Холстеда.

Цикломатична складність – метрика програмного забезпечення, розроблена Томасом Мак Кабе, що використовується для оцінки складності програм. Обчислює кількість лінійно незалежних шляхів в алгоритмі роботи програми на основі її вихідних текстів. Зазвичай чим вище цей показник тим більший відсоток допущення помилки при виконанні і тим більша загальна складність додатку.

Середня важкість за методом Холстеда розраховується як:

$$D = \frac{\eta_1}{2} * \frac{N_2}{\eta_2} \quad (1)$$

де η_1 – кількість різноманітних операторів, η_2 – кількість різноманітних операндів, N_2 – загальна кількість операндів.

Результати розрахунку приведені у табл. 1

Розрахунок показників вихідного коду фреймворків

Показник програмного забезпечення	Angular	Vue	React
1	2	3	4
Кількість рядків у кодї	36595	11965	2333
Кількість виразів	14775	2655	875
Кількість рядків коментарів	8839	996	329
Співвідношення кількості рядків коментарів та рядків коду	0,24	0,08	0,14
Цикломатична складність	3544	1173	415
Глибина вкладеності	10	9	9
Кількість попереджень	19	176	100
Середня важкість функцій згідно методу Холстеда	1323	5333	4356

Для проведення тестування продуктивності розглянутих фреймворків необхідно визначитися з предметом оцінки. Найбільш коректним буде використання веб-додатків, що мають абсолютно ідентичну функціональність, але реалізованих за допомогою різних фреймворків і мінімальною кількістю сторонніх бібліотек - лише тих, без яких практичне застосування фреймворків неможливо [4]. Для цього було реалізовано додаток для управління списком справ на трьох досліджуваних фреймворках.

Результати тестування наведені на діаграмі швидкості завантаження фреймворків (рис. 1).

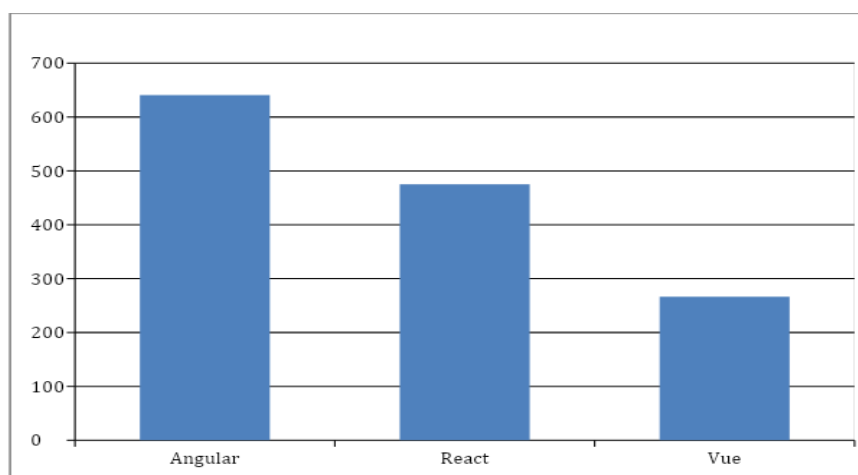


Рис. 1. Діаграма швидкості завантаження фреймворків (мс)

Наукова новизна полягає в тому, що отримали подальший розвиток методи ефективності використання фреймворків виходячи з потреб проекту.

Висновки. Був проведений практичний аналіз вихідного коду фреймворків за числовими показниками якості та ефективності з використанням методу

Холстеда та аналізом цикломатичної складності, зроблено аналіз швидкості завантаження фреймворків та виділено ключові потреби проектів.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Rakic G., Budimac Z. Problems in Systematic Application of Software Metrics and Possible Solution [Электронный ресурс]. – Режим доступу: <https://arxiv.org/ftp/arxiv/papers/1311/1311.3852.pdf>, своб.
2. Gizas A., Christodoulou S., Papatheodorou T. Comparative Evaluation of JavaScript Frameworks // Proceedings of the 21st Annual Conference on World Wide Web Companion. – 2012. – P. 513–514.
3. Barkmann H., Lincke R., Löwe W. Quantitative Evaluation of Software Quality Metrics in OpenSource Projects // Proceedings of International Conference on Advanced Information Networking and Applications. – 2009. – P. 1067–1072.
4. Graziotin D., Abrahamsson P. Making Sense out of a Jungle of JavaScript Frameworks, Towards a Practitioner-friendly Comparative Analysis // Lecture Notes in Computer Science. – 2014. – P. 334– 337.
5. Гуренко В. В., Бородин А. Ф., Назарков Д. А. Сравнительный анализ фреймворков для вебразработки. Технологии инженерных и информационных систем. 2017. № 2. С. 3–14.

УДК 004.415.3:681.6

В.В. Гнатушенко¹, І.О. Гненний¹, Д.К.Єфременко¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ДОСЛІДЖЕННЯ ВЗАЄМОДІЇ ФРЕЙМВОРКІВ ТА ПРОГРАМНОЇ ПЛАТФОРМИ NODE.JS З МЕТОЮ РОЗРОБКИ ВИСОКО ДИНАМІЧНИХ ВЕБЗАСТОСУНКІВ З ВИКОРИСТАННЯМ JAVASCRIPT

Анотація. Зібрано, систематизовано та досліджено інформацію про фреймворки та їх взаємодію з програмною платформою Node.js. Описано процес вибору технологій для ефективного використання веброзробниками можливих інструментів і шляхів розробки вебзастосунків, досягнення, шляхом інформування про новітні тенденції, фреймворки та методи їх використання, ефективності у виборі стеку технологій для виконання виробничих завдань. В ході створення кваліфікаційної роботи був розроблений вебзастосунок «SL1nk». Метою розробки є демонстрація результатів добре підібраних інструментів розробки програмного забезпечення на прикладі застосунку, що надає можливість великій кількості користувачів одночасно послуговуватися розробленим сервісом скорочення посилань з подальшою їх експлуатацією для будь-яких цілей і веденням аналітики переходів за посиланнями.

Ключові слова: *Node.js, фреймворк, клієнт, сервер, full-stack розробка, вебзастосунок, фронтенд, бекенд.*

Вступ. Веброзробка – один з найпростіших і тому популярних напрямків серед початківців програмістів. Для роботи у будь-якому текстовому редакторі і браузері не обов'язково вивчати алгоритми на високому рівні, результат кожного етапу написання програм очевидний. Для допомоги є величезний набір інструментів, і немає потреби на початку знати як все влаштовано. Достатньо розуміти як досягти результату за допомогою інструментів та конструкцій мови програмування, як то фреймворки, а вони, у свою чергу, дбають про те, щоб все працювало як слід. В усіх різновидах операційних систем, в усіх браузерах, на всіх видах пристроїв підтримується скриптова інтерпретована мова – Javascript. Згідно статистики вона використовується у 95% вебзастосунків. Знання Javascript вважається ключовою навичкою в контексті веброзробки. Програмний каркас або фреймворк (англ. software framework) – це готовий до використання комплекс програмних рішень, включаючи дизайн, логіку та базову функціональність системи або підсистеми. За наявності великого різноманіття систем з різними властивостями і потребами існує необхідність у зборі та аналізі інформації про наявні сучасні засоби розробки. Знання про існуючі фреймворки, бібліотеки та можливості їх взаємодії дає змогу правильно підібрати інструменти для розробки конкретних проєктів, що позитивно впливає на побудову злагодженої архітектури. Вона слугує програмістам у подальшій розробці, неодмінно зменшує вартість розроблюваних систем і їх подальшу підтримку, а також істотно знижує ризик необхідності перебудови системи при масштабуванні.

Постановка задачі. Для досягнення поставленої мети в роботі формуються та вирішуються наступні завдання: розглянути основні принципи використання мови Javascript у веброзробці; розглянути принципи побудови, властивості платформи Node.js та її застосування для розробки на стороні серверу; розглянути основні проблеми вибору технологій для виконання завдань веброзробки; зібрати, систематизувати та дослідити інформацію про фреймворки для Node.js; розробити вебресурс для організації обробки та скорочення URL посилань мовою програмування Javascript з метою демонстрації можливостей правильно підбраного набору інструментів.

Основний зміст роботи. Під час виконання даної роботи були розглянуті основні принципи вебпрограмування, та деякі проблеми пов'язані з розробкою у цій сфері. Початково жодної мови програмування всередині браузера не передбачалося. Виникнення легкої скриптової мови програмування, код якої можна було б писати прямо в тексті веб-сторінок було швидше схоже на детектив. Початково створений для більш вузьких задач, Javascript, завдяки еволюції веб в сторону інтерактивних вебзастосунків, отримав цілу інфраструктуру фреймворків, бібліотек, протоколів і методів. Термін “вебзастосунок” був створений, щоб стерти принципову відмінність між програмами, які передають користувачам, і програмами, які працюють на сервері. З'явилась платформа для виконання клієнтських і серверних застосунків – Node.js. Аналізуючи інформацію з відкритих джерел, ми зробили деякі попередні висновки про Node.js. – позитивні властивості:

- Власний інтерфейс на C++ для взаємодії з пристроями вводу-виводу комп'ютера. Тобто ця платформа фактично перетворює Javascript з вузькоспеціалізованої скриптової мови у мову загального призначення. І це означає, що Node.js підходить для побудови гібридних мобільних застосунків, комп'ютерних програм, ігор, вебсерверів, навіть для IoT (англ. Internet of Things – концепція мережі, що здійснює обмін даними між фізичними пристроями і комп'ютерними системами в автоматичному режимі, з використанням стандартних протоколів зв'язку).

- Застосунки на Node.js можуть виконуватися операційними системами Windows, Linux, MacOS та інших.

- Є full-stack ПЗ, адже дає можливість працювати як над клієнтськими (front-end), так і над серверними (back-end) частинами застосунків, використовуючи одну і ту ж саму мову програмування – Javascript.

- Відносна простота, швидкість та легкість написання програм.

- Можливість масштабування застосунків у всіх відношеннях. З Node.js легко обробляти тисячі синхронних запитів, потік даних відносно швидший і для обробки необхідно менше оперативної пам'яті.

- Node.js підходить для створення кроссплатформених застосунків. Такі технології дають змогу користувачу відправляти і отримувати повідомлення з телефону, веб версії застосунку та комп'ютерній програмі.

- Постійний розвиток завдяки рушію Google V8, що постійно розширює межі і є одним з найшвидших інтерпретаторів динамічних мов.

- Велика спільнота користувачів, що дає можливість отримувати вирішення проблемних питань, багато з них вже вирішені.

- Node package manager або npm – величезна кількість бібліотек які можуть бути встановлені написанням однієї строки в терміналі.

Також можна визначити очевидні мінуси використання Node.js:

- Швидкий розвиток призводить до необхідності постійно контролювати оновлення, щоб раніше розроблені застосунки продовжували працювати коректно. Наприклад, в минулому, один розробник видалив свою бібліотеку з npm і велика кількість застосунків, що користувалися бібліотекою перестали працювати.

- Підтримка об'ємних процесорних обчислень низька. Важкий запит на обчислення блокує обробку усіх інших завдань і уповільнить додаток, написаний на Node. Тому він не підходить для проєктів, що вимагають складних багатопоточних обчислень. Worker threads не можуть забезпечити координованих обчислень між потоками і потребують більше пам'яті, ніж звичайна багатопоточна програма.

- Node.js вимагає чіткої архітектури, ця платформа орієнтована на події, тому вона може запускати декілька подій одночасно, але тільки якщо стосунки між ними добре прописані.

- Документація Node.js недорозвинена, у ній занадто багато альтернатив, тому не завжди зрозуміло, що вибрати.

З часом інтерфейсні застосунки ставали все більш і більш складними, для задоволення цієї складності з'явилися інструменти – програмні каркаси, або фреймворки (англ. software framework) – це готові до використання комплекси програмних рішень, включаючи дизайн, логіку та базову функціональність системи або підсистеми. Фреймворк – це комплект налагодженого коду для вирішення конкретних завдань. Всі специфічні компоненти реалізуються розробником на їх основі. У фреймворку весь потік управління вже знаходиться на місці, в ньому розробники можуть розмістити свій власний код. В результаті фреймворк визначає скелет, в якому розробники конкретного застосунку визначають свої власні функції для заповнення цього самого скелета. Використовуючи фреймворки не потрібно писати наново десятки, а то і сотні тисяч рядків, тим самим досягається велика продуктивність. які найчастіше стоять перед розробниками.. Це не тільки скорочує витрати часу і грошей, але і дозволяє досягти більш високої стабільності рішення, адже компонент, що використовується і допрацьовується тисячами інших розробників зазвичай більш якісно реалізований і краще протестований на всіляких сценаріях, ніж рішення, яке може в адекватні терміни розробити один розробник або навіть невелика команда. Можлива відносно проста реалізація будь-якої бізнес логіки, а не тільки тої, що була закладена в систему спочатку. Фреймворки використовуються в різних областях залежно від їх основних характеристик. Оскільки існує безліч різновидів фреймворків і бібліотек – це сильно ускладнює процес вибору фреймворку для роботи окремим розробником.

Під час вибору фреймворку можуть виникнути певні труднощі, пов'язані з визначенням завдань, які він може виконувати, та його призначення. Якщо для створення сайту потрібно знайти зручний і простий в освоєнні фреймворк, то необхідно ретельно підійти до питання його вибору та зважити всі «за» і «проти». Адже неправильний вибір фреймворку потенційно може стати однією з основних причин невдачі проекту. Ми виділили основні критерії раціональності вибору та використання фреймворків:

- Підтримка баз даних – питання підтримки баз даних у фреймворку дуже важливе. Наприклад, частина фреймворків мають вбудований ODM/ORM шар, частина – ні. Залежно від бази даних, яка використовується для розробки проекту доводиться вибирати той чи інший фреймворк.

- Підтримка спільноти – при розробці застосунків, часто виникають моменти, коли розробник сам не може впоратись з вирішенням якоїсь проблеми, але, за наявності великої спільноти готових допомогти розробників, можна вирішити більшість проблем.

- Документація – деякі фреймворки не мають достатньої кількості документації або вона не є актуальною. Тому, перед вибором фреймворка, необхідно переконатися в тому, що документація актуальна і вчасно оновлюється, доповнюється, а інструкція із застосування проста для розуміння.

- Швидкодія – пропускна здатність системи – кількість операцій або опрацьованих запитів за одиницю часу на певному апаратному забезпеченні, або інша подібна вимірювана величина. Від швидкодії вебсистеми залежить: кількість користувачів, які одночасно можуть працювати з нею (наприклад,

кількість одночасних відвідувачів сайту), сатисфакція користувачів від взаємодії (прийнятність часу відповіді сайту на запити відвідувача суттєво впливає на те, чи користувач продовжить відвідування цього сайту і стане замовником, що платить, чи залишить його і, можливо, перейде до конкурентів та скористається їх послугами), як наслідок двох попередніх факторів – отримуваний від користувачів прибуток сайту, як через рекламну модель монетизації, так і прямий – від замовлень відвідувачів, що стали клієнтами.

- Коректність – задача гарантування коректності програмної системи завжди була і залишається складною і трудомісткою. У випадку використання сторонніх компонентів для її побудови (зокрема, фреймворків) задача стає практично нерозв'язною у класичному смислі (повністю) ще і з тієї причини, що частина системи (ті ж самі компоненти та фреймворки) стає «чорною скринєю», у яку немає способу «зазирнути» і стверджувати напевно будь-що відносно функціонування.

- Поріг освоєння – не всі фреймворки прості в освоєнні. Це дуже важливо враховувати при виборі, так як на освоєння одного фреймворка може не вистачити й року, а на освоєння іншого – вистачить всього тижня. У першому випадку зміни у технології можуть наступити раніше, ніж будуть якісно опановані попередні стабільні версії.

- Швидкість розробки – треба оцінювати в комплексі, тобто : час, необхідний для вивчення технології. Програмування з використанням технології. Відслідковування та застосування змін у технології (включаючи повторні тестування) – залежатиме від частоти та масштабності таких змін. Необхідність зміни у команді, що працює над проектом. В результаті такої комплексної оцінки може виявитись, що вибраний фреймворк не підходить, тому що вимагає значно більше часу на повний цикл розробки програмної системи.

- Архітектура – не захищають від помилок проектування архітектури програмної системи. Код, структурований розробником самостійно (скажімо, при класичному ООП), потребуватиме реорганізації, згідно визначених фреймворком правил. Це може зробити код менш цілісним і організованим складніше порівняно з тим же ООП.

- Швидкість розвитку – цей пункт так само дуже важливий, адже деякі фреймворки оновлюються раз на пару років, а деякі – раз на пару місяців. Це дозволяє уникнути використання старого, не доопрацьованого коду при розробці. Водночас з випуском нових версій можливі суттєві внесення коректив до частин розробленої системи, якщо нові версії не мають зворотної сумісності (тобто не підтримують старі API, що цілком можливо у сучасному світі IT)

- Шаблони для створення інтерфейсів – наявність даної функції значно спрощує створення інтерфейсу сайту, а іноді коли потрібен простий сайт то може звести написання інтерфейсу до мінімуму.

- Створення і перевірка форм – наявність даного функціоналу фреймворка полегшує створення полів для вводу тексту, логіну чи паролю до мінімуму. Полегшує створення валідації форм.

Визначившись з критеріями для оцінювання фреймворків, та ознайомившись із документацією можна зробити вибір. Але також важливо проаналізувати оцінку кінцевих користувачів. Це є ключовим фактором на ринку ІТ послуг. Навіть короткий відгук стане для розробника показником того, чи достатньо зручний і простий інтерфейс його проекту. Незалежно від складності процесів, головним є те, що бачить і відчуває користувач. Також важливо проаналізувати показник використання фреймворку, задоволеністю ним і зацікавленістю розробників. В даній роботі ми показали приклад такого аналізу та, користуючись результатами, виділили 3 групи фреймворків для подальшого поглибленого вивчення. Всього було проаналізовано 9 фреймворків:

1. Front-end фреймворки – React.js, Angular, Svelte.js.
2. Фреймворки для управління даними – GraphQL, Apollo.js, Redux.js.
3. Back-end фреймворки – Express.js, Next.js, Gatsby.js.

Наданий їх опис, специфіка та приклади використання у веброботці. Аналіз інформації про існуючі фреймворки, бібліотеки та можливості їх взаємодії дасть змогу правильно підібрати інструменти для розробки конкретних проєктів, вибрати оптимальний варіант для конкретного вебзастосунок з урахуванням поставлених задач.

В ході дослідження, в рамках кваліфікаційної роботи був розроблений вебзастосунок «SL1nk» («Еслінк» від словосполучення «shorten link» – скоротити посилання). Мета створення програми – демонстрація можливостей добре підібраних інструментів розробки ПЗ на прикладі застосунку, що надає можливість великій кількості користувачів одночасно послуговуватися розробленим сервісом скорочення посилань з подальшою їх експлуатацією для будь-яких цілей і веденням аналітики переходів за посиланнями. Такий сервіс і переваги скорочення посилань несуть користь як для окремих користувачів, так і для SMM, B2C, B2B та інших видів компаній інтернет бізнесу. «SL1nk» дозволяє користувачам створювати особисті кабінети, авторизуватись, використовувати інструмент створення скорочених посилань (рис. 1), переглядати та редагувати списки раніше утворених посилань з відображенням детальної інформації про кожне з них а також контролювати кількість переходів за кожним скороченим посиланням.

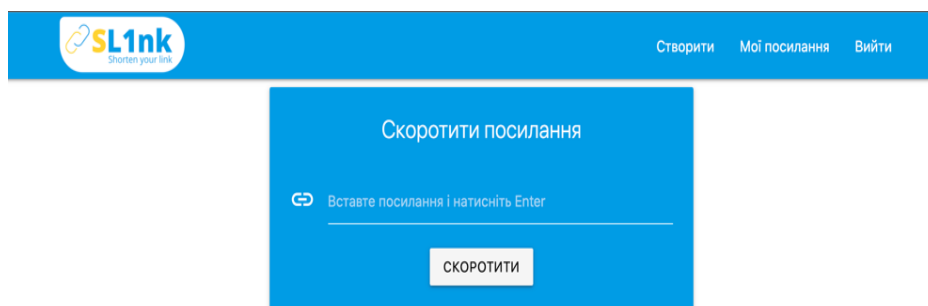


Рис. 1. Інтерфейс «SL1nk»

Крім того, користувачі не мають обмежень у кількості або унікальності таких посилань. Тобто, за потреби, можна створювати декілька коротких

посилань для одного і того самого оригінального посилання, що надає змогу контролювати кількість переходів за посиланнями з різних каналів публікування. Це може стати у нагоді для формування більш детального аналізу трафіку клієнтів з різних рекламних напрямків. Наприклад, одне посилання користувач може розмістити на фейсбук, друге на рекламну платформу, третє запустити в інстаграм чи твіттер і т.д. Таким чином користувач отримує можливість аналізувати – яка кількість відвідувачів приходить з цих каналів. Часто рекламні кампанії використовують декілька різних креативів для зацікавлення аудиторії. Власні короткі посилання для кожного з креативів, створені у розробленому сервісі, в тандемі з аналітикою публікацій у соціальних мережах, дозволяють аналізувати який з креативів отримав найбільший рівень конверсії, тобто кількість зацікавлених користувачів, що перейшли за посиланням по відношенню до загальної кількості переглядів публікації. Так, на одну вебадресу користувач «SL1nk» може отримати безліч скорочених посилань. Така можливість передбачена лише в деяких подібних сервісах і зазвичай не є безкоштовною. Переваги скорочених посилань. Зручні для сприйняття аудиторії. Часом посилання є настільки довгими, що їх публікація на тих чи інших платформах чи друкування на папері в такому вигляді виявляється неможливою. Ця користь скорочення посилань мабуть найбільш очевидна і не вимагає детального пояснення. Можливість відстеження трафіку з певного напрямку/публікації. Це відбувається, тому що кожна нова генерація скорочення дає новий URL, який, тим не менш, незмінно веде до заданої користувачем сторінки. Сервіс повністю пристосований для відображення на популярних пристроях.

Для розробки сервісу «SL1nk» були використані такі програмні каркаси, як: Node.js, React.js, Express.js а також деякі бібліотеки і пакети з npm. В якості бази даних для зберігання даних авторизації користувачів, списків їх створених посилань і кількості переходів виступає MongoDB. Таким чином розробку проведено за допомогою відомого набору інструментів розробки веборієнтованого програмного забезпечення – MERN (MongoDB, Express, React, Node). Цей набір технологій дозволяє розробникам веб застосунків писати код серверної і клієнтської частин ПЗ з використанням однієї МП – Javascript. Клієнтська частина застосунку написана на основі: компонентів, хуків стану, ефекту, контексту, маршрутизатора та інших можливостей програмного каркасу React.js, що дають змогу розробникам створювати вдало сконструйовані програми і використовувати прийоми декомпозиції коду, які суттєво покращують архітектуру складних, масштабованих програм, їх подальшу підтримку іншими розробниками та інтеграцію зі сторонніми системами. Віртуальна об'єктна модель документа, вбудована у React.js, зберігається в пам'яті і дозволяє застосунку працювати без перезавантаження всієї сторінки для оновлення елементів інтерфейсу. А маршрутизація і fetch запити дають змогу користувачам переключатись між сторінками застосунку без перезавантаження, контролюючи своє місцезнаходження в адресному полі браузера. Основна стилізація інтерфейсу програми виконана за допомогою бібліотеки Materialize.css, але в проєкті також присутня індивідуальна

стилізація з метою реалізації зручного відображення таблиць посилань та інших частин інтерфейсу на мобільних пристроях. MongoDB – це сервіс для створення і керування базами даних з відкритим вихідним кодом, що має зрозумілий інтерфейс і простий спосіб використання. Для взаємодії застосунку з БД використовувалась ODM бібліотека Mongoose, що надає рішення на основі об’єктів для моделювання схем БД. В ході розробки «SL1nk» за допомогою цієї бібліотеки були створені схеми User та Link, що моделюють збережену у MongoDB інформацію про користувача і скорочене посилання відповідно. Серверну частину вебзастосунку було написано на програмній платформі Node.js та каркасі Express.js, за допомогою можливостей маршрутизації і залучення проміжного ПЗ. Функції створення користувачів, створення посилань і додавання даних у БД, перевірки даних БД і маршрутизації запитів від скорочених посилань на оригінальні – вся ця логіка реалізована у маршрутизаторах. Для асинхронних запитів на сервер з клієнтської частини та отримання відповідей задіяний API Fetch. В «SL1nk» дані користувачів надійно захищені методами шифрування, використанням унікальних ключів, що звуть токенами, генеруються при авторизації і потрібні для ідентифікації користувачів. У реалізації системи захисту особистих даних використана бібліотека JSON Web Token. Коли користувач натискає «Вхід», сервер отримує запит з емейл адресою і паролем користувача, звіряє їх з інформацією у БД і якщо такий користувач існує в системі – сервер створює унікальний токен, в який закодовано інформацію про користувача, і відправляє його на сторону клієнта. Цей токен зберігається в локальному сховищі браузера, має обмежений час життя, що у «SL1nk» закінчується через годину після авторизації і слугує пропуском користувача в систему. Програма розкодує токен і звіряє його з даними кожного разу як здійснюється запит на сервер. Таким чином на стороні клієнта не зберігаються особисті дані як то пошта чи пароль. Для генерації скорочених посилань використовується бібліотека ShortID. З метою побудови гнучкого дизайну, для полегшення подальших модифікацій і створення підґрунтя для масштабування вебзастосунку використано шаблон проєктування і розробки ПЗ MVC. Модель життєвого циклу програми – каскадна. За методологією цього шаблону та прийомів декомпозиції код програми розділено на логічні структури – моделі, контролери і блоки представлення.

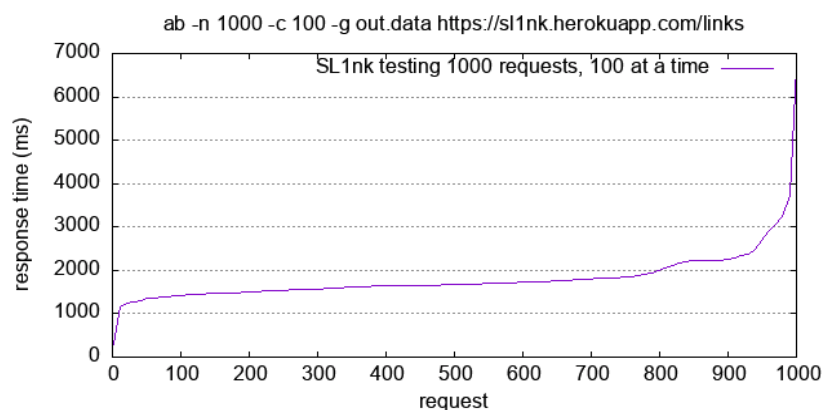


Рис. 2. Тест серверу

З метою вимірювання продуктивності серверної частини вебзастосунок було проведено декілька тестів за допомогою програми Apache Bench з різними вхідними параметрами. Перший тест на 100 почергових запитів показав, що сервер в середньому обробляв 3 – 3,8 запити. Другий тест на 100 запитів з 10 одночасних запитів пройшов успішно і показав обробку 27,45 запитів/секунду і загальний час обробки 100 запитів зменшився з 26,3 секунд у першому тесті до 3,64 секунди у другому. Це вказує на те, що серверу вдається в рази швидше обробляти об'єднані запити. У третьому тесті кількість запитів – 1000, а одночасних – 100. На графіку третього тестування видно що сервер зміг обробити 80% запитів доки швидкість обробки не почала стрімке падіння через завантаженість. Цього разу сервер відповідав із швидкістю 52,58 запити/секунду а загальний час обробки всіх запитів склав 19,02 секунди (рис.2). На графіку час відповіді сервера у мілісекундах відображено на вертикальній вісі координат, а кількість запитів – на горизонтальній.

І тут варто наголосити, що вебзастосунок був розміщений в мережі інтернет за допомогою безкоштовних планів MongoDB та Heroku.

Наукова новизна результатів кваліфікаційної роботи полягає в удосконаленні сучасних методів динамічної обробки URL посилань з використанням мови програмування Javascript на платформі Node.js.

Висновки. Представлена у роботі інформація і проведене дослідження, можуть бути корисними для початківців, студентів, аспірантів та всіх, хто цікавиться веброзробкою. Вони допоможуть сформуванню початкового розуміння області і теоретичні знання технологій. А розроблений вебзастосунок – слугуватиме практичним прикладом.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Artetxe M., Labaka G., Agirre E. (2019). An Effective Approach to Unsupervised Machine Translation / M. Artetxe, G. Labaka, E. Agirre // ArXiv. – Vol. 1902 – 2019 – P. 10.

2. Bahdanau D., Cho K., Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate. / D. Bahdanau, K. Cho, Y. Bengio// ArXiv. – Vol. 1409 – 2014 – P. 15.

3. Jurafsky D. Speech and Language Processing. An Introduction to Natural Language Processing. Computational Linguistics, and Speech Recognition 3rd edition draft/ D. Jurafsky, J. H. Martin - Stanford University - Stanford University Press, 2019 – 621 P.

4. Kavita G., Subotin M. (2015). A general supervised approach to segmentation of clinical texts. Proceedings. / G. Kavita, M. Subotin // 2014 IEEE International Conference on Big Data, IEEE Big Data 2014 – P. 33 – 40.

5. Ng A. Y., and Jordan M. I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. / A. Y. Ng, M. I. Jordan // Conference on Neural Information Processing Systems – Vol. 14 - 841–848 P.

С.Д. Приходченко¹, О.С. Шевцова¹, М.О. Нугуманов¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ПРОСЛУХОВУВАННЯ ТЕЛЕФОНІВ ДОДАТКАМИ З МЕТОЮ ТАРГЕТУВАННЯ РЕКЛАМИ

Анотація. Описано процес прослуховування телефонів з метою таргетування різних видів реклами для забезпечення конфіденційності.

Ключові слова: прослуховування, таргетування, реклама, мікрофон, користувач мобільні телефони, девайс, Internet.

Вступ. Із розвитком комп'ютерних технологій мобільні телефони стали невід'ємною частиною життя для переважної більшості людей. Ці девайси знаходяться поруч із власником майже цілодобово. Мобільні телефони володіють можливостями персональних комп'ютерів, але мають значно більшу розповсюдженість. У мобільному девайсі, завдяки мережі Internet, користувачі мають доступ до величезної кількості інформації та можуть користуватись різноманітними online послугами.

Нажаль, є й негативна сторона цих інновацій. Під загрозою опиняються особисті данні власників мобільних девайсів. Багато користувачів мобільних телефонів помічали, що річ, згадана в приватній розмові, з'являється в рекламі в браузері, або в соціальних мережах. Після того як люди обговорюють деяку річ, припустимо телевізор, по телефону, чудодійним чином в браузері з'являється реклама телевізорів на будь-який смак. Такі збіги змушують будь-кого замислитись про конфіденційність особистого життя. Ця проблема є актуальною для багатьох розробників й потребує вирішення.

Основний матеріал. Завантажені на телефон додатки часто мають велику кількість прав, які користувач дав машинально натиснувши «так» на всі спливаючі вікна. Чимало людей не замислюються про те якими правами володіють ті чи інші додатки в їх гаджетах. Дозволивши деякому додатку використовувати мікрофон, користувач не зазначає за яких обставин він може цим правом скористатись. Отже, мобільний застосунок має право користуватись мікрофоном будь-коли, навіть, під час телефонних дзвінків з метою прослуховування. Мова не йде про прослуховування і запис телефонних розмов вагою в гігабайти пам'яті, в даному випадку використовуються скрипти, що реагують на ключові слова: додаток, фіксуючи тригер, відправляє сигнал на необхідний сервер. Таким чином рекламні агенти отримують базу для таргетування. Прослуховування за ключовими словами – річ не нова, подібні методи, також, використовуються в таких проектах як американська система «Ешелон» [1]. Тільки в даному випадку зібрана інформація використовується не для запобігання терактам, а для підбору цільової аудиторії комерційних організацій. Також прослуховування можливе не під час телефонного дзвінка. Таке прослуховування використовується для того щоб підрахувати скільки разів у присутності людини прогала та чи інша реклама. Таким чином

рекламодавець може оцінити обсяг населення, яке побачило його рекламний ролик. Необхідною умовою є те щоб телефон лежав поруч з власником під час перегляду телевізора. (Якщо мова йде про телевізійну рекламу) Програма порівнює почуте із базою рекламних роликів й шукає спільні моменти. Проблема потребує вирішення, так як подібні методи прослуховування в руках зловмисників можуть використовуватися для заподіяння шкоди як морального так і фізичного характеру. До того ж, те, що сторонні компанії мають відомості про людей без їх згоди, нехай і з метою запропонувати холодильник, неправильно і є порушенням прав людини.

Деякі виробники вже зробили перші кроки до безпеки особистої інформації своїх покупців. В цьому році на своїй щорічній Всесвітній конференції розробників Apple оголосила, що планує впровадити мітки конфіденційності до додатків в своєму App Store. Отже, в 2021 році розробники повинні будуть прямо попросити користувачів iOS включити процеси збору даних. Ця функція повинна була з'явитися в iOS 14 на початку цього року, але Apple відклала реалізацію, щоб дати розробникам більше часу на підготовку [2]. На цю новину дуже бурхливо відреагував Facebook, розмістивши оголошення в національній газеті, в ньому стверджується, що малі підприємства постраждають, оскільки вони активно використовують платформу цільової реклами Facebook [3]. Поки що компанія Apple додала до iOS 14 тільки індикатор, який спалахує зеленим якщо включена камера, жовтим якщо включений мікрофон (рис. 1).



Рис. 1. Індикатор на iPhone, що з'явився в iOS 14

Компанія Apple має успіхи в цій галузі, але користувачі Android залишаються беззахисними перед крадіями особистої інформації. Можна вирішити ситуацію, дозволяючи додаткам користуватися мікрофоном та камерою тільки певний час. В деяких телефонах така можливість є, але для доступу до місця знаходження. Нажаль, для мікрофона та камери такої можливості немає (див. рис. 2).

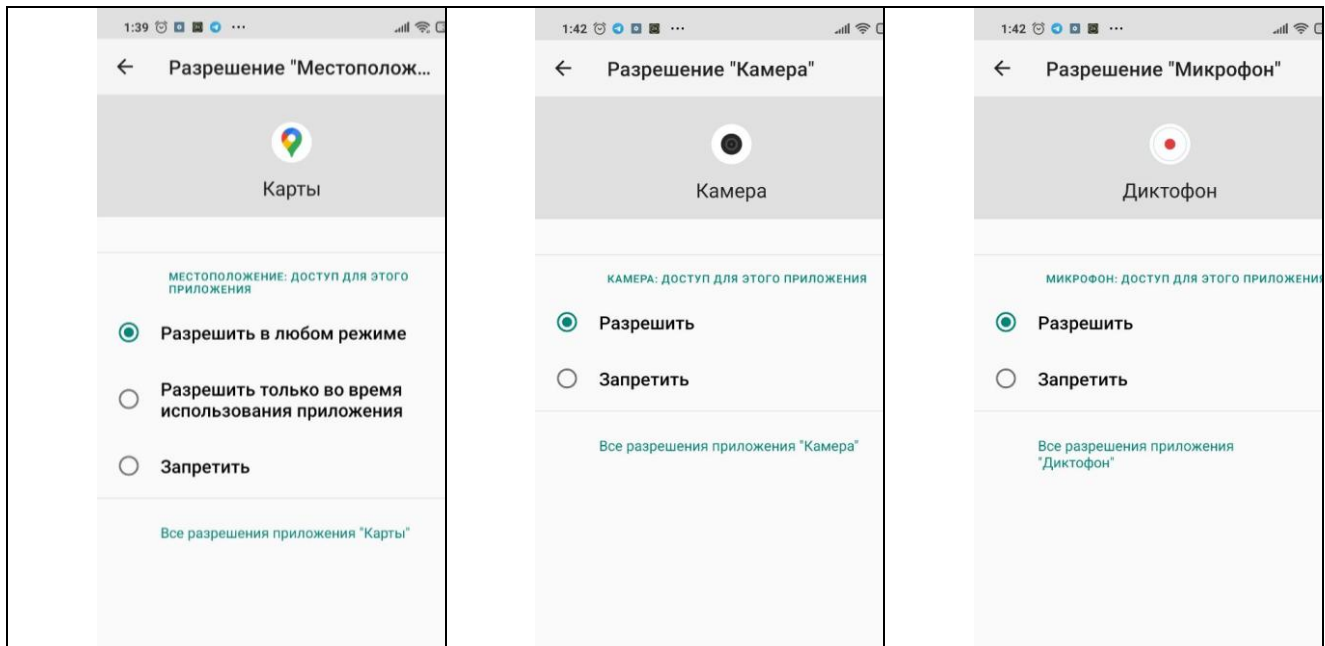


Рис. 2. Налаштування конфіденційності у нових телефонах на ОС Android.

Ця проблема може бути вирішена на правовому рівні. Наприклад, можна ввести закон, який зобов'язує додатки використовувати мікрофон і камеру тільки в певних, зазначених в заздалегідь заявленій політиці конфіденційності, випадках, а не коли завгодно. Програмісти можуть запропонувати користувачеві програму, яка буде мати доступ до мікрофона і камери на рівень вище ніж всі інші програми і контролювати коли і навіщо останні використовують особисті дані. Рішень проблеми може бути багато та її, безумовно, не можна залишати без уваги. Подальші дослідження покажуть який з варіантів рішення є найбільш вдалим.

Висновок. Проблема конфіденційності є гострою в сучасному світі так як, майже, всі люди зв'язані єдиною мережею Internet, в якій кожен залишає свій цифровий слід. Internet дає необмежену свободу пізнання і спілкування, але при цьому загальною прозорістю ставить під загрозу конфіденційність. Прослуховування, з появою суперкомп'ютерів, може досягти колосальних масштабів. Ця проблема, безумовно, потребує вирішення.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Kurt Kleiner. Trade Secrets: Is the U.S.'s most advanced surveillance system feeding economic intelligence to American businesses? URL: <https://fas.org/irp/program/process/991101-echelon-mj.htm> (дата звернення: 10.12.2020)
2. Stan Horaczek. Check these new privacy labels before you download another iPhone app. URL: <https://www.popsci.com/story/technology/apple-app-store-privacy-labels/> (дата звернення: 10.12.2020)
3. Tom Warren. Facebook criticizes Apple's iOS privacy changes with full-page newspaper ads. URL: <https://www.theverge.com/2020/12/16/22178068/facebook-apple-newspaper-ads-ios-privacy-changes> (дата звернення: 10.12.2020)

Н.І. Рябокони¹, К.С. Родна¹, М.О. Нугуманов¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

СТВОРЕННЯ ПРОГРАМИ ДЛЯ ШВИДКОГО ФОРМУВАННЯ ЗВІТІВ

Анотація. Описано процес створення програмного застосунку, що має можливість сформувавши якісний звіт з виконаної студентом роботи і є актуальною проблемою для розробників, що має необхідний функціонал, і матиме попит серед користувачів.

Ключові слова: *дистанційна освіта, мобільний застосунок Report and Run, необхідний функціонал можливостей.*

Вступ. У зв'язку з пандемією тенденція до цифровізації навчання значно посилилась. Дистанційна освіта замінила традиційну як в школах так і в закладах вищої освіти. Не дивлячись на те, що до пандемії з появою новітніх технологій в освіті вже відбулися значні трансформації, до таких принципів змін не були готові як викладачі так і студенти. Разом із деякими позитивними явищами, що є наслідками дистанційної освіти, на жаль, є велика кількість незручностей, які заважають навчанню. Цифровізацією навчання стурбовані цілі нації та держави. У вітчизняній літературі неодноразово розглядалися проблеми становлення та розвитку дистанційної освіти в Україні [1, 2]. Однією з проблем, яка постала перед студентами під час дистанційного навчання є звіти по виконаній роботі. Вирішенням може стати створення спеціалізованої програми що формує звіт з виконаної на папері роботи. Такі електронні комплекси називаються генераторами звітів. Генератор звітів – програма або бібліотека, що дозволяє представити інформацію в зрозумілому людині структурованому вигляді. Іншими словами перетворити дані на інформацію (документ, звіт), яку можна роздрукувати або зберегти в різних електронних форматах. Ця проблема є актуальною для розробників програмного забезпечення й потребує вирішення.

Постановка завдання. Метою пропонованої статті є дослідження теоретичних і практичних засад створення системи звітності робот студентів в умовах пандемії та дистанційної освіти. У процесі досягнення мети вирішені такі основні завдання:

– з'ясовано передумови ефективного навчання учнями в умовах створених пандемією в системі дистанційного навчання, місце та значення інформації, зокрема, системи звітності в процесі навчання;

– обґрунтовані вимоги до побудови системи звітності з метою ефективного використання в навчальному процесі.

Основний матеріал. Значною проблемою дистанційного навчання є звіти по виконаній роботі, які раніше формувалися іншим чином: робота здавалася на папері. Під час дистанційного навчання кожен учень має відправляти електронні звіти по різноманітних предметах в незалежності від того

контрольна це або домашнє завдання. Учні зобов'язані робити звіти у форматі PDF або Word, процес складання яких займає багато часу. Значною проблемою таких звітів є те що вони часто мають неякісний вигляд: у звітах з'їжджають фотографії, плутається порядок сторінок роботи тощо.

Процес складення звітів на даний момент є незручним для студентів. Для створення звіту спочатку необхідно зробити фото на телефон, потім відправити його на комп'ютер, щоб сформувати Word документ. Також є можливість скористатися мобільним додатком Microsoft Word, що є менш зручним за використанням. У програмі Microsoft Word досить багато часу йде на те, щоб розмістити фотографії у правильному положенні. Програма Microsoft Word не має спеціальних алгоритмів для швидкого розміщення фото на сторінках. У результаті на формування звіту йде час, який міг би витратитися на саморозвиток або відпочинок. Ця проблема досі не розв'язана. Немає такого додатку, яке могло б відповідати вимогам, що задовольняли б потреби учнів та вчителів. Найбільш оптимальним рішенням поданої вище проблеми було б створення мобільного додатку для формування звітів.

Зараз у додатку PlayMarket є можливість встановити програми для формування звітів, наприклад, найпопулярніший за рейтингом PlayMarket мобільний застосунок Report and Run. Проблемою є те, що ці додатки створенні не для дистанційного навчання. Мобільний додаток Report and Run дає змогу створювати та відправляти звіти в форматі PDF з фотографіями, текстовими полями та малюнками, але він не є зручним для формування звітів, що використовуються при дистанційній освіті. Перш за все цей інструмент був розроблений у 2016 році, щоб полегшити роботу менеджерів по будівництву для створення і відправки професійних фотографічних звітів з поля, тому у цьому мобільному додатку є багато функцій, які не потрібні студентам та школярам, вони ускладнюють формування звіту. Отже, Report and Run не оптимізовано для навчання: додаток є не зручним для нього. Тому ідея розроблення мобільного додатку для формування звітів під час дистанційної освіти є актуальною.

До функціональних можливостей мобільного застосунку мають входити: можливість зробити фото або завантажити фото з пам'яті мобільного девайсу, можливість редагувати фото, можливість додати малюнок та додатковий текст, а також можливість створити заголовок документа. Додаток має розпізнавати фото і, керуючись заданими алгоритмами, розміщувати картинки по листах документу. Після розміщення фотографій та оформлення звіту згідно до вимог файл має бути збережений у форматі PDF або Word (див. рис. 1). Таке застосування значно полегшить життя школярам та студентам. Більш того якісні і структуровані звіти набагато легше перевіряти викладачам, що у свою чергу полегшить їм роботу.

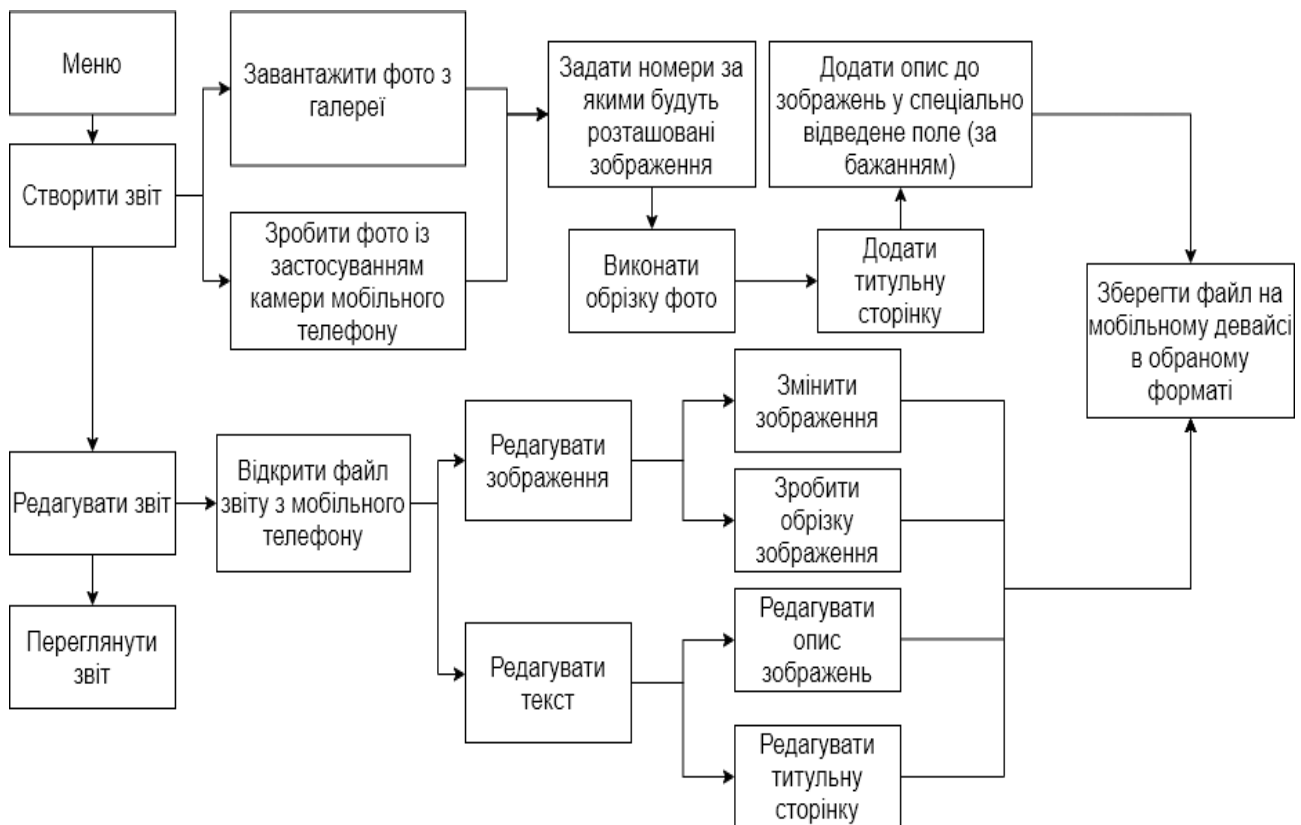


Рис. 1. Схема, що відображає функціональні можливості додатку.

У зв'язку з ускладненням форм і кількості документів особливо гостро постає завдання створення електронних комплексів, що дозволяють автоматизувати процес «виробництва» документів, використовуючи різноманітну інформацію. Програми для створення різного ступеня складності звітів актуальні в світі і можуть бути створені різними засобами [3]. Розробниками розглядаються сучасні методи введення структурованих документів шляхом заповнення електронних форм [4]. Розробка програм для формування графічних звітів є актуальною, тому що будь-яка система обліку і опису об'єктів виграє від додавання до описів зображень. Навіть найдосконаліший опис може поступатися в інформативності зображенню. Створення програмного застосунку, що має можливість сформувавши якісний звіт з виконаної студентом роботи є актуальною проблемою для розробників та додаток, що має необхідний функціонал, матиме попит серед користувачів. Ця проблема потребує подальших досліджень.

Висновок. Пандемія безповоротно змінила наше життя. Перспективу до вдосконалення системи дистанційного навчання в Україні складає впровадження в процес комп'ютерної, аудіо-візуальної техніки та спеціалізованих програмних застосунків. На даний момент проблему дистанційної освіти вирішують практично всі ВУЗи на території України. У зв'язку зі зміною принципів навчання виникли потреби в нових інструментах, які дозволяють виконувати завдання швидко та якісно. На цей час вже створено достатню кількість програмних застосунків, що сприяють дистанційній освіті. Більше того цим питанням активно займаються передові світові виробники. Не дивлячись на згадане вище є велика кількість проблем, що досі не вирішені.

Звіти під час дистанційної освіти, як і в будь-якій іншій професійній діяльності, займають важливе місце. Їх формування має бути швидким, простим та якісним, що не досягнуто на даний час. Отже ця проблема, безумовно, потребує вирішення.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Трайнев В. А. Информационные коммуникационные педагогические технологии : учеб. пособие /В. А. Трайнев, И. В. Трайнев. – К. : Освіта, 2008. – 327 с.
2. Хассон В. Дж. Критерії якості дистанційної освіти / В. Дж. Хассон, Е. К. Вотермен //Вища освіта. – 2004. – № 1. – С. 92-99.
- 3 Соловйов А. В. Огляд засобів генерації звітів // Документообіг. Концепції та інструментарій / Збірник праць ІСА РАН. Під ред. РАН В. Л. Арлазарова і д. Т. Н. проф. Н. Е. Ємельянова. М. : УРСС, 2004. С. 174-192.
4. Engelson V., Fritzson D., Fritzson P. Automatic Generation of User Interfaces from Data Structure Specifications and Object-Oriented Application. P. Coite (Ed.): ECOOP '96, LNCS 1098, 1996. P. 114–141.

УДК 004.415.3:681.6

А.Л. Ширін¹, А.О. Сергієнко¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ПЕРСПЕКТИВИ СТВОРЕННЯ ІДЕАЛЬНОЇ МОВИ ПРОГРАМУВАННЯ

Анотація. Описано аналіз існуючих мов програмування на предмет виявлення кращих сторін (властивостей, можливостей та ін.). Розглянуто можливість створення єдиної ідеальної мови програмування.

Ключові слова: *ідеальна мова, єдина мова програмування, віртуальна машина, реалізація об'єктно-орієнтованого програмування.*

Вступ. У нашому світі багато дивних речей, такі як: літаки, величезні кораблі, машини та ракети - все це складні механізми. Але що їх об'єднує? Раніше весь технічний прогрес був зосереджений на механізмах, проте зараз більшу частину перебирає на себе електроніка. Величезні обчислювальні центри побудовані по всьому світу й працюють над своїми чітко поставленими задачами. Одні прогнозують погоду, інші розраховують транспортний рух – це облегшує життя багатьом людям, хоча ми можемо цього не помічати. Тільки чому механізми схожі за будовою та архітектури виконують такі складні й різні обчислення? Вся справа в мовах програмування – це для комп'ютера, як для людини мова. Тільки як і в людському світі, так і в машинному – існує величезна кількість мов.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- зробити порівняння існуючих мов програмування побудованих за принципом віртуальної машини;
- викласти принципи побудови мов програмування для роботи з Інтернетом (серверна та клієнтська частина);
- визначити кращі властивості та можливості різних мов програмування працювати з оперативною пам'яттю, великою кількістю файлів, ресурсами тощо;
- зробити висновки щодо можливості створення єдиної ідеальної мови програмування.

Основний зміст роботи. За період навчання в школі та університеті мені пощастило працювати з багатьма мовами програмування, такими як: Java, C #, C ++, php, JavaScript і Kernal. Тільки наука завжди прагнула об'єднати все під єдиним стандартом. Тому існує вірогідність створення єдиної найкращої мови програмування, яка зможе поєднати в собі кращі сторони всіх створених мов і полегшити життя програмістам.

На початок пропоную розглянути дві, як на мене, схожі мови програмування Java і C #. Обидві мови програмування належать до сімейства C подібних мов і мають дуже багато спільного. Вони обидві побудовані за принципом віртуальної машини, яка має свої цікаві ідеї та проблеми. Здавалося б віртуальна машина – це рай для розробників під різні системи, проте це не так. Одна операційна система може мати один функціонал, якого немає в іншій і віртуальній машині потрібно балансувати на те, що є у них обох, а це суттєвий обріз функціональності.

Далі хочу розглянути php і JavaScript. Ці дві мови мають загальну область роботи з Інтернетом, але два різних напрямки. У той час, як php повністю серверна частина, JavaScript – клієнтська. Роботу з Інтернетом реалізовує теж так звана віртуальна машина, але трохи іншого характеру. Вона має назву Apache. JavaScript знаходиться в деякому симбіозі з іншими серверними мовами (jsp, ruby і т.д.), адже рідко коли виконується на стороні сервера (крім Node.js). Поки серверна частина обробляє варіанти сторінки, котрі має отримати користувач, клієнтська частина може швидко обробляти й реалізовувати дії користувача [1].

Прийшов час поговорити про досить стару, але по наш час поважну мову C++. Вона виникла дуже давно, коли люди ще користувались assembler. На відміну від інших ця пережила період бурхливого розвитку. Зараз, якщо перед програмістом стоїть завдання написати швидко й ефективно програму з найбільшим контролем з боку пам'яті й навантаження, то жодна інша мова так добре не підійде як ця. Тільки не дивлячись на всі її позитивні сторони, вона має свої недоліки, такі як: погана реалізація ООП, складність процесу розробки й незручна система роботи з великою кількістю файлів [2].

Ми розглянули декілька з найпоширеніших мов програмування, однак існують ще шейдери, які самі по собі працювати не будуть, такі як Kernal. Це набір інструкцій, які виконує відеокарта. Тільки чим вони відрізняються від інших мов? Вони завжди працюють з великою кількістю потоків одночасно. Звичайно, робота з потоками реалізована і в інших мовах, але якщо в них ми

будемо обраховувати мільярди однотипних формул і зводити все до одного, процесор буде задихатися від кількості запитів. Відеокарта ж спеціально створена для таких речей, тому особливих труднощів вона не відчуває в таких задачах [3].

Висновки. Попрацювавши з усіма цими мовами, приходиш до висновку, що створення ідеальної мови програмування - це не міф, а необхідність.

По-перше, мова повинна надавати вільний доступ як до обчислень на відеокарті, так і на процесорі. Це має дозволити набагато швидше перемикатися між завданнями й спростити розробку систем з великою кількістю паралельних обчислень.

По-друге, мова повинна мати добре опрацьовану систему ООП, як це реалізовано в Java і C#, де все належить до якогось об'єкту. Це має дозволити створювати величезні проекти та не заплутати в їх архітектурі. Було б добре поєднати різні типи класів, як це реалізовано в Java (Class, Interface, Enum), зі системою operator від C#, яка дозволяє звести взаємодію двох і більше схожих класів до одного оператора.

По-третє, я думаю, що ідеальній мові варто було б цілком і повністю відмовитися від роботи на віртуальній машині й працювати безпосередньо на машинному рівні, як C++. Такий спосіб втратить можливість швидкого перенесення на різні платформи, але дозволить використовувати всі можливості операційної системи. Також, вважаю, що програміст повинен мати вибір, чи хоче він використовувати заготовлений автоматичний збирач сміття, або написати свій власний.

Перелік посилань

1. Мельник Р. А. / Програмування веб-застосунків (фронт-енд та бек-енд) // Львів: Львівська політехніка. – 2018.
2. Трофіменко, О. Г., Прокоп, Ю. В., Логінова, Н. І., Задерейко, А. В. / C++. Алгоритмізація та програмування: підручник. // Одеса: Фенікс – 2019.
3. Kaeli, D. R., Mistry, P., Schaa, D., & Zhang, D. P. / Heterogeneous computing with OpenCL 2.0. // USA: Morgan Kaufmann. – 2015.

РОЗДІЛ 4

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ СИСТЕМ ОБРОБКИ ІНФОРМАЦІЇ ДЛЯ ВИРІШЕННЯ ЗАВДАНЬ ОСВІТИ, НАУКИ І УПРАВЛІННЯ ВИРОБНИЦТВОМ

УДК 004.415.3:681.6

Б.І. Мороз¹, О.О. Шишацький¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ВИКОРИСТАННЯ ОПТИЧНИХ ДЕТЕКТОРІВ ДОРОЖНЬОГО РУХУ У СИСТЕМАХ АДАПТИВНОГО КЕРУВАННЯ СВІТЛОФОРАМИ

Анотація. Описано процес покращення систем адаптивного керування світлофорами шляхом використання оптичних детекторів дорожнього руху.

Ключові слова: відеоспостереження, трафік, затори, автомобілі, адаптивне керування, світлофори.

Вступ. З ростом автомобілізації перед дорожніми службами поступово постало завдання оптимізації пропускної здатності дорожньої мережі. Загальновідомо, що пропускна здатність вулиці визначається, переважно, пропускною здатністю перехрестя, тому поступово були розроблені та впроваджені найпростіші контролери світлофорних об'єктів. Регулювання, при якому кожен світлофорний об'єкт програмується на періодичність та тривалість сигналів, називається жорстким, оскільки не враховує потоки транспорту та пішоходів на сусідніх перехрестях, не реагує на дорожню обстановку (аварії, тощо), не зважає на час доби. Трохи покращує ситуацію багатопрограмне жорстке регулювання, що змінює час циклів протягом доби. Це зменшує середній час очікування на перехресті, проте не є найоптимальнішим, бо не може обробляти випадкове прибуття транспорту або пішоходів до перехрестя (наприклад, у випадку ДТП). Вирішення проблеми стало можливим лише з появою більш потужних контролерів, що могли реалізувати обробку зворотного зв'язку від транспортного потоку.

Неоптимальні цикли світлофорного регулювання викликають затримки міського транспорту, пішоходів, особистого транспорту та дратують учасників дорожнього руху. Також неоптимальне регулювання впливає на екологію: середньостатистичний автомобіль з двигуном типу ЕВРО-5 щохвилини викидає в повітря близько 1 граму шкідливих речовин: оксиди азоту, вуглекислий газ, сажа. Одночасна концентрація багатьох автомобілів в одному місці перед світлофором в результаті затору понаднормово збільшує забруднення повітря в конкретному місці.

Так, за даними моніторингу, джерелом 78% викидів речовин-забрудників повітря у Києві 2018 року були автомобілі. Концентрації деяких забрудників у повітрі перевищують гранично допустимі рівні. У 2019 році у великих містах України були постійно перевищені концентрації діоксиду азоту, формальдегіду

і оксиду азоту; час від часу – концентрації завислих речовин, оксиду вуглецю та фенолу. Різні дослідження оцінюють ризики від забруднення у 240-1900 додаткових випадків раку на один мільйон осіб протягом середньої тривалості життя (70 років).

Одна з найкращих систем адаптивного керування дорожнім трафіком на даний момент – це програмне рішення (схему представлено на рис. 1) SCOOT від компанії Siemens [1]. В порівнянні з іншими системами, SCOOT пропонує експлуатантам деякі важливі переваги як-от проактивний аналіз трафіку, тобто ще до того, як він досягне перехрестя. Та незважаючи на відмінності, база подібних адаптивних систем незмінна – це комплекс детекторів різного типу, що встановлюються біля перехресть.



Рис. 1. Схема систем типу SCOOT

Найперше та найочевидніше рішення – детектор на основі ваги автомобіля – було запропоновано ще в 1930 р., проте не стало розповсюдженим через проблеми в експлуатації. На перших дослідних полігонах було виявлено проблему: взимку снігоочисники пошкоджували рухомі деталі детектору, що призводило до дорогого ремонту. Також були витрати на переустановлення детектора після зміни профілю дороги. Ці проблеми призвели до пошуку датчиків потоку руху на основі більш тонких властивостей, таких як: звук (акустичні датчики), непрозорість (оптичні та інфрачервоні датчики), геомагнетизм (магнітні датчики, магнітометри), тощо.

Постановка задачі. Для виконання поставленої задачі у ході роботи були сформульовані й вирішені наступні завдання:

1. проаналізувати існуючі рішення для детекторів дорожнього руху (далі – ДР), визначити їх слабкі та сильні сторони, відсотковий показник коректності розпізнавання різних класів об'єктів;

2. проаналізувати існуючі технології та алгоритми розпізнавання об'єктів на відео, визначити ті, що підходять для вирішуваної задачі;

3. запроектувати та розробити демонстраційну модель оптичного детектору, що має успішно підраховувати кількість автомобілів та пішоходів, що перетинають перехрестя. Визначити ефективність моделі та порівняти її з існуючими рішеннями;

4. проаналізувати, як заміна детекторів на оптичний впливає на систему ATLC та як змінюється її ефективність.

Основний зміст роботи. Усі існуючі варіанти датчиків мають свої переваги та недоліки, проте найперспективнішим бачиться саме оптичний детектор, адже, на відміну від інших, потенційно здатен не лише розпізнавати автомобілі, але й проводити більш складну класифікацію об'єктів: пішоходів, велосипедистів, автобусів (приклад див. на рис. 2).

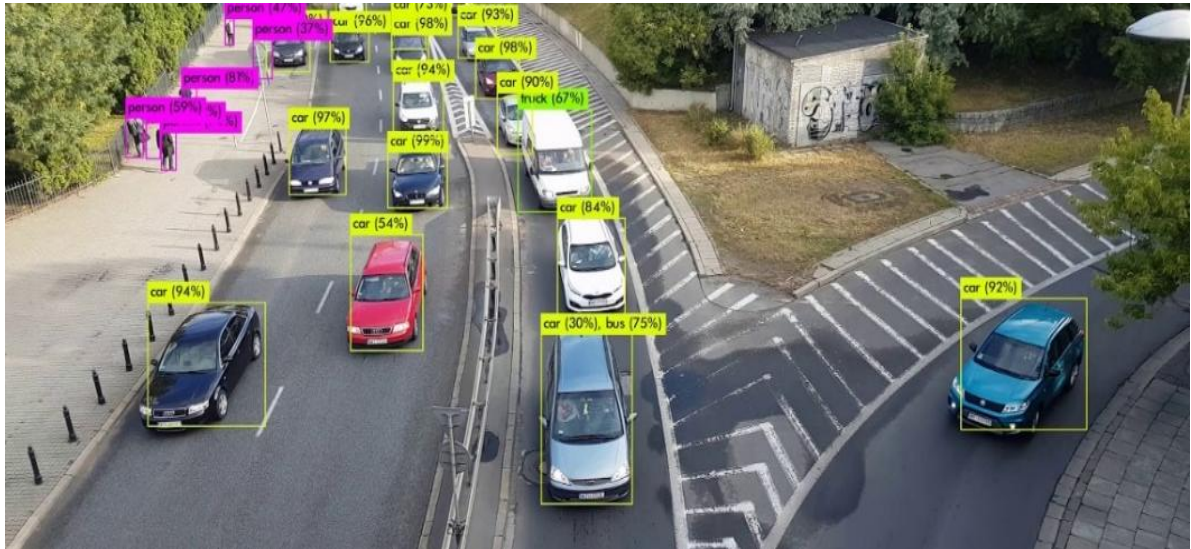


Рис. 2. Результат роботи нейронної мережі YOLO COCO

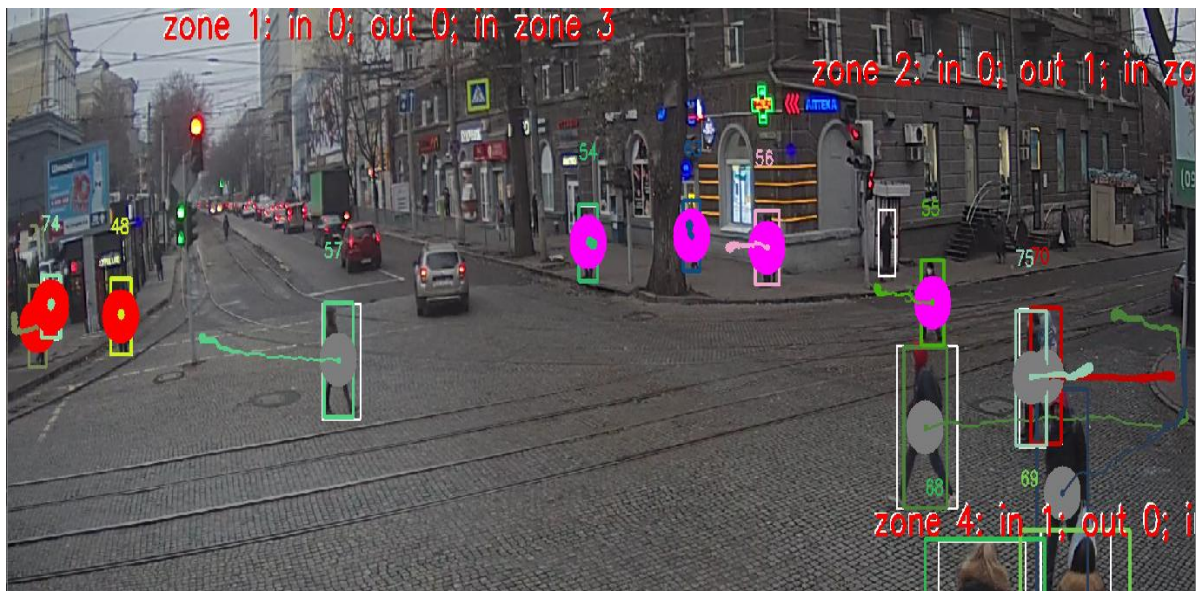


Рис. 3. Результат розробленої системи

Правильним чином натренована нейронна мережа може також ідентифікувати один і той самий об'єкт що з'являється в полі зору різних камер. Таким чином, встановлення декількох камер на перехресті може

потенційно замінити з десяток, наприклад, наразі більш розповсюджених датчиків на основі електромагнітної індукції [2].

Наукова новизна полягає в удосконаленні методів регулювання дорожнього руху за допомогою нових алгоритмів та принципів роботи детекторів автомобільного та пішохідного руху, заснованих на технології комп'ютерного бачення.

Висновки. В результаті роботи було розроблено алгоритм оптичного детектору автомобілів та пішоходів, який в результаті проведених експериментальних досліджень показав ефективність у 58% правильно оброблених об'єктів вночі та 75% – вдень. Визначені показники ефективності було закладено у модель системи адаптивного керування дорожнім рухом та додатково проведено моделювання руху автомобілів на перехресті для двох варіантів системи: класичної та удосконаленої. В якості показника ефективності перехрестя визначено середній час очікування учасників дорожнього руху на перехресті. В результаті визначено, що ефективність системи з новим типом детектору відносно індуктивно падає лише на 15%. Для порівняння, без системи адаптивного контролю, ефективність перехрестя падає на 50-70%.

Таким чином, вважаємо, що представлене удосконалене рішення може використовуватись, адже подібні системи передбачено встановлювати там, де їх немає наразі взагалі, а простота встановлення та експлуатації, відносно класичних систем, нівелює недоліки у вигляді меншої ефективності. Отже, розробка та подальше вдосконалення програмного забезпечення для роботи оптичних детекторів є реальним та перспективним завданням, бо така технологія дозволить значно зменшити затрати на впровадження адаптивних систем.

Перелік посилань

1. Johnson E.E. Completing an MIMD Multiprocessor Taxonomy // Computer Architecture News, 1988. – Vol. 16. –No 2. – P. 44-48.
2. D. Parallel Programming with Microsoft Visual Studio 2010 Step by Step / D. Marshall. – O'Reilly Media, Inc., 2011. – 249 p.
3. McCool M., Robison A.D., Reinders J. Structured Parallel Programming: Patterns for Efficient Computation. – Morgan Kaufmann, 2012. – 433 p.
4. 82 26. Midkiff S. P. Automatic Parallelization. An Overview of Fundamental Compiler Techniques / S. P. Midkiff. – Morgan & Claypool, 2012. – 170 p.
5. Miller R., Boxer L. Algorithms Sequential & Parallel: A Unified Approach / R. Miller, L. Boxer. – Cengage Learning, 2012. – 448 p.
6. Stephen F. Smith Initial Application of the SURTRAC Adaptive Traffic Signal Control System.

Л.І. Мещеряков¹, С.І. Випанасенко¹, Н.С. Дрешпак¹, Д.О. Трофименко¹
¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

РОЗРОБКА ТА ОПТИМІЗАЦІЯ МЕТОДІВ І АЛГОРИТМІВ АВТОМАТИЧНОГО УПРАВЛІННЯ РОЗГОРНУТИМИ КОНТЕЙНЕРІЗОВАНИМИ ВЕБ-ДОДАТКАМИ ТА СЕРВІСАМИ

Анотація. Описано процес оптимізації автоматичного управління при роботі з розгорнутими веб-додатками. Оптимізація досягається за рахунок розробки та інтегрування власних методів та алгоритмів автоматизованого управління у платформу Kubernetes.

Ключові слова: контейнеризовані додатки, image-образ, Docker, Kubernetes, под, деплоймент, сервіс, кластер, оператор.

Вступ. Однією з найбільших проблем для підприємств, які застосовують практики DevOps та хмарні можливості, є підтримка загальних та послідовних середовищ протягом усього життєвого циклу додатку. Проблеми виникають у коректному функціонуванні додатків без належного управління конфігурацією. Конфігурації додатків часто суперечать власним цілям розробників щодо безпеки, масштабованості та доступності. Проблема гнучкої доставки полягає у тому, що конфігурація додатків та конфігурація інфраструктури повинні бути досягнуті спільно, але мають чітку визначену власність та чітке розділення ролей. Kubernetes – це портативна розширювана платформа з відкритим вихідним кодом для управління контейнеризованими робочими навантаженнями та сервісами, що спрощує як декларативне налаштування, так і автоматизацію. З популярністю контейнерів, Kubernetes (K8s) став найпопулярнішою платформою для організації контейнерів. Додатки, упаковані як контейнери, можуть бути розгорнуті в будь-якому середовищі K8s, що працює де завгодно, і програма буде поводитися однаково завдяки незмінності. Наразі в сфері автоматизації управління не хватає достатньої кількості методів та алгоритмів щоб забезпечити необхідну гнучкість налаштування та управління роботою розгорнутих веб-додатків та сервісів.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання: викласти принципи роботи платформи Kubernetes; викласти принципи автоматичного управління розгорнутими веб-додатками; визначити основні, мінімально необхідні компоненти системи; оптимізувати та розширити методи та алгоритми автоматичного управління веб-додатками, шляхом аналізу напрацювань у цій сфері та вибору найбільш підходящих технологій; спроектувати взаємодію ПЗ з платформою K8s, шляхом інтегрування першого; зробити висновки щодо доцільності розробки подібних алгоритмів.

Основний зміст роботи. Під час роботи були використані наступні методи та інструменти: метод контейнеризації додатків; шаблон автоматизації

оператор з користувацькими ресурсами; Docker, Kubernetes, Java, Maven, MySQL.

При розгортанні Kubernetes уся робота виконується у кластері. Кластер K8s складається з набору машин, так званих вузлів, які запускають контейнеризовані додатки. Кластер має як мінімум один робочий вузол. У робочих вузлах розміщені поди, що є основними компонентами програми. Площина управління керує робочими вузлами і подами в кластері. У промислових середовищах площина управління зазвичай запускається на декількох комп'ютерах, а кластер, як правило, розгортається на кількох вузлах, гарантуючи відмовостійкість і високу надійність.

Компоненти панелі управління відповідають за основні операції кластера (наприклад, планування), а також обробляють події кластера (наприклад, запускають новий под, коли поле replicas розгортання не відповідає необхідній кількості реплік). Компоненти панелі управління можуть бути запущені на будь-якій машині в кластері.

Серед основних компонентів K8s було виділено наступні: поди; деплойменти; сервіси.

Поди - це найменші одиниці обчислювальної техніки, які можна створити та керувати ними в Kubernetes. Под – це група одного або декількох контейнерів із загальними ресурсами зберігання або мережі та специфікацією запуску контейнерів. Вміст подів завжди розміщується спільно та за розкладом і працює у спільному контексті. Под моделює специфічний для програми «логічний хост»: він містить один або кілька контейнерів програм, які відносно тісно пов'язані. У нехмарних контекстах програми, що виконуються на одній фізичній або віртуальній машині, є аналогами хмарних програм, що виконуються на одному логічному хості. У контексті поду, окремі додатки можуть застосовувати подальші субізоляції. Кожен под призначений для запуску одного екземпляра програми. Якщо є необхідність масштабувати додаток горизонтально (щоб забезпечити більше загальних ресурсів, запустивши більше екземплярів), потрібно використовувати декілька подів, по одному для кожного екземпляра. У Kubernetes це зазвичай називають реплікацією. Реплікаційні підсистеми зазвичай створюються та керуються у вигляді групи за допомогою ресурсу робочого навантаження та його контролера. Поди розроблені як відносно швидкоплинні одноразові сутності. Коли под створюється (безпосередньо користувачем або власне контролером), новий под планується запуснитися на вузлі у кластері. Под залишається на цьому вузлі до тих пір, поки не закінчить виконання, об'єкт буде видалений, под знищений за браком ресурсів або вузол не вийде з ладу. Можливо використовувати ресурси робочого навантаження, щоб створити та управляти кількома подами. Контролер ресурсу обробляє реплікацію та розгортання та автоматичне загоєння у випадку відмови поду. Наприклад, якщо вузол виходить з ладу, контролер помічає, що поди на цьому вузлі перестали працювати, і створює заміну під. Планувальник розміщує заміник поду на здоровий вузол. Контролери ресурсів робочого навантаження створюють

підсистеми з шаблону підсистеми та керують цими підсистемами від імені користувача.

Сервіси – це абстрактний спосіб виставити програму, що працює на наборі подів, як послугу мережі. З Kubernetes не потрібно модифікувати програму, щоб використовувати незнайомий механізм виявлення служби. Kubernetes надає подам їх власні IP-адреси та єдине DNS-ім'я для набору подів, і може балансувати навантаження між ними. Поди Kubernetes створюються та знищуються відповідно до стану кластера. Поди - це непостійні ресурси. Якщо використовується Deployment для запуску програми, він може динамічно створювати та знищувати поди. Кожен под отримує свою власну IP-адресу, однак у розгортанні набір подів, запущених за один момент часу, може відрізнитися від набору подів, що запускає цю програму мить пізніше. Це призводить до проблеми: якщо якийсь набір подів (назвемо їх "backends") надає функціональність іншим подам (назвемо їх "frontends") всередині кластера, як інтерфейси дізнаються та відстежують, до якої IP-адреси потрібно підключитися, щоб інтерфейс міг використовувати внутрішню частину робочого навантаження? Для цього вводяться сервіси. У Kubernetes сервіс - це абстракція, яка визначає логічний набір подів та політику, за допомогою якої можна отримати до них доступ (іноді цей шаблон називається мікросервісом). Набір подів, на які спрямована послуга, зазвичай визначається селектором.

Деплоймент представляє собою набір з декількох однакових подів без унікальних ідентифікаційних даних. Деплоймент запускає кілька реплік додатку та автоматично замінює будь-які екземпляри, які не працюють або не реагують. Таким чином, деплоймент допомагає забезпечити доступність одного або декількох екземплярів програми для обслуговування запитів користувачів. Деплойменти використовують шаблон поду, який містить специфікацію для своїх подів. Специфікація поду визначає, як повинен виглядати кожен под: які програми повинні запускатися всередині його контейнерів, які обсяги слід монтувати поду, його мітки тощо. За замовчуванням, коли розгортання запускає оновлення, деплоймент зупиняє структури, поступово зменшує кількість структур до нуля, потім зливає та припиняє структури. Потім розгортання використовує оновлений шаблон поду для створення нових подів. Старі поди не видаляються, поки не запускається достатня кількість нових подів, а нові поди не створюються, поки не буде видалено достатню кількість старих подів. Розгортання можуть гарантувати, що працює щонайменше на одну менше ніж бажано кількості реплік, причому щонайменше один под недоступний. Подібним чином, розгортання можуть забезпечити роботу щонайбільше більше на одну, ніж бажана кількість реплік, причому щонайменше на один под більше, ніж бажано. Частіше за все, деплоймент виповнює одне з наступних завдань: масштабування розгортання, автомасштабування розгортання, видалення розгортання.

Контролери – це цикли управління, які спостерігають за станом кластера, а потім вносять або вимагають змін, де це необхідно. Кожен контролер намагається перемістити поточний стан кластера ближче до бажаного стану. Контролер відстежує принаймні один тип ресурсу K8s. Ці об'єкти мають поле

специфікації, яке представляє бажаний стан. Контролери для цього ресурсу відповідають за наближення поточного стану до бажаного стану. Контролер може здійснити дію самостійно, частіше в Kubernetes контролер надсилатиме на сервер API повідомлення, які мають корисні побічні ефекти. Вбудовані контролери управляють станом, взаємодіючи з сервером API кластера.

Робота – це ресурс K8s, який запускає поди, або, можливо, декілька подів, для виконання завдання, а потім зупинки. Коли контролер завдання бачить нове завдання, він переконується, що у кластері на наборі вузлів запускається потрібна кількість подів, щоб виконати роботу. Натомість контролер завдання вказує серверу API створювати або видаляти підсистеми. Після створення нового завдання бажаним станом є виконання цього завдання. Контролер завдання робить поточний стан для цього завдання ближчим до бажаного стану: створюючи структури, які виконують роботу, яку необхідно для цього завдання, щоб завдання було ближче до завершення. Контролери також оновлюють об'єкти, які їх налаштовують. Контролери, які взаємодіють із зовнішнім станом, знаходять бажаний стан із сервера API, а потім взаємодіють безпосередньо із зовнішньою системою, щоб наблизити поточний стан у відповідність. Важливим моментом тут є те, що контролер вносить певні зміни для досягнення бажаного стану, а потім повідомляє про поточний стан назад на сервер API кластера.

Kubernetes можна легко налаштувати та розширити. Як результат, рідко виникає необхідність подавати виправлення до коду проекту K8s. Підходи до налаштування можна розділити на конфігурацію, яка передбачає лише зміну прапорів, локальних файлів конфігурації або ресурсів API, та розширення, які передбачають запуск додаткових програм або послуг. Прапори та файли конфігурації не завжди можуть бути зміненими у розміщеній службі K8s або розповсюдженні з керованою інсталяцією. Коли вони міняються, вони, як правило, змінюються лише адміністратором кластера. Крім того, вони можуть бути змінені в майбутніх версіях K8s, і для їх налаштування може знадобитися перезапуск процесів. З цих причин їх слід використовувати лише тоді, коли інших варіантів немає.

Розширення – це програмні компоненти, які розширюються та глибоко інтегруються з K8s. Вони адаптують його для підтримки нових типів та нових видів обладнання.

Спеціальні ресурси – це розширення API Kubernetes. Ресурс – це кінцева точка в API K8s, що зберігає колекцію об'єктів API певного типу, наприклад, вбудований ресурс под містить колекцію об'єктів подів. Спеціальні ресурси не обов'язково доступні в установці K8s за замовчуванням. Вони являють собою налаштування конкретної установки K8s. Однак багато основних функцій побудовані з використанням власних ресурсів, що робить Kubernetes більш модульним. Спеціальні ресурси можуть з'являтися та зникати в запущеному кластері за допомогою динамічної реєстрації, а адміністратори кластера можуть оновлювати власні ресурси незалежно від самого кластера. Після встановлення користувацького ресурсу користувачі можуть створювати та отримувати доступ до його об'єктів за допомогою `kubectl`, як це роблять для вбудованих ресурсів,

таких як поди. Спеціальні ресурси самостійно дозволяють зберігати та отримувати структуровані дані. Коли поєднується власний ресурс із користувацьким контролером, користувацькі ресурси надають справжній декларативний API.

Оператор – це патерн розробки для автоматизації роботи більш складних додатків або частин інфраструктури шляхом створення подів (контейнерів), які працюють у Kubernetes та мають вбудовану логіку управління. Ця логіка називається контуром управління. Цикл управління може бути реалізований за допомогою Java або будь-якої іншої мови програмування загального призначення, яка може спілкуватися з API K8s. Вхідними даними для циклу керування є об'єкти, створені на сервері Kubernetes API, який підтримує власні схеми для цих об'єктів. Вони називаються спеціальними (кастомними) ресурсами.

Оператори можуть використовуватись для автоматизації у таких випадках: розгортання додатку за необхідністю; отримання та відновлення резервних копій стану цього додатка; обробка та оновлення коду програми разом із відповідними змінами, такими як схеми бази даних або додаткові налаштування конфігурації; імітація повного або часткового збою у кластері, щоб перевірити його стійкість.

Найпоширеніший спосіб розгортання оператора – це додавання Custom Resource Definition та пов'язаного з ним контролера до кластера. Контролер, як правило, працює поза площиною управління, подібно до того, як запускається будь-яка контейнерна програма. Після розгортання оператора можливо використовувати його, додаючи, змінюючи або видаляючи тип ресурсу, який використовує оператор. Якщо в екосистемі немає оператора, який реалізує потрібну поведінку, можливо створити свій власний.

Після аналізу проблем предметної області, було вирішено оптимізувати методи та алгоритми автоматичного управління розгорнутими контейнеризованими веб-додатками та сервісами.

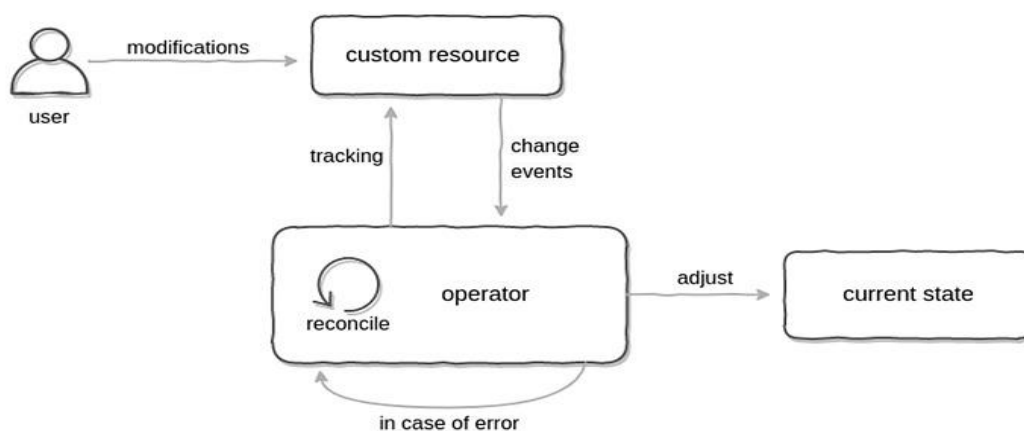


Рис. 1. Логіка роботи оператора та користувацьких ресурсів

Було визначено, що для оптимізації управління веб-додатками необхідно: розробка власного оператора, який буде виконувати роль контролера та стежити за користувацькими ресурсами, за їх бажаним та теперішнім станом та буде намагатися змінювати стан, якщо він не є бажаним (рис. 1); створення власних (користувацьких) ресурсів, які будуть у повній мірі описувати додаток, або його сервіс та будуть керуватися оператором; інтегрування користувацьких ресурсів та оператору з платформою Kubernetes.

Цього можна досягти розробивши патерн оператор на Java, який виступає аналогом контролеру у Kubernetes, та успадковується платформою K8s і вдало інтегрується у неї. Оператор виконує команди, які направлені на підтримку бажаного стану об'єктів, він слідкує за ними увесь час та намагається виправити ситуацію, у якій стан об'єктів виходить із бажаного стану. Підтримка бажаного стану виконується за рахунок базових команд створення, оновлення та знищення.

Оператор слідкує та підтримує бажаний стан користувацьких ресурсів, які було створено із Custom Resource Definition. Оператор постійно відстежує події кластера, що стосуються певного типу користувацьких ресурсів. Коли оператор отримує будь-яку інформацію, він вживає заходів для регулювання кластера Kubernetes або зовнішньої системи до бажаного стану як частини свого циклу узгодження в користувацькому контролері. Спеціальний ресурс розширює можливості K8s, додаючи нові види об'єктів, які можна розгортати, масштабувати та балансувати згідно з впровадженою конфігурацією.

Наукова новизна: удосконалено та поширено функціонал методів та алгоритмів автоматичного управління контейнерізованими веб-додатками та сервісами за рахунок розробки операторів, користувацьких ресурсів і конфігурацій та інтегрування з платформою Kubernetes; отримав подальший розвиток метод розробки операторів та користувацьких ресурсів.

Висновки. Було оптимізовано та розширено методи та алгоритми автоматизації управління розгорнутими контейнерізованими веб-додатками та сервісами у середі K8s. Оптимізація відбулася за рахунок розробки власних операторів-контролерів, впровадження логіки нагляду за користувацькими ресурсами, створення нових об'єктів, які описують додатки і сервіси та інтегрування з платформою Kubernetes.

Перелік посилань

1. Marko Luksa Kubernetes in Action: 1st Edition, 2019 – 272 с.
2. Brayden Smith Kubernetes: A Step-by-Step Guide to Learn and Master Kubernetes, 2018 – 162с.
3. Bilgin Ibryam Rolan Huss Kubernetes Patterns: Reusable Elements for Designing Cloud-Native Applications, 2019 – 197 с.
4. Eva Tuczai Asena Hertz Managing Kubernetes Performance at Scale, 2019 – 87 с.
5. James Turnbull The Docker Book: Containerization is the new virtualization, 2018 – 502 с.

КЕРУВАННЯ ДОВЖИНОЮ ДРІБНОСОРТНОГО ПРОКАТУ ЗА РАХУНОК ЗБІЛЬШЕННЯ ДОВЖИНИ ЙОГО СЕРЕДНЬОЇ ДІЛЯНКИ

Анотація. Представлено базові основи керування довжиною дрібносортного прокату за рахунок збільшення довжини його середньої ділянки

Ключові слова: дрібносортний прокат, система керування, середньої ділянки прокату, розроблено програмне забезпечення, математичний пакет MATLAB.

Вступ. Дрібносортний прокат одна з найбільш затребуваних експортних позицій. В даний час цей тип прокату випускається на високопродуктивних дрібносортних станах. Основні витрати при виробництві на таких станах припадають на матеріали вихідної заготовки, які складають більше 90% від собівартості прокату. При виробництві мірного прокату утворюється технологічна обрізь, що призводить до втрати товарної продукції, тому питання скорочення технологічного обрізі є актуальними [1].

Основний зміст роботи. Зазвичай прокатний стан налаштовується таким чином, щоб геометричні розміри середньої ділянки лежали в мінусовому полі допуску [2], а кінцева ділянка в плюсовому. Таким чином пріоритетною стратегією є зменшення довжини середньої ділянки за рахунок збільшення її геометричних розмірів. При не виконанні вимог даної стратегії запропоновано використати стратегію збільшення середньої ділянки прокату. Дана стратегія використовується в разі, коли недостатня до міри прогнозована довжина прокату менша діапазону регулювання довжини середньої ділянки при прокатці з негативним полем допуску:

$$l_{MC} - l_H < \Delta l_{CDHPD}, \quad (1)$$

де l_{MC} – довжина мірного стрижня (м), l_H – довжина немірної частини прокату (м), Δl_{CDHPD} – діапазон регулювання довжини середньої ділянки прокату при прокатці з негативним полем допуску (м).

Довжина немірної частини прокату є різницею між довжиною прокату та довжиною прокату кратною мірній довжині:

$$l_H = l - l_{KM}, \quad (2)$$

де l – довжина прокату (м), l_{KM} – довжина прокату кратна мірній довжині (м).

Довжина мірної частини прокату залежить від довжини мірного стрижня:

$$l_{\text{KM}} = \left\lfloor \frac{l}{l_{\text{МС}}} \right\rfloor \cdot l_{\text{МС}}, \quad (3)$$

де $l_{\text{МС}}$ – довжина мірного стрижня (м), $\lfloor \]$ – округлення до найближчого меншого цілого.



Рис. 1. Стратегія збільшення довжини середньої ділянки прокату

Діапазон регулювання довжини середньої ділянки прокату при прокатці з негативним полем допуску визначається виходячи з довжин середніх ділянок:

$$\Delta l_{\text{СДНПД}} = l_{\text{СДНПД}} - l_{\text{СД}}, \quad (4)$$

де $l_{\text{СДНПД}}$ – довжина середньої ділянки прокату при прокатці з негативним полем допуску (м), $l_{\text{СД}}$ – довжина середньої ділянки прокату (м).

Довжина середньої ділянки прокату при прокатці з негативним полем допуску визначається на підставі об'єму середньої ділянки прокату та його геометричних розмірів, одержаних при прокатці з негативним полем допуску:

$$l_{\text{СДНПД}} = \frac{V_{\text{СД}}}{h_{\text{НПД}} \cdot b_{\text{НПД}}}, \quad (5)$$

де $V_{\text{СД}}$ – об'єм середньої ділянки прокату (м^3), $h_{\text{НПД}}$ – висота прокату при прокатці з негативним полем допуску (м), $b_{\text{НПД}}$ – ширина прокату при прокатці з негативним полем допуску (м).

Об'єм середньої ділянки прокату залежить від її довжини та геометричних розмірів:

$$V_{\text{СД}} = l_{\text{СД}} \cdot h_{\text{СД}} \cdot b_{\text{СД}}, \quad (6)$$

де $h_{\text{СД}}$ – висота прокату на середній ділянці (м), $b_{\text{СД}}$ – ширина прокату на середній ділянці (м).

При виконанні умови (1) довжина середньої ділянки прокату формована системою керування визначається як:

$$l_{сдс} = l_{сд} + l_{мс} - l_{н}, \quad (7)$$

де $l_{сдс}$ – довжина середньої ділянки прокату формована системою керування (м).

Висновки. Відповідно до запропонованої стратегії в математичному пакеті MATLAB розроблено програмне забезпечення яке виконує прогнозування довжини прокату, визначення стратегії керування довжиною його середньої ділянки та результат порізки прокату на прутки. Використання запропонованої стратегії керування довжиною прокату на його середній ділянці дозволило зменшити кількість технологічної обрізі, за умови коли недостатня до міри прогнозована довжина прокату менша діапазону регулювання довжини середньої ділянки прокату при прокатці з негативним полем допуску.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Бешта А.С. Система рациональных технико-экономических показателей при производстве мелкосортного мерного проката в стержнях / А.С. Бешта, О.А. Бойко, Т.В. Куваева // Збірник наукових праць НГУ. – 2015. – №48. – С. 183 – 188.

2. Куваєв В.М. Зниження втрат металу на основі узгодженого керування довжиною і розрізанням сортового прокату / В.М. Куваєв, О.О. Бойко, Н.О. Новодранова // XVII науково-технічної конференції «ПОТУРАЇВСЬКИ ЧИТАННЯ»: тез. доп. міжнар. наук.-практ. конф., Дніпро, 25 січня 2019 р. – Д. НТУ «ДП», 2019. – С. 13.

УДК 681.5.017

А.М. Мацуй¹

¹Центральноукраїнський національний технічний університет, Кропивницький, Україна

ТЕОРЕТИЧНІ ОСНОВИ ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНОЇ СИСТЕМИ ОЦІНЮВАННЯ РОБОТИ КУЛЬОВИХ МЛИНІВ З МЕТОЮ ЕНЕРГОЗБЕРЕЖЕННЯ

Анотація. Вперше отримана статична математична модель прямого оцінювання об'єму зруйнованої руди в запропонованому первинному перетворювачі, що характеризує роботу кульового млина. Інваріантно скомпенсовано вплив на результат оцінювання спрацювання футеровки та непрямих ударів куль. На цій базі реалізована інформаційно-вимірjuвальна система.

Ключові слова: інформаційно-вимірювальна система; оцінювання роботи кульових млинів; первинний, вторинний перетворювачі.

Вступ. Кульові млини, які широко використовують в процесах рудопідготовки на збагачувальних фабриках, відрізняються низьким механічним коефіцієнтом корисної дії і непродуктивно перевитрачають багато електричної енергії. Це робить вітчизняний залізорудний концентрат та вироби з нього неконкурентоспроможними на міжнародному ринку, що приводить до вагомих збитків.

Однією з вагомих причин такого становища є відсутність інформаційно-вимірювальних систем оцінювання ефективності руйнування руди безпосередньо в кульових млинах перших стадій подрібнення. Автором даної публікації здійснена спроба вирішення даної задачі.

Постановка задачі. В роботі сформовані і вирішені наступні завдання:

- запропоновано первинний перетворювач оцінювання об'єму зруйнованої руди в конкретному перерізі барабана млина, що характеризує ефективність подрібнення;
- розроблені теоретичні основи інформаційно-вимірювальної системи прямого оцінювання руйнування руди безпосередньо в кульовому млині;
- отримана статична математична модель оцінювання об'єму зруйнованої руди в конкретному перерізі барабана млина;
- запропоновано інваріантний підхід підвищення точності оцінювання об'єму зруйнованої руди;
- обґрунтовано тип вторинного перетворювача;
- визначена результируюча відносна похибка інформаційно-вимірювальної системи прямого оцінювання роботи кульових млинів.

Основний зміст роботи. В процесі виконання досліджень були використані наступні методи:

- методи теорії кульових млинів;
- методи теорії удару;
- методи деформації пружних тіл;
- методи вимірювання деформацій та малих переміщень;
- методи теорії похибок;
- інваріантні методи компенсації збурень.

Методами теорії кульових млинів встановлено, що в них можливо створити перерізи барабана, де в зовнішньому шарі куль будуть діяти молотні тіла одного розміру, які рухаються з незмінною швидкістю. Тоді кінетична енергія цих подрібнюючих тіл в даному перерізі буде незмінною. Якщо в точці удару такої кулі розмістити перетворювач, то він може фіксувати втрату кінетичної енергії в залежності від того, який об'єм руди зруйновано. Об'єм зруйнованої руди падаючою кулею визначається концентрацією крупних шматків матеріалу в пульпі. Концентрація крупних шматків матеріалу в пульпі є мірою недовантаження, номінального завантаження та перевантаження млина рудою. Тенденція зміни всіх критеріїв концентрації (об'єму, маси, кількості крупних часток руди в одиниці об'єму пульпи) однакова. На концентрацію крупних шматків руди можливо впливати зміною вихідного живлення млина,

забезпечуючи оптимальне завантаження і найкращу ефективність подрібнення матеріалу. За втратою кінетичної енергії кулі можливо оцінювати ефективність подрібнення руди в кульовому млині і визначати найкращий режим подрібнення, коли електрична енергія, споживана технологічним агрегатом, не перевитрачається. Тобто, можливо реалізувати пряме вимірювання технологічного параметра безпосередньо у кульовому млині. Первинний перетворювач інформаційно-вимірювальної системи оцінювання роботи кульового млина можливо реалізувати відповідно схемі зображеній на рис.1. При ударі кулі 4 в торець стержня 3 руда 5 руйнується, на що витрачається частина її кінетичної енергії. Залишкова кінетична енергія молоткового тіла через стержень 3 деформує пружну пластину 2, яка прогинається у центральній частині, оскільки її кінці розміщені на опорах 1. Знайдемо залежність величини прогину центральної частини пружної пластини 2 від сили удару. Будемо розглядати центральний удар, при якому в мить зіткнення тіл їх центри інерції знаходяться на лінії удару.

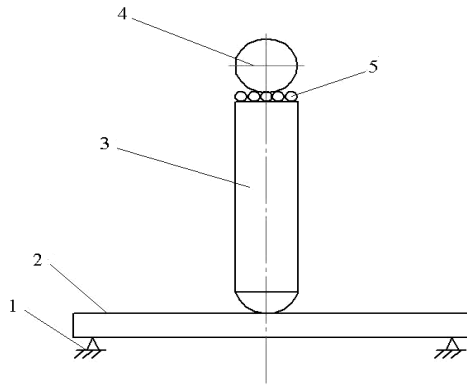


Рис.1. Схематичне зображення первинного перетворювача інформаційно-вимірювальної системи оцінювання роботи кульового млина: 1 – опори; 2 – пружна пластинка; 3 – стержень; 4 – куля; 5 – шматки крупної руди

Розглянемо перетворювач без руди. Робота, виконана при падінні кулі з висоти h до найбільшого прогину пластини x_{II} , буде дорівнювати

$$A_p = m_k g h + x_{c1} + x_{II}, \quad (1)$$

де m_k – маса кулі; x_{c1} – скорочення стержня при ударі.

Виконана при падінні кулі робота перетворюється в кінетичну енергію деформованої пружної пластини і стержня

$$E_n = \frac{c_c x_{c1}^2}{2} + \frac{c_{II} x_{II}^2}{2},$$

де c_c, c_{II} – відповідно коефіцієнти жорсткості стержня і пружної пластини. Укорочення стержня буде дорівнювати

$$x_{c1} = \frac{l_0 F}{E_{Ю} S_C},$$

де l_0 – довжина стержня; F – прикладена сила до його торця; $E_{Ю}$ – модуль Юнга матеріалу; S_C – площа поперечного перерізу стержня.

Аналіз показує, що в даних умовах параметри стержня будуть такими, що величина його укорочення x_{c1} буде набагато меншою, ніж $x_{П1}$. Тому укороченням стержня при ударі можливо знехтувати і рахувати, що в перетворювачі (рис.1) деформується лише пружна пластина. Пружна пластина деформується від удару кулі вздовж стержня і переданої стержнем енергії.

Відомо, що рухоме тіло при пружному зіткненні з нерухомим тілом такої ж маси може передати останньому значну частину своєї енергії. При прямому ударі енергія передається повністю. Однак при пружних зіткненнях між тілами з більшою різницею мас обмін енергією ускладнюється. При цьому більш легке тіло відскакує від нерухомого важкого як від стінки, передаючи важкому лише незначну частину своєї енергії. Таку взаємодію тіл можливо описати більш точно математично. У перетворювачі (рис.1) куля рухається зі швидкістю v_k , а швидкість стержня $v_c=0$. Після пружного зіткнення відповідно закону збереження енергії можна записати

$$\frac{m_k v_k^2}{2} = \frac{m_k v_k'^2}{2} + \frac{m_c v_c'^2}{2}, \quad (2)$$

де m_c – маса стержня; v_k' , v_c' – швидкості руху кулі і стержня після пружного зіткнення.

У виразі (2) член $\frac{m_c v_c'^2}{2}$ являє собою кінетичну енергію, передану стержню на деформування пружної пластини. Член $\frac{m_k v_k'^2}{2}$ представляє енергію, що не передана падаючою кулею стержню в наслідок відскоку. Тому, нехтуючи деформуванням стержня, можна записати

$$m_k g (h + x_{П1}) = \frac{m_k v_k'^2}{2} + \frac{c_{П1} x_{П1}^2}{2}, \quad (3)$$

де v_k' – невідома величина.

Записавши закон збереження імпульсу і використавши вираз (2) можна визначити швидкість

$$v_k' = \frac{m_k - m_c}{m_k + m_c} v_1. \quad (4)$$

З іншого боку, робота, виконана падаючою кулею, відповідає кінетичній енергії і тому можна подати

$$A_P = \frac{m_k v_1^2}{2},$$

Звідки

$$v_1 = \sqrt{\frac{2A_P}{m_k}}. \quad (5)$$

З врахуванням виразу (1) при $x_{c1} = 0$, значень (5) за (4) можна записати

$$v_k' = \frac{m_k - m_c}{m_k + m_c} \sqrt{2g(h + x_{П1})}. \quad (6)$$

Підставивши v_k' (6) у (3) і перетворивши вираз, отримаємо

$$c_{П} x_{П1}^2 - 2m_k g \left[1 + \left(\frac{m_c - m_k}{m_k + m_c} \right)^2 \right] x_{П1} - 2m_k g h \left[1 + \left(\frac{m_c - m_k}{m_k + m_c} \right)^2 \right] = 0. \quad (7)$$

Розв'язавши рівняння (7) відносно $x_{П1}$, отримаємо

$$x_{П1} = \frac{m_k g \left[1 + \left(\frac{m_c - m_k}{m_k + m_c} \right)^2 \right]}{c_{П}} + \frac{\sqrt{m_k g \left[1 + \left(\frac{m_c - m_k}{m_k + m_c} \right)^2 \right] \left\{ m_k g \left[1 + \left(\frac{m_c - m_k}{m_k + m_c} \right)^2 \right] + 2c_{П} h \right\}}}{c_{П}}. \quad (8)$$

Залежність (8) показує рівень вихідного сигналу перетворювача при ударі стержня кулею, що падає з висоти h .

З врахуванням шматків руди, що руйнуються на торці стержня (рис.1), можна записати рівняння балансу роботи і енергії в перетворювачі

$$m_k g (h + x_{П2}) - m_k g (h + x_{П2}) \left(\frac{m_c - m_k}{m_k + m_c} \right)^2 - k_{MP} k_1 V_P = \frac{c_{П} x_{П2}^2}{2},$$

яке після перетворення можна подати у вигляді

$$c_{П} x_{П2}^2 - 2m_k g \left[1 + \left(\frac{m_c - m_k}{m_k + m_c} \right)^2 \right] x_{П2} - 2 \left\{ m_k g h \left[1 + \left(\frac{m_c - m_k}{m_k + m_c} \right)^2 \right] - k_{MP} k_1 V_P \right\} = 0 \quad (9)$$

де $x_{П2}$ – деформація пружної пластини; k_{MP} – коефіцієнт пропорціональності, що залежить від міцності руди; k_l – стала, що характеризує зв'язок між загальним і деформованим об'ємом шматка руди; V_p – об'єм шматка руди, що знаходився під дією кулі.

Розв'язавши рівняння (9) відносно $x_{П2}$, отримаємо

$$x_{П2} = \frac{m_k g \left[1 + \left(\frac{m_c - m_k}{m_k + m_c} \right)^2 \right]}{c_{П}} + \frac{\sqrt{m_k^2 g^2 \left[1 + \left(\frac{m_c - m_k}{m_k + m_c} \right)^2 \right]^2 + 2c_{П} \left\{ m_k g h \left[1 + \left(\frac{m_c - m_k}{m_k + m_c} \right)^2 \right] - k_{MP} k_l V_p \right\}}}{c_{П}} \quad (10)$$

Рівняння (10) отримано за умови, що маса кулі m_k значно менше маси стержня m_c . Встановлено, що в якості молоткового тіла в первинному перетворювачі доцільно використати кулю діаметром 50 мм [1]. Вихідний сигнал первинного перетворювача при прийнятій масі кулі m_k буде тим вищим, чим більша висота її падіння h , яка в певному типі млина також буде величиною незмінною. Сигнал первинного перетворювача також визначається масою стержня m_c . При незмінних цих параметрах вихідний сигнал перетворювача $x_{П2}$ є функцією об'єму руди V_p на внутрішньому торці стержня.

Первинний перетворювач працює за умови незмінної маси кулі m_k , що можливо забезпечити в кульових млинах. Недопустимість одночасного удару кількох куль в поверхню торця і незмінність розглянутих умов руйнування руди забезпечуються вибором діаметра стержня d_c , який приймається меншим порівняно з діаметром кулі d_k , наприклад $d_c \approx 0,7d_k$. Аналіз показав, що ефективно використання кулі $d_k=50$ мм при руйнуванні шматків руди розміром $d_p=(5...6)$ мм відповідають діаметру стержня близько 30...35 мм. При обертанні барабана млина кулі незмінної маси наносять удари в торець передаточного стержня, руйнуючи руду, яка на ньому в цю мить розташовується. За кількістю зруйнованої руди робиться судження про ефективність подрібнення матеріалу.

В процесі роботи кульового млина футеровка спрацювується, а разом з нею і стержень первинного перетворювача. Оскільки маса стержня m_c входить до математичного виразу статичної характеристики первинного перетворювача, при спрацюванні стержня (зменшенні m_c), буде вноситись похибка в результати вимірювання. Аналіз показав, що відносна похибка вимірювання може доходити до 13%. Ще одним джерелом виникнення похибки може бути нецентральний удар кулі по стержню. Ці похибки необхідно нейтралізувати. Теоретичне дослідження інваріантності первинного перетворювача до впливу експлуатаційних факторів розглянуте в роботі [2]. Запропонований первинний перетворювач з розширеною стержневою неробочою частиною і зміщеною масою володіє інваріантністю до дії згаданих факторів, оскільки володіє подвійним механізмом компенсації впливу експлуатаційного спрацювання стержня і нецентральних ударів на результати вимірювання. Це дозволяє звести похибку вимірювання параметра до 1,171%.

Вихідним сигналом первинного перетворювача є переміщення центральної частини пружної пластини. Обґрунтування типу вторинного перетворювача

виконано в роботі [3]. Встановлено, що достатньо ефективним є тензометричний вторинний перетворювач, особливо за умов збільшення чутливості в два рази і компенсації впливу температури навколишнього середовища. Слід відмітити, що ще одним засобом підвищення точності оцінювання параметра є здійснення тарування пристрою при умовно спрацьованій наполовину футеровці, коли буде допускаться систематична похибка у два рази менша порівняно з отриманою. Експериментальна перевірка засобу показала, що відносна похибка оцінювання зруйнованого об'єму руди не перевищує $\pm 3,0\%$, що не виходить за межі технологічних вимог. Отже, можливо стверджувати, що створено засіб прямого оцінювання ефективності роботи кульового млина безпосередньо в його барабані, що запропоновано в роботі [4].

Наукова новизна розробки полягає в тому, що вперше отримано статичну математичну модель, а на її основі створено інформаційно-вимірювальну систему оцінювання роботи кульових млинів безпосередньо у їх барабанах, що виводить інформаційне забезпечення складного технологічного процесу на якісно новий рівень.

Висновки. Показано, що первинний перетворювач «стержень-пружна пластина» з кулею діаметром 50 мм при діаметрі стержня 30...35 мм дозволяють здійснити пряме оцінювання об'єму зруйнованої маси руди безпосередньо в барабані технологічного агрегату. Точність оцінювання параметра підвищується забезпеченням інваріантності до впливу спрацювання стержня, нецентрального ударів кулі, а також використанням тензоперетворювачів за умов збільшення чутливості в два рази і компенсації впливу температури навколишнього середовища. Це дозволяє реалізувати інформаційно-вимірювальну систему оцінювання роботи кульових млинів з метою енергозбереження, відносна похибка якої не перевищує $\pm 3,0\%$.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Мацуй А. М., Кондратець В. О. Математичне моделювання автоматичного контролю енергоефективності подрібнення руди кульовими млинами при спрацюванні футеровки. *Проблеми інформаційних технологій*. 2018. №23. С. 111–117.
2. Мацуй А. М., Кондратець В. О., Абашина А. А. Теоретичне дослідження інваріантності первинного перетворювача енергоефективності подрібнення руди до впливу експлуатаційних факторів. *Science, research, development #14. Technics and technology: monogr. pokonf.* (London, 27–28 лют. 2019 р.). Warszawa, 2019. С. 9–18.
3. Мацуй А. М., Кондратець В. О. Обґрунтування вторинного перетворювача системи контролю енергоефективності кульового подрібнення руди оптимізацією, аналізом, моделюванням. *Прикладні питання математичного моделювання*. 2018. №2. С. 81–95.
4. Измельчение. Энергетика и технология / Пивняк Г.Г. и др. Москва: Изд. дом «Руда и металлы», 2007. 226 с.

ПЛАНУВАННЯ ШЛЯХУ ДЛЯ НАВІГАЦІЇ МОБІЛЬНИХ РОБОТІВ З УНИКНЕННЯМ ПЕРЕШКОД ЗА ДОПОМОГОЮ ДЕРЕВА ОКТАНТІВ

Анотація. Описано процес розробки удосконаленої системи планування шляху для навігації мобільних роботів та розпізнавання перешкод у реальному часі з використанням лазерів шляхом додавання камери, яка розраховує відстань до об'єктів. Удосконалена система була успішно реалізована та випробувана на мобільному роботі, забезпечуючи проходження шляху без зіткнень. Отримані результати моделювання представлені у роботі.

Ключові слова: ROS, навігація, мобільні роботи, уникнення перешкод, дерево октантів, лазер, хмара точок.

Вступ. Робототехніка з кожним роком розвивається все більше і більше. Створюються нові підходи для вирішення проблем руху, локалізації та автоматизації роботів. Багато моделей досягли значних успіхів у вирішенні різних проблем. Багато технічних комплексів створюється для військових цілей: виявлення цілей, її ліквідація. Створюються роботи-пожежники; рятувальні роботи, здатні витягнути людей з води, із завалів обвалених будівель. Однією з багатьох тенденцій робототехніки є перехід від систем з дистанційним управлінням, які вимагають постійної участі людини для виконання всіх дій робота, до автономних систем, в яких оператор вказує лише кінцеві та проміжні цілі.

Мобільні роботизовані системи сьогодні використовуються в найрізноманітніших галузях промисловості. Чим ширше сфера їх застосування, тим жорсткішими стають вимоги до їх виконання для конкретних завдань. Одна з найбільш нагальних таких вимог стосується автономності робота та його навігаційних можливостей [1].

Навігація залишається головною проблемою усіх існуючих мобільних пристроїв, що рухаються самостійно. Для успішної навігації в просторі бортова система робота повинна вміти планувати шлях, як стратегію пошуку шляху до цільового місця, правильно інтерпретувати інформацію про навколишній світ, отриману від датчиків. Більш того, робот повинен контролювати параметри руху, власні координати та мати можливість адаптуватися до змін навколишнього середовища.

Робототехніка з кожним роком розвивається все більше і більше. Створюються нові підходи для вирішення проблем руху, локалізації та автоматизації роботів. Багато моделей досягли значних успіхів у вирішенні різних проблем. Багато технічних комплексів створюється для військових цілей: виявлення цілей, її ліквідація. Створюються роботи-пожежники; рятувальні роботи, здатні витягнути людей з води, із завалів обвалених будівель.

Постановка задачі. Для досягнення поставленої мети в роботі формуються та вирішуються наступні завдання:

1. Розглянути основні принципи навігації.
2. Розглянути існуючі датчики та вибрати найбільш підходящий для даного мобільного робота.
3. Розглянути необхідні елементи ROS для навігації.
4. Використати метод зменшення роздільної здатності хмари точок і відсікти зайві елементи за допомогою PCL.
5. Удосконалити систему безперервного генерування дерева октантів з хмари точок, використовуючи Octomap, з подальшим проектуванням зайнятих клітин на підлогу.
6. Розробити пакет для перетворення отриманої проекції в багатокутники, а потім перетворити їх у перешкоди, відправивши до `teb_local_planner`.

Основний зміст роботи. Під час роботи над даною роботою були використані наступні методи та інструменти.

- SLAM (Simultaneous localization and mapping) algorithms;
- point cloud, voxel, octree, octomap;
- path planning algorithm;
- ROS tools and simulators (RViz, Gazebo, tf package, move_base, `teb_local_planner`);
- point cloud filtering with PCL.

Проаналізувавши сучасну роботу в галузі навігації мобільних роботів та уникнувши перешкод у режимі реального часу, було встановлено, що в сучасній теорії управління, дослідженнях робототехніки найбільш перспективними напрямками розвитку є використання алгоритмів, заснованих на обробці даних від датчиків.

Автори [3] зосереджуються на внутрішній системі локалізації робота на основі ультразвуку. Пропоноване рішення полягає у вимірюванні положення роботів, розрахованих за допомогою формул триангуляції.

Розроблений в [4] алгоритм дозволяє визначати положення роботів у будь-який час, використовуючи положення орієнтирів. Алгоритм враховує асинхронні кроки часу та неоднакові дані вимірювань для розробки своїх оцінок.

Одним з найпростіших засобів руху за вказаним маршрутом є одометрія, яка заснована на встановленні маршруту руху робота шляхом визначення руху коліс робота [5]. Сильним недоліком цієї техніки є накопичення помилок під час руху, тому доцільно використовувати її з іншими засобами навігації, щоб виправити цю помилку [6].

Існує два типи навігації мобільних роботів: 2D та 3D. 3D застосовується для літаючих роботів, коли темою цього дослідження є 2D-навігація. Робот, що є предметом дослідження, може рухатися лише за допомогою коліс у двовимірному просторі.

Сам ROS забезпечує реалізацію декількох 2D-навігаційних підходів, що називаються 2D-стеком навігації. Стек бере інформацію з датчиків, одометрії та пози цілі та видає команди швидкості для безпечного переміщення мобільної

бази. Як необхідна умова використання стека навігації, робот повинен запускати ROS, мати на місці дерево перетворення tf та публікувати дані датчиків, використовуючи правильні типи повідомлень ROS. Крім того, стек навігації повинен бути налаштований для форми та динаміки робота, щоб він працював на високому рівні.

Реалізація дії передбачена в пакеті `move_base`. Поставленої мети у світі потрібно досягти за допомогою мобільної бази. Глобальний та локальний планувальник пов'язані між собою вузлом `move_base` для виконання свого глобального навігаційного завдання. У пакунку `nav_core` вказано інтерфейс `nav_core:BaseGlobalPlanner`, і глобальний планувальник прикріплений до нього. Така сама ситуація з будь-яким місцевим планувальником. Будь-який локальний планувальник, який дотримується інтерфейсу `nav_core::BaseLocalPlanner` із пакета `nav_core`.

Вузол підтримує дві карти витрат - від глобального та місцевого планувальників, які використовуються для виконання навігаційних завдань.

`Timed Elastic Band` оптимізує траєкторію робота з локальним часом виконання, відокремленням від перешкод та дотриманням кінодинамічних обмежень під час роботи. Пакет `teb_local_planner` виконує плагін до `base_local_planner` 2D-стека навігації.

Цей оптимальний локальний планувальник траєкторій руху для управління мобільним роботом та навігації реалізований цим пакетом як плагін для навігаційного пакету в ROS.

Початкова траєкторія, яка генерується глобальним планувальником, оптимізована під час виконання. Як результат, мінімізація часу виконання траєкторій, відокремлення від перешкод, дотримання кінодинамічних обмежень, таких як задоволення максимальних прискорень і швидкостей.

Поточна реалізація відповідає кінематиці неголономних роботів, таких як диференціальний привід та роботи, схожі на автомобіль. Підтримка голономних роботів включена з `Kinetic`.

Досліджуваний робот має два типи датчиків: площинні лазерні сканери та глибинні камери.

Лазерний `SICK S300` використовується як основний сканер середовища для навігації. Це лазерний сканер з кутом сканування 270 та кутовою роздільною здатністю 0,5.

Інший тип датчика – це камера глибини. Камера глибини представлена камерою `Intel Realsense D435`. Використання подвійної установки камери забезпечує глибоке зображення, яке можна перетворити на точкову хмару. Це перетворення здійснюється вузлом камери ROS `Realsense`. Коли камера буде налаштована належним чином в описі робота в тому самому місці, що і в реальному світі, вузол буде надавати дані хмарних точок відносно робота. Якщо робот добре розташований, дані також будуть відносно світу.

Щільність наданої хмари необроблених точок занадто велика. Його можна сміливо знімати до різко меншої щільності. Це робиться за допомогою фільтра `VoxelGrid`, наданого `PCL`.

PCL також пропонує ще один дуже корисний фільтр під назвою PassThrough, який використовується для вирізання занадто низьких і занадто високих даних (підлога та стеля), які не слід розглядати як перешкоду. І ми вирізаємо занадто близькі предмети, які є частинами самого робота, як правило, це рука з'являється в поле зору камери.

SLAM - одночасна локалізація та картографування - це підхід до обробки декількох кадрів та позицій датчиків для побудови об'ємної моделі середовища.

Візуальний шлем може виконуватися навіть за допомогою однієї камери (монокуляр). Це дешево і просто. Глибина не доступна повністю з одного зображення; натомість нам потрібні декілька зображень, щоб відповідати їм і обчислити диспропорцію.

З двома камерами це ще простіше - маємо фіксовану відстань між камерами. Маючи невідповідність, можна встановити відстань від камери для кожної точки і на її основі побудувати тривимірне зображення зображення для поточного кадру.

Весь підхід SLAM передбачає створення повної карти. SLAM може використовувати декілька різних типів датчиків, а потужності та обмеження різних типів датчиків стали важливим рушієм нових алгоритмів. Статистична незалежність необхідна для того, щоб впоратися з метричним зміщенням та шумом у вимірах. Дані різних типів датчиків можуть оброблятися різними алгоритмами SLAM, підходи яких більш сумісні з датчиками. Лазерні сканери або візуальні функції надають деталі багатьох точок в межах області, іноді надання висновку SLAM є непотрібним, оскільки фігури в цих хмарах точок можна швидко і однозначно вирівняти на кожному кроці за допомогою реєстрації зображень. На протилежній крайності тактильні датчики занадто розріджені, оскільки містять лише інформацію про точки, дуже близькі до агента. Отже, вони потребують потужних моделей для компенсації в чисто тактильному SLAM. Більшість застосовних завдань SLAM знаходяться десь між цими візуальними та тактильними сферами. Можна використовувати одометрію, GPS та локалізацію на основі спільних кутів для локалізації на основі зображень камери.

Дерево октантів - це дерево, в якому кожна вершина має по вісім дітей. Деревя октантів найчастіше використовуються для поділу тривимірного простору шляхом рекурсивного розділення на октанти. Деревя октантів - це ієрархічні деревоподібні структури, які описують кожну область тривимірного простору як вузли.

Зазвичай дерева октантів широко використовуються в інформатиці. Вони мають загальні структури даних, тому вони є відповідним способом зберігання інформації в області на непараметризованих сітках.

Бібліотека OctoMap реалізує підхід до відображення сітки в 3D, забезпечуючи структури даних та алгоритми відображення. Реалізація карти базується на дереві октантів. Потім ці моделі можна використовувати для навігації та уникнення перешкод.

OctoMap забезпечує перетворення даних хмар точок в дерева октантів сітки. Потім ці октанти (воксели) проєктуються на підлогу, забезпечуючи зображення окупованої площі.

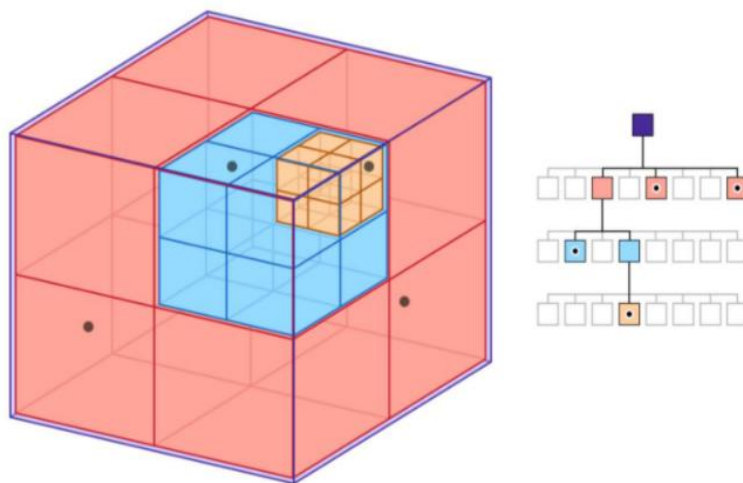


Рис. 1. Дерево октантів на тестових даних

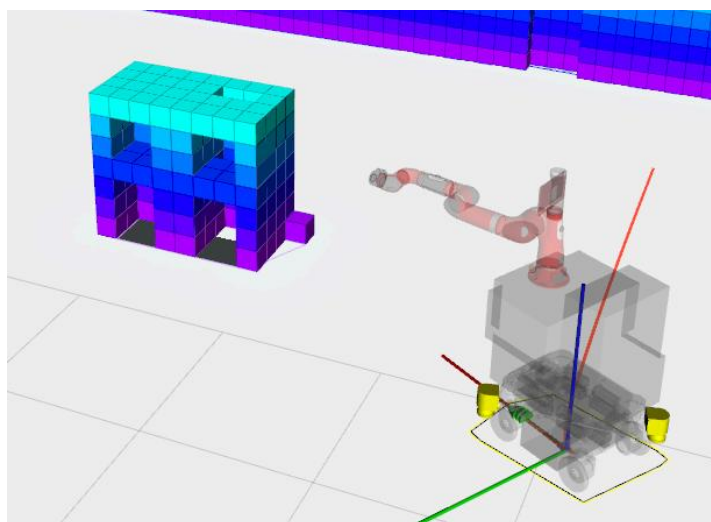


Рис. 2. Дерево октантів збудоване на розпізаному об'єкті

Але сама проєкція не є перешкодою для навігаційного стека. Найбільш підходящим введенням з перешкодами для стеку навігації з обходом перешкод, є `teb_local_planner`. Він може сприймати перешкоди як точкові перешкоди або багатокутники.

По-перше, було розроблено пряме перетворення із зайнятих пікселів у точкові перешкоди. Це рішення не забезпечує необхідного планування шляху, оскільки `teb_local_planner` намагається побудувати маршрут через перешкоду між близькими точками, де недостатньо місця.

Кращий спосіб - побудувати на проєкції багатокутники навколо окупованих територій.

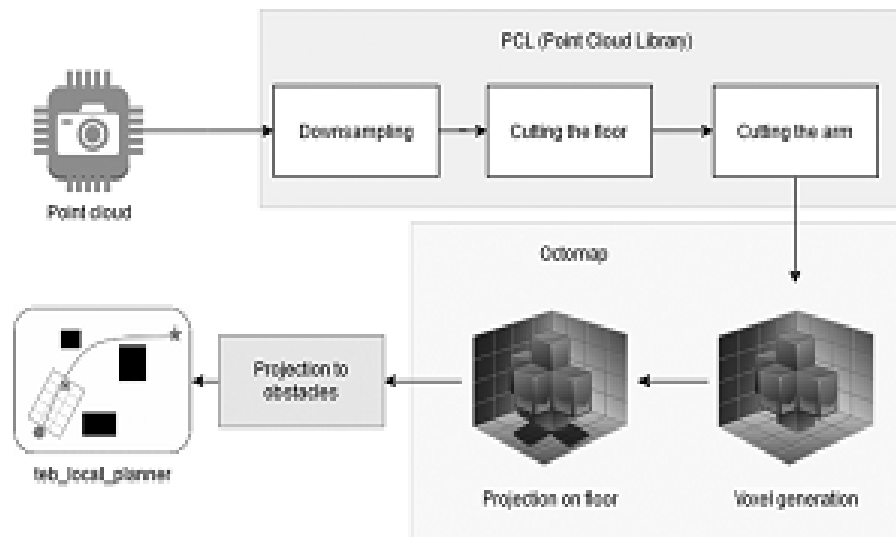


Рис. 3. Схема дерева октантів для навігації

У цьому випадку всередині окупованої території немає невеликого простору, і шлях прокладений належним чином навколо перешкоди.

Таким чином, мобільний робот зможе розпізнати та уникнути перешкоди, яку виявили на шляху від вихідної точки до цілі. Мобільний робот змінить маршрут, який був побудований, відповідно до знайденої перешкоди.

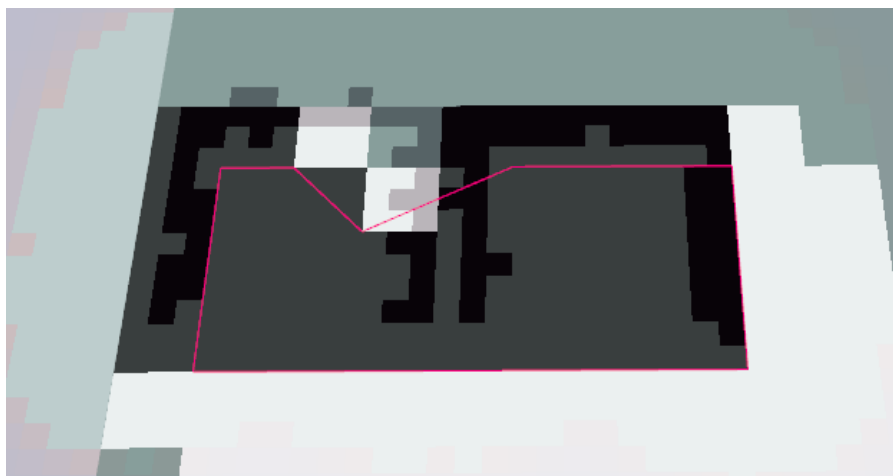


Рис. 4. Проекція зібрана у полігони

Наукова новизна полягає у тому, що удосконалено методи системи планування шляху для навігації мобільних роботів на основі дерева октантів для якісного та точного шляху від початкової точки до заданої у просторі.

Висновки. Результатом роботи, представленої вище, стало покращення системи розпізнавання об'єктів на шляху робота, які знаходяться не на одному рівні з лазерами мобільного робота. Вище були розглянуті різні існуючі методи та алгоритми навігації. Також описані етапи дослідження та отримані результати. В результаті використання удосконаленої системи з використанням камери Realsense було виявлено, що така система є кращою та ефективнішою, ніж використання лише лазерів.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Бобровский С. Навигация мобильных роботов (в 3 ч.). Ч. 1 // PC Week /RE. 2004. №9. С.52.
2. Минин А. Навигация и управление мобильным роботом, оснащенным лазерным дальномером: диссертация кандидата технических наук 05.02.05/А.А. Минин; Москва, 2008. 182с.
3. Nagy C. Ultrasound-Based Indoor Robot Localization Using Ambient Temperature Compensation/C. Nagy, Z. Biró-Ambrus, L. Márton// Acta Universitatis Sapientiae Electrical and Mechanical Engineering. – Vol. 8. – 2016. – pp. 19-28.
4. Petrovski K. On the Kalman Filter Approach for Localization of Mobile Robots / K. Petrovski, S. Jovanovski, M. Mirchev, L. Basnarkov // International Conference of ICT Innovations. – 2016. – pp. 123-133.
5. Chong K.S., Kleeman L. Accurate Odometry and Error Modelling for a Mobile Robot, MECSE-1996-6.
6. Михайлов Є.П. Локальна навігація мобільних роботів з використанням засобів одометрії/ Михайлов Є.П., Крись М.В., 2015.
7. Зенкевич С.Л., Назарова А. В., Лисицын Д. Моделирование движения мобильного робота по сложному маршруту // Материалы Науч. школы-конференции «Мобильные роботы 2000». - М.: Изд-во Моск. ун-та, 2000. - С. 14-27.
8. Mohd Nasir N.Z. Autonomous mobile robot localization using Kalman filter/N. Z. Mohd Nasir, M. A. Zakaria, S. Razali, M.Y. bin Abu/ MATEC Web of Conferences. – Vol. 90. – 2017.
9. Siegwart, R., Ourbakhsh, I. (2004). Introduction to Autonomous Mobile Robots. Boston: MIT Press, 336.

УДК 004.415.3:681.6

Л.І. Мещеряков¹, М.М. Одновол¹, Н.Л. Гресь¹, Д.Д. Массалітін¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ЗАСТОСУВАННЯ МЕТОДУ ЛОГІСТИЧНОЇ РЕГРЕСІЇ ДЛЯ КЛАСИФІКАЦІЇ НАУКОВИХ ТЕКСТІВ У НАВЧАЛЬНОМУ АСИСТЕНТІ

Анотація. Описано процес створення освітнього бота-асистента, що класифікує й рекомендує тексти користувачу залежно від його пріоритетів та прогресу за темами. У ході роботи було проаналізовано й порівняно сучасні підходи до обробки природної мови, в результаті чого було використано логістичну регресію для задачі класифікації текстів.

Ключові слова: класифікація текстів, логістична регресія, машинне навчання, обробка природної мови, NLP, SVM, word2vec, парсинг, Python, PostgreSQL, scikit-learn, word embedding.

Вступ. Тема класифікації текстів й інформаційних джерел надзвичайно важлива сьогодні, так як ми живемо в інформаційну епоху, для якої є

характерним інформаційний простір, переповнений інформацією різного типу та якості, від художніх творів й новин до наукових праць. Хоча вільний доступ до широкого різноманіття публікацій є безсуперечною перевагою, разом з цим ця обставина кидає свої виклики людині.

Студенту або науковцю у всі часи було надзвичайно важливо постійно оновлювати свої знання, пізнавати й вивчати нове, на жаль, у вируючому інформаційному потоці буває важко сконцентруватись на одній темі й досконально вивчити її. Тож технології класифікації текстів набувають все більшої актуальності й особливо очевидна їх важливість у часи пандемії, коли навчання та робота в основному перейшли у дистанційний формат. Як окремим людям, так бізнесу й науковим установам важливо точно й ефективно класифікувати й пріоритизувати інформацію, що надходить, у цьому їм допомагають інструменти машинного навчання та статистики.

Дана робота ставить за мету наглядно продемонструвати на прикладі навчального бота-асистента, що навіть давно відомі методи класифікації тексту, зокрема логістична регресія, сьогодні актуальні й можуть успішно вирішувати практичні завдання.

Постановка задачі. Для виконання поставленої задачі у ході роботи були сформульовані й вирішені наступні завдання:

- описати принцип роботи й будову моделей для класифікації текстів;
- проаналізувати й порівняти методи обробки природної мови й представлення текстової інформації для задач класифікації;
- проаналізувати й порівняти моделі класифікації текстів;
- дослідити особливості комбінування методів представлення текстової інформації й моделей-класифікаторів;
- обґрунтувати використання логістичної регресії для задач класифікації текстів;
- створити навчального асистента, що базується на моделі логістичної регресії для класифікації текстів, скомбінованої з методом ранжування.

Основний зміст роботи. У ході роботи було використані наступні методи та інструменти:

- методи напряму NLP та машинного навчання;
- TF-IDF як вхідний параметр для моделі-класифікатора;
- метод логістичної регресії для класифікації текстів;
- Python, scikit-learn, nltk, VSCode, PostgreSQL, request, BeautifulSoup.

Задача класифікації текстів, яка вирішується в ході створення навчального бота-асистента, відноситься до області NLP (Natural language processing, обробки природної мови). Це завдання включає в себе 3 етапи: збір, обробка та підготування вхідних й навчальних даних, обрання придатної моделі класифікації й створення серверного додатку, що буде отримувати й обробляти запити від користувачів.

Перш ніж детально описувати процес підготовки даних потрібно розглянути наявні моделі класифікації. Сьогодні існує багато методів розпізнавання текстів, вони діляться на 2 великі групи: базовані на правилах та

на моделях машинного навчання (евристичний підхід). У кожного з цих підходів є свої переваги та недоліки.

Базований на правилах підхід до класифікації текстів полягає у тому, що лінгвісти аналізують стилістичні особливості властиві визначеним класам текстів, закономірності та лінгвістичні конструкції мови, з яких пізніше формуються формалізований набір правил. Цей підхід може мати більшу точність порівняно з евристичним, за умови, що формат та різновид текстів чітко окреслений. Однак недоліком цього методу є те, що його використання дещо обмежене за розміром тексту та його структурою, тож цей підхід може невірно працювати на текстах незвичного формату чи структури. Окрім цього, попередній досвід використання цього підходу показав, що систему базовану на правилах важко розширяти за кількістю категорій, адже кожний випадок потребує повторного лінгвістичного аналізу та створення нових правил, які б не суперечили старим. Отже, сьогодні цей підхід виглядає малоперспективним й не представляє особливого інтересу для комп'ютерних наук.

Евристичний підхід працює на основі імовірнісних моделей машинного навчання, які у свою чергу підрозділяються за типом навчання на 2 групи: з вчителем та без. Основна відмінність між цими 2 принципами полягає у тому, що модель з вчителем включає в себе етап навчання, який потребує прикладів вхідних даних та правильних відповідей. Під час цього етапу відповіді моделі порівнюються з наданими прикладами й на основі цього «вчитель» регулює коефіцієнти моделі доки не буде досягнута бажана точність. Прикладами моделей з вчителем можуть бути такі моделі як наївний Баєс, метод опорних векторів, згорткові та рекурентні нейронні мережі, логістична регресія тощо. Моделі без вчителя у свою чергу намагаються самостійно віднайти взаємозв'язки та закономірності між наданими об'єктами й розбити їх на групи за спільними ознаками. До моделей без вчителя можуть бути віднесені моделі на основі латентно-семантичного аналізу (ЛСА), методу к-середніх та інші. Хоча між цими методами є істотні відмінності, вони мають ряд загальних переваг та недоліків. Перевагами цих методів являється те, що їх можна порівняно легко розширити новими класами та регулювати їх параметри. Окрім цього, евристичні методи здатні відносно точно класифікувати незвичні чи невідомі їм тексти й не потребують активного втручання спеціалістів предметної області при кожному новому випадку. До недоліків цих методів можна віднести наступні фактори: їм бажано мати велику вибірку для навчання чи кластеризації та потрібно уважно підходити до конструювання та вибору ознак класифікації. На додачу до цього, для моделі з вчителем важливо уникати перетренування, щоб вона могла коректно класифікувати тексти поза навчальною вибіркою. Сьогодні, саме евристичні моделі в основному вирішують завдання класифікації та відбору в різноманітних сервісах.

У даній роботі розглядаються лише моделі класифікації текстів з вчителем з декількох причин. По-перше, результати моделей без вчителя можуть бути спірними й неочевидними, по-друге, в роботі важливо щоб класи мали зрозумілі для користувача імена, модель же без вчителя просто розбиває множину об'єктів на умовні групи без осмислених назв, по-третє, моделі з

вчителем більш популярні та надійні, частково з причини, що їх похибку можна отримати під час навчання. Отже, для даної роботи підходять лише моделі класифікації з вчителем.

Одним з найбільш популярних та простих методів є метод наївного Баєсу, він, наприклад, часто використовується для фільтрації спаму. У основі цього методу лежить теорема Баєсу, яка дозволяє точніше визначити ймовірність події, якщо до цього сталася інша статистично пов'язана з нею подія. Принцип цього методу полягає у тому, що він працює з вірогідністю появи ключових слів для кожного класу тексту, тобто у даному випадку приналежність тексту до одного з класів статистично залежна подія від набору слів. В результаті для кожного класу вираховується ймовірність що текст з таким набором слів належить до нього, в результаті обирається клас з найбільшою вірогідністю. Найбільша перевага цього алгоритму, що він простий та ефективний коли вхідні дані й обчислювальні ресурси обмежені, так як цьому метод не потрібно робити множинні ітерації задля оптимізації на відміну від регресії й нейронних мереж. Однак, ця модель має й недоліки, точність цієї моделі задовільна, але гірше ніж в інших моделях, також ця модель рано досягає ліміту навчання, тобто при досягненні деякого порогу нові приклади для навчання не дають покращення точності моделі. Інший відомий підхід це моделі глибинного навчання (Deep Learning). Глибинне навчання – це концепція, що за допомогою математичної абстракції нейронної мережі симулює роботу людського мозку, а саме процес отримання інформації та прийняття рішень. По суті, нейрона мережа - це зважений граф, кожне ребро у якому має різну вагу, у цій моделі вузол моделює нейрон, а його ребра виконують роль дендритів та аксонів, сигнал, що передається між нейронами – це результат активаційних функцій нейронів помножений на вагу відповідного ребра. З одного боку, алгоритми глибинного навчання досить складні, вимагають багато часу для побудови, великих об'ємів даних, а отже й обчислювальної потужності для ефективного навчання. З іншого боку, нейронні мережі показують найкращі результати серед моделей, можуть мають досить високий поріг навчання, завдяки чому датасет можна довго нарощувати й покращувати показники. З цих причин, нейронні моделі зараз найбільш популярні у великих рішеннях, що потребують високої точності та класифікують велику кількість класів.

Третя популярна модель – це логістична регресія. Логістична регресія – це лінійна модель, сигмоїдальна функція, що обчислює вірогідність деякого випадку залежно від вхідних даних. Принцип логістичної регресії у тому, що вона приймає параметри тексту, такі як частота чи ймовірність появи слова у тексті, й потім за допомогою градієнтного спуску коефіцієнти функції регулюються задля отримання оптимальної точності. Хоча логістичну регресію часто порівнюють з методом наївного Баєсу, принцип роботи у цих моделей істотно відрізняється. Наївний Баєс – це генеративний класифікатор, тобто він намагається визначити наскільки вхідні параметри властиві одному з класів, а логістична регресія – це дискримінативний класифікатор, він шукає відмінності між класами й проводить умовну межу між об'єктами. Завдяки таким особливостям логістична регресія показує краще результати на великих об'ємах

даних, у той час як Баєс підходить для простих невеликих задач чи для випробовування інших ознак класифікації.

Остання у переліку модель – це опорно-векторні машини (SVM від Supported Vector Machines). SVM - це набір алгоритмів, що базуються на методі опорних векторів. Ці алгоритми можуть використовуватись як для регресійного аналізу, так й для задач класифікації. Цей класифікатор часто порівнюють з логістичною регресією, тому що обидва класифікатори лінійні, тобто в результаті роботи вони умовно ділять простір за допомогою гіперплощини (у випадку 2-вимірного простіру це просто лінія). Відмінність лише у методах, якими досягається ця мета. Логістична регресія поступово зменшує показник врат до потимального значення, а SVM прагне провести таку гіперплощину, що максимально далека від обох класів. У випадку коли класів більше SVM відділяє один клас від інших й проводить більше гіперплощин.

У ході аналізу було виявлено, що логістична регресія добре підходить для класифікації текстів, при правильному підборі параметрів регресія може бути непоганою альтернативою іншим більш складним методам, тому що її результати простіші для інтерпретації й вона менш вимоглива до об'ємів тренувальних та навчальних вибірок, ніж глибинні нейронні моделі. У порівнянні з іншими лінійними класифікаторами такими як Баєс та SVM, логістична регресія краще підходить для задач класифікації з двох причин. По-перше, логістична регресія краще працює з великими об'ємами ніж Баєс, у неї також вище поріг навчання. По-друге, хоча SVM може давати трохи точніші результати, але логістична регресія на відміну від SVM дає ймовірнісну відповідь, у випадку класифікації текстів це корисно тим, що у тексті можуть бути декілька взаємопов'язаних тем, тож у цьому випадку доречніше віднести до декількох категорій.

Після обрання моделі потрібно обробити вхідні та навчальні дані й визначити ознаки для прогнозової моделі. Ознаки – це числові параметри тексту, які дозволяють моделі робити прогнози.

У випадку класифікації текстів вхідною ознакою може бути частота появи слів у тексті, проте ця ознака досить не точна й може призводити до низької точності самого класифікатора. Для класифікації рекомендується використовувати показник TF-IDF (TF — частота терміну, IDF — інверсивна частота у документі). TF-IDF – це показник, що прямо пропорційний кількості вживань терміну у тексті, але обернено пропорційний частоті появи терміну у текстах з колекції. Це означає, що TF-IDF дає низьку вагу розповсюдженим словам, що трапляються у багатьох текстах, але високу для термінів, що характерні лише для окремих класів текстів. Отже, цей статистичний показник добре підходить для задач класифікації.

Використана модель не буде враховувати синтаксичні особливості англійської мови й пунктуацію, вона базується лише на використаних словах, тож у нашому випадку наступні тексти будуть оброблятися наступним чином:

- видаляються розповсюджені, неінформативні слова;
- прибираються пунктуаційні знаки, спеціальні символи та номери, так як вони не несуть ніякого смислу

- усі слова зводяться до нижнього регістру й лемматизуються (зведені до словникових кореневих форм за допомогою пакету NLTK).

Наступним кроком у розробці моделі-класифікатора після обрання моделі та ознак для класифікації – це створення навчальної вибірки. Як основне джерело навчальних даних було обрано інтернет енциклопедію Wikipedia, так як вона має широкий вибір наукових текстів, які впорядковані за темами, що облегшує пошук й відбір даних для навчання. Окрім цього, усі статті структуровані, що спрощує їх обробку та отримання з них корпусу слів. За допомогою бібліотек для запитів й парсингу request та BeautifulSoup було зібрано та оброблено 100 статей за темами економіка, історія, математика та IT, у процесі їх обробки були видалено математичні формули, зображення й HTML теги. Усі 100 були рівномірно розподілені між 4 темами. Загальний розмір вибірки становить приблизно 1 000 000 слів. Далі навчальна вибірка розбивається на 2 частини – навчальну й тестову, навчальна частина становить 70 текстів, а тестова – 30 відповідно, це потрібно для того щоб перевірити точність моделі. Після навчання модель класифікує тестовий набір й видає точність класифікації 97% на тестовому наборі розміром 30 текстів. Отримана модель була потім використана у боті, що дозволило випробувати її у реальних умовах. Модель вдало класифікувала тексти з цих 4 тем у 90% випадків, навіть з джерел раніше невідомих для неї.

Наукова новизна:

- вдосконалено та розширено можливості методу логістичної регресії завдяки комбінуванню її з методом ранжування;
- отриману модель логістичної регресії було вдало інтегровано у навчальний бот-асистент;

Висновки. В ході виконання роботи було проаналізовано й розглянуто основні методи обробки природньої мови з метою створення навчального бота асистента. В результаті, було виявлено, що при правильному підборі ознак (зокрема завдяки використанню TF-IDF показника) та параметрів навіть проста модель на основі методу логістичної регресії здатна досить успішно використовуватись у завданнях класифікації текстів, при цьому потребуючи менше ресурсів й часу ніж більш складні моделі. Завдяки цьому, отриманий класифікатор на основі логістичної регресії, поєднаний з методом ранжування, було використано у навчальному асистенті, де він продемонстрував високу точність.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Artetxe M., Labaka G., Agirre E. (2019). An Effective Approach to Unsupervised Machine Translation / M. Artetxe, G. Labaka, E. Agirre // ArXiv. – Vol. 1902 – 2019 – P. 10.
2. Bahdanau D., Cho K., Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate. / D. Bahdanau, K. Cho, Y. Bengio// ArXiv. – Vol. 1409 – 2014 – P. 15.
3. Jurafsky D. Speech and Language Processing. An Introduction to Natural Language Processing. Computational Linguistics, and Speech Recognition 3rd

edition draft/ D. Jurafsky, J. H. Martin - Stanford University - Stanford University Press, 2019 – 621 P.

4. Kavita G., Subotin M. (2015). A general supervised approach to segmentation of clinical texts. Proceedings. / G. Kavita, M. Subotin // 2014 IEEE International Conference on Big Data, IEEE Big Data 2014 – P. 33 – 40.

5. Ng A. Y., and Jordan M. I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. / A. Y. Ng, M. I. Jordan // Conference on Neural Information Processing Systems – Vol. 14 - 841–848 P.

УДК 004.891

Б.І. Мороз¹, В.В. Спірінцев¹, Д.С. Таргунаков¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ІНФОРМАЦІЙНА ПІДТРИМКА ПРИЙНЯТТЯ РІШЕНЬ В КОМПЛЕКСНИХ ТЕХНОЛОГІЧНИХ ПРОЦЕСАХ ВИРОБНИЦТВА

Анотація. Був виконаний аналіз щодо інформаційної підтримки комплексних виробничих процесів. Запропоновані шляхи щодо підвищення ефективності виробництва за рахунок впровадження системи підтримки прийняття рішень.

Ключові слова: система підтримки прийняття рішень, автоматизовані системи управління, системи диспетчерського управління та збору даних, програмовані логічні контролери, розподілені системи управління, *online analytical processing*

Вступ. В сучасному світі, такі чинники, як глобалізація торгівлі та запекла конкуренція, сприяють збільшенню вартості помилок на підприємствах. Будь-які помилки – це фінансові втрати, тобто зменшення прибутку. А коли діяльність підприємства пов'язана з виробництвом, то помилки є причиною бракованої продукції. Також помилки призводять до перевитрат енергетичних та матеріальних ресурсів і, навіть, до аварій.

Тому на виробництві ціна диспетчерського рішення зазвичай дуже велика, що психологічно тисне на диспетчера. Складність прийняття рішення зростає при відсутності або недостовірності частини параметрів. Крім того, з плином часу тенденція погіршення ситуації може наростати, в зв'язку з чим різко зменшиться резерв часу на прийняття рішення. Це взагалі може бути однією з причин аварії, у зв'язку з прийняттям або не прийняттям управлінських рішень. Таким чином, в збільшенні ступеня технологічного ризику і зниження виробничої безпеки, а також ефективності в системі «людина - машина» істотну роль відіграє людський фактор.

Постановка задачі. Для досягнення мети підтримки прийняття рішень в роботі були сформовані і вирішені наступні завдання: викладені принципи та методи систем підтримки рішень в комплексних технологічних процесах; обрано джерело вхідних даних для системи; визначені потрібні функціональні

можливості системи щодо візуалізації технологічного процесу, адресної доставки інформації про порушення та додаткової інформації про шляхи виправлення; визначені вимоги щодо звіту у вигляді OLAP куба; спроектована та розроблена програмна частина.

Основний зміст роботи. Для випуску якісної продукції та ефективної витрати енергетичних та матеріальних ресурсів, важливо швидко приймати рішення, пов'язані з виявленням помилок роботи технологічного процесу та своєчасним їх виправленням. Тобто, якщо підприємство має можливість впровадити у себе системи для надання інформаційної допомоги працівникам, відповідальним за прийняття рішень, то воно забезпечить собі більш ефективні умови праці [1].

Такі системи зазвичай виступають в ролі помічника, який дозволяє розширити можливості людини, але не замінює її. Системи підтримки прийняття рішень (СППР) призначені для використання в ситуаціях, коли процес прийняття рішень через необхідність урахування суб'єктивної думки не може бути повністю формалізований і реалізований на електронній обчислювальній машині (ЕОМ) [2]. Тобто, коли людино-технічна система функціонує в складних умовах.

Під складними умовами мається на увазі великий масштаб об'єкта управління, його нестационарність, довготермінова динамічна пам'ять, розподіленість параметрів, нелінійність, значні запізнення керування, неможливість контролю зовнішніх впливів, різноманіття ситуацій, мінливість цілей, критеріїв і т.д. За рівнем допомоги, ці системи можуть надавати пасивну, активну або комбіновану підтримку особі, яка приймає рішення (ОПР).

При використанні пасивного підходу, СППР в інтерактивному режимі за допомогою зручного інтерфейсу пропонує вибрати певні варіанти вирішення, оцінити їх в процесі вибору. Така система не здатна генерувати нові варіанти, але здатна використовувати базу знань, яка вже створена експертами.

Активна система підтримки передбачає використання аналітичних методів і багатокритеріального аналізу. В цьому випадку система підтримки аналізує, пояснює дане рішення, обґрунтовує переваги і втрати, якщо вони можуть бути. ОПР все це осмислює і на основі пропозиції приймає рішення, знаючи про наслідки, до яких воно може привести.

Комбіновані системи підтримки прийняття рішень забезпечують ітеративний процес взаємодії між ОПР та системою. Тобто, оператор має можливість змінювати, доповнювати або уточняти пропозиції та відправляти їх назад на перевірку. Даний процес триває доти, доки не буде отримане оптимальне рішення.

Бази знань і бази даних в СППР є ядром системи і надають інформацію для підтримки рішення. Цією інформацією активно користується особа, яка приймає рішення. Також ОПР може використовувати звіти, документи, технологічні інструкції.

СППР, враховуючи складність завдань, діляться на три класи. Системи підтримки першого класу використовують великі компанії, коли необхідно реалізувати дуже багато функціональних можливостей. Такі системи служать

для обґрунтування рішень, які повинні бути включені в план дій, щоб досягти кінцевої мети програми. Зазвичай такі системи припускають колективне використання, тому бази знань для таких СППР формуються фахівцями різних областей. Система підтримки прийняття рішень другого класу передбачає самостійне формування баз знань користувачем, який сам користується цією системою. Таким користувачем може бути адміністратор або службовець невеликої організації. Системи підтримки третього класу, це системи індивідуального використання, які адаптуються до досвіду користувача. Такі системи забезпечують отримання рішення поточної задачі, базуючись на результатах, прийнятих користувачем в минулому.

В сучасному світі, автоматизовані системи управління (АСУ) можуть стати джерелом даних для систем, які будуть надавати підтримку в прийнятті рішень та функціонувати як єдине ціле на виробничій лінії в процесі виробництва. Це дозволить отримувати дані для моніторингу та пошуку порушень у роботі технологічного процесу. Одним з напрямків розвитку АСУ технологічними процесами є диспетчерське управління, організоване в рамках галузевої системи оперативно-диспетчерського управління. В ієрархії управління підприємства, диспетчерській службі відводиться важлива функція по інтегруванню управління всіма технологічними об'єктами, що забезпечують процес виробництва.

Існує безліч типів систем управління технологічними процесами, включаючи системи диспетчерського управління та збору даних (Supervisory Control And Data Acquisition – SCADA), програмовані логічні контролери (Programmable Logic Controller – PLC) або розподілені системи управління (Distributed Control System - DCS), і вони працюють для збору і передачі даних, отриманих під час виробничого процесу [3].

Сучасні системи автоматизації виробництв повинні задовольняти пропонованим до них вимогам в умовах необхідності прийняття відповідальних рішень в обмежені терміни. Системи оперативного диспетчерського управління технологічними процесами повинні мати у своєму розпорядженні потенціали для виявлення та діагностики позаштатних і аварійних ситуацій на основі методів підтримки прийняття рішень [3]. Тобто, на основі вищесказаного можливо сформулювати вимоги для інформаційної системи підтримки рішень для комплексних технологічних процесів.

Весь технологічний процес комплексного виробництва повинен бути добре візуалізований. Він повинен відображати поточні показники роботи основних технологічних параметрів, колірну індикацію ключових процесів при виході із заданих діапазонів. Це зробить технологічний процес прозорим і дозволить отримувати додаткову інформацію керуючому персоналу.

Так як в комплексному виробництві за технологічний процес відповідають як правило декілька осіб, повинний бути механізм, який відстежує технологічні параметри в автоматичному режимі і в разі, якщо параметри виходять за рамки допустимого, розсилати оповіщення. Тобто отримувати їх буде тільки той, хто прямо або побічно відповідає за усунення. З впровадженням СППР, персонал буде отримувати не тільки оповіщення про відхилення, але також і інформацію

про способи вирішення, додаткову інформацію про проблему. На даному етапі інформація про виконані дії повинна бути зафіксована. Надалі, вона буде потрібна для зведеного звіту.

Остаточним кроком є зведений звіт у вигляді OLAP куба, який надасть можливість отримувати інформацію про роботу технологічного процесу, а також дій персоналу. Що дозволить виявляти прийняття неефективних рішень.

Звіт повинен надавати інформацію про порушення технологічного процесу по кожній технологічній ділянці і містити наступну інформацію: місце виникнення тривоги; причини виникнення тривоги; виконані дії; кількість порушень у цілому, а також по окремим ділянкам.

Наукова новизна полягає в удосконаленні методів інформаційної підтримки прийняття рішень при управлінні комплексним виробництвом.

Висновки. В результаті виконання роботи були визначені функціональні можливості та запропоноване програмне забезпечення, яке можна використовувати для оптимізації процесу прийняття рішень в комплексних технологічних процесах управління, які не можуть бути повністю формалізованими.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Daouda K. Decision Support Framework for Crisis Management / Daouda K., Pascale Z., François P. A // EWG-DSS 2013 - Thessaloniki. 2013, P. 1-10.
2. Uday K. Decision Support for Global Enterprises / Uday K., Daniel J.P., Ramesh S. - Springer, 2007, P. 281.
3. Timothy S. A Conceptual Model for Operational Control in Smart Manufacturing Systems / Timothy S., Leon F. M. // IFAC-PapersOnLine, Vol. 48, No. 3, 2015. P. 1865-1869.
4. John T. W. Intelligent Building Control Systems / John T. W., Sandipan M. - Troy: Springer, 2018, P. 313.
5. Pedro P.C. Intelligent Control Systems with LabVIEW / Pedro P.C., Fernando D. R.F. - London: Springer, 2009, P. 216.

УДК 621-926

В.А. Бородай¹, О.Ю. Нестерова¹, О.Р. Ковальов¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

СПОСІБ СИМЕТРУВАННЯ АКТИВНОГО НАВАНТАЖЕННЯ В ЧОТИРИПРОВІДНИХ МЕРЕЖАХ ЗМІННОГО СТРУМУ

Анотація. Проведено аналіз відомих методів симетрування активних навантажень у мережах змінного струму із заземленою нейтраллю та запропоновано оригінальний спосіб і схемне рішення для підвищення якості енергії у системах електропостачання такого типу.

Ключові слова: чотирипровідна мережа із заземленою нейтраллю, якість електроенергії, методи вирівнювання навантажень, спосіб симетрування, схемне рішення.

Вступ. Останнім часом централізоване водяне опалення при значній його вартості і несвоєчасній оплаті суттєво втратило свою ефективність. Тому є маса об'єктивних причин і, перш за все, це зависока вартість первинного енергоносія (природного газу). Як наслідок, пересічні споживачі відмовляються від централізованого і переходять на альтернативні способи індивідуального опалення. На теперішній час досить непоганий результат дає електричне опалення, як водяного типу (електричний котел), так і підігрів завдяки, наприклад, використанню теплої підлоги. Відповідно користувачі електричної енергії самостійно змінюють проектні потужності споживання, наслідком чого стає несиметричне навантаження фаз, а це негативно впливає на інших споживачів та призводить до росту витрат у мережі.

Мета роботи – розробка системи вирівнювання фазних навантажень за умови хаотичного привнесення змін у проектний рівень потужності споживання при наявності прийнятної вартості на апаратуру вирівнювання та одночасного зниження мережних витрат і забезпечення надійного та якісного електропостачання усіх без винятку споживачів.

Основний зміст роботи. Розглянемо можливі варіанти реалізації симетрування мережі. Найпростіший спосіб вирішення проблеми - розробка нового проекту розподілу навантажень для кожної окремої фази. Зрозуміло, що такий підхід буде вимагати значних витрат праці, часу на офіційні узгодження та фінансів.

Автори джерела [1] пропонують здійснювати уникнення перекосу фаз завдяки переходу від системи трифазного постачання на централізоване однофазне постачання. Це передбачає розробку та виготовлення спеціального силового трансформатора, оплату якого буде покладено на усіх мешканців багатоповерхового будинку. Очевидно, що здійснення такої реконструкції мало ймовірно оскільки ні мешканці, ні постачальник енергії не погодяться на такі фінансові витрати.

Можливий варіант установки і автоматичної централізованої системи вирівнювання навантажень [1]. Сутність роботи такого проекту полягає у примусовому створенні протилежного перекосу фаз шляхом стабілізації напруги в окремих фазах. Такій спосіб вимагає наявності силових стабілізаторів на кожній фазі мережі, що теж буде супроводжуватись значними капітальними витратами. При цьому витрати у мережі можуть залишатись.

Достатньої якості енергопостачання можливо досягти при наявності індивідуальних абонентських стабілізаторів напруги. Немає сумніву, що у цьому випадку надійність електропостачання кожного конкретного споживача виросте, але у цілому система постачання залишиться із значними енерговитратами, що для умов теперішнього часу не є припустимим.

Оснащення кожного абонента системою автоматичного керування, яка за миттєвої композиції фазних напруг перемикає навантаження конкретного

споживача на фазу [2], що має вищу напругу, також вимагає значних фінансових вкладень, які не завжди можливо реалізувати.

Підбір схемної композиції пристрою завантаження-розвантаження фаз мережі у залежності від величини струму нульового дроту дає позитивний результат симетрування [3], але має недолік аналогічний попередньому варіанту.

Варіант постачання електроенергії постійного струму замість змінної не витримує критики оскільки усі побутові електро-приймачі розраховані на живлення виключно від мережі змінного струму. Реалізація такого проекту можлива за наявності індивідуальних інверторів у кожного абонента, що за нинішніх умов є не досить просто. Крім цього останні два способи мають труднощі пов'язані із обліком спожитої енергії. Тому напрашується висновок, що відомі способи симетрування чотирипровідної мережі до 1000 В є мало придатними для систем індивідуального електричного опалення. Відповідно є нагальна потреба для створення іншого підходу до вирішення означеної проблеми електропостачання.

Авторами цієї публікації пропонується здійснювати симетрування потужного навантаження (до яких слід віднести системи електроопалення) шляхом його живлення через мостову схему випрямлення у комплексі із регулятором потужності, призначенням якого є виконання двох функцій - обмеження початкової напруги на рівні номіналу елементів обігрівання та власне регулювання потужності для здійснення ощадного споживання енергії із мережі. Щодо обліку спожитої енергії, то його рекомендується виконувати на ланці змінного струму за допомогою трифазного лічильника. Відповідно це дозволить окремо контролювати споживання тільки опалення, а інші споживачі у цьому випадку працюють за звичайною (проектною) схемою. До переваг такого рішення слід віднести:

- досить високий ступень симетрії за відсутності додаткових витрат у мережі живлення;
- відносну простоту схемного рішення системи симетрування, що знизить її собівартість;
- простоту обліку енергії та надання можливості енергоощадного споживання;
- підвищена електробезпека внаслідок живлення обігрівальних елементів постійним струмом.

Наукова новизна роботи полягає у розробці оригінального способу живлення опалювальних елементів за умови ощадності як для постачальника, так і для споживача енергії за одночасної простоти реалізації і низьких капітальних вкладень.

Висновки. Запропонований оригінальний спосіб та схемна реалізація попри залучення всебічного принципу енергоощадності має ще і доступність для пересічного споживача, що за нинішніх умов господарювання є перевагою у порівнянні з іншими відомими способами керування.

ПЕРЕЛІК ПОСИЛАНЬ:

1. <http://www.energsovet.ru/entech.php?idd=135>
2. Черкашина Г.І. Симметризованиережимов работы системы электроснабжения бытового сектора // Энергосбережение • энергетика • энергоаудит. – 2015. – № 4 (135). – С. 21– 28.
3. <https://findpatent.ru/patent/254/2548656.html>

УДК 004.9

І.В. Тихоню¹, І.С. Дмитрієва¹

¹Національна металургійна академія України, Дніпро, Україна

ДОСЛІДЖЕННЯ СТАНУ ТОВСТОСТІННИХ ВТУЛОК ПІСЛЯ ДОРНУВАННЯ

Анотація. Дослідження процесу дорнування за допомогою комп'ютерної моделі, яку було створено у програмному середовищі ANSYS, підтвердило теоретичні відомості і допомогло зібрати статистичні данні на різних етапах процесу.

Ключові слова: дорнування, ANSYS, комп'ютерна модель, моделювання

Вступ. Дорнування - прогресивний процес обробки отворів заготовок холодною пластичною деформацією, при якій робочий інструмент - дорн, переміщаючись уздовж твірної отвору, забезпечує за рахунок натягу зміцнення металу, згладжування мікронерівностей, утворення залишкових напружень, виправлення форми поперечного перерізу отвору.

Постановка задачі. Процеси об'ємного дорнування найбільш ефективно застосовують замість чорнового розточування при обробці отворів трубних заготовок типу гільз і довгих циліндрів. Як правило, при цьому забезпечується значна економія металу і зниження загальної трудомісткості виготовлення виробів.

Основний зміст роботи. Дорнування відноситься до методів обробки з зміцненням за розміром, які частіше використовуються в серійному і масовому виробництві. Цей метод виключає попередню обробку. За один або два проходи можливо отримати дуже точні і чисті отвори. У той же час відбувається зміцнення металу. Це дає дорнуванню важливу технічну та економічну перевагу над іншими технологічними процесами. Залежно від технологічного процесу дорнування підрозділяється на вільне і невільне. Вільне оброблення виконується для труб із середньою товщиною стінок, значення яких не перевищує 200 мм. Переважно це безшовна і електрозварні труба.

Невільне дорнування застосовують для тонкостінних труб. Після закінчення операції на оброблених виробах відсутнє викривлення осі і наявність неякісно випрасуваних ділянок металу. У поздовжньому напрямку труба залишається стійкою до навантажень. Процедура дорнування отворів виконується в жорсткому закріпленні. Дорнування широко використовується в

промисловості при виготовленні напрямних вимірювальних приладів, поршневих пальців двигунів внутрішнього згоряння, підшипників кочення, пневматичних і гідравлічних гільз циліндрів. Також використовується при відновленні зношених деталей: запаси пневматичних і гідроциліндрів, поршневі пальці, універсальні шарніри вали та ін.

При використанні програмного забезпечення для аналізу кінцевих елементів ANSYS для тривимірного моделювання процесу дорнування отворів, важливо, щоб різним елементам були правильно присвоєні відповідні властивості матеріалу, бо на самперед правильна передача властивостей матеріалу впливає на точність результатів моделювання.

У зв'язку із симетричністю геометрії моделі відносно площин XZ та XY, для скорочення часу обчислення, моделювання проводимо на чверті від реальної геометрії. Симетрія задано було додаванням опор без тертя на поверхню, зі сторін, де було вилучено частини (рис. 1).

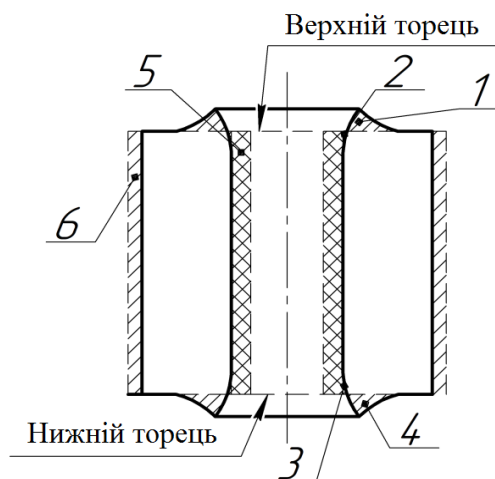


Рис. 1. Схема зсуву металу після дорнування: 1 - наплив металу у верхнього торця; 2 - спотворення циліндричного отвору у верхнього торця; 3 - спотворення циліндричного отвору у опорного торця; 4 - наплив металу у опорного торця; 5 - залишкова деформація по всьому отвору (без урахування спотворення); 6 - збільшення обсягу по зовнішньому діаметру за рахунок залишкових деформацій

При вході дорна в отвір, який обробляється (і виході з нього) в основному пластичне деформування поширюється в напрямку вільного торця. Тому поблизу отвору на поверхні торців утворюється наплив металу, а утворені отвори поблизу торця відхиляється від прямої лінії. В результаті деформацій відбуваються спотворення торців отвору. Найбільші напливи металу спостерігаються на вихідному торці.

При виконанні імітаційного аналізу за допомогою пакету прикладних програм ANSYS насамперед створюємо ескіз взаємодії елементів. Потім необхідно встановити сітку (mesh) для 3D-моделі. У нашому випадку зразок

було подрібнено на 95670 елементів та мав 21420 вузлів. При цьому розмір елемента склав 0,7 мм.

За допомогою методу кінцевих елементів у програмному продукті ANSYS було проведено аналіз залишкової напруги і залишкової деформації.

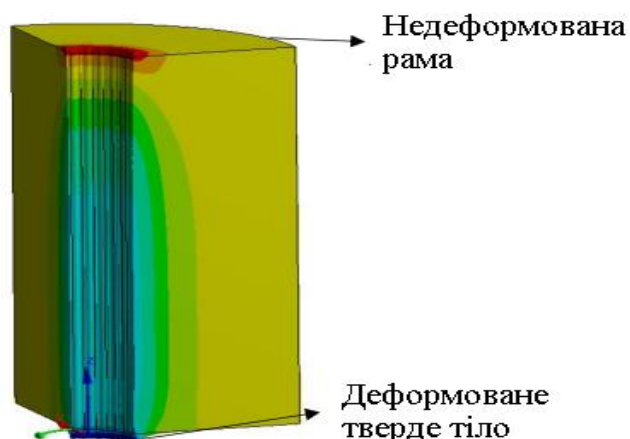


Рис. 2. Схема залишкової деформації заготовки після дорнування

Висновки. В ході виконання роботи було отримано комп'ютерну модель дорнування, зібрано статистичні данні на різних етапах процесу, проаналізовані отримані данні і можна зробити такі висновки:

1) Діаметр отвору в опорі слабо впливає на величину залишкових напружень, які формуються при дорнуванні товстостінних втулок.

2) Найбільший вплив діаметр отвору в опорі надає на максимальну радіальну залишкову напругу і максимальну окружну залишкову напругу.

3) Висота напливу на вихідному торці завжди на 55% більше ніж на вхідному.

4) Вплив діаметру отвору в опорі на висоту наплив незначний.

5) Діаметр отвору в опорі слабо впливає на зміну зовнішнього діаметра і внутрішнього діаметра заготовки.

Таким чином програмний комплекс ANSYS може успішно застосовуватися для вивчення процесу дорнування, а саме, для дослідження формування залишкових напруг і деформацій.

УДК 621-926

Д.Ю. Макуха¹, В.А. Бородай¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ТЕХНІКО-ЕКОНОМІЧНА ЕВОЛЮЦІЯ ГОЛОВНОГО ПРИВОДА ПРОМИСЛОВИХ ПРАЛЬНИХ МАШИН

Анотація. Проведено критичний аналіз існуючих кінематичних схем та недоліків і переваг привідних електродвигунів активаторів промислових

пральних машин. Запропоновані рекомендації щодо раціональних кінематичних схем та їх приводів.

Ключові слова: промислова пральна машина, кінематичні схеми, привод активатора, раціональний привід.

Вступ. Сучасні тенденції розвитку механізації побутових робіт передбачає втілення у кожне господарство пристроїв типу пральних машин. Їх наявність суттєво розвантажує кожного власника від рутинної та важкої і непродуктивної роботи. Одночасно, при наявності такої техніки у кожному господарстві, це призводить до суттєвого зростання витрат ресурсів у масштабах міста та держави. Тому деякі промислово розвинуті держави світу йдуть по шляху розробки централізованих пральних станцій, які додатково мають більшу довговічність, надійність, а саме головне мають можливість економії всіляких ресурсів, що є первинною задачею теперішнього дня.

Мета роботи – аналіз та обґрунтування напряму сучасного техніко-економічного розвитку головного приводу промислових пральних машин.

Основний зміст роботи. На сьогодні існує декілька поширених типів електроприводу промислових пральних машин:

дводвигунова роз'єднана кінематична система, кінематична система поєднаного приводу, кінематичні системи бокового та центрального приводів. У кожній схемі є свої переваги та недоліки.

Дводвигунова роз'єднана система складається із двох асинхронних двигунів підключених паралельно. Один із них є двигун, що забезпечує прання із швидкістю обертання приблизно 1000 або 1500 оборотів за хвилину. Другий це асинхронний двигун віджиму зі швидкістю обертання до 3000 об/хв. Основними недоліками такої системи є використання двох асинхронних двигунів, що підвищує вартість механізму, а з'єднуючі муфти знижують надійність системи.

Дводвигунова поєднана система складається із двох двигунів, зазвичай, з асинхронного та колекторного. Двигуни розташовані на одному валу і приводяться у рух одночасно. У режимі прання працює асинхронний двигун, а у режимі віджиму – колекторний. У порівнянні із попередньою конструкцією, поєднана має менші масо-габаритні показники. Але, значним недоліком є наявність колекторно-щіткового вузла, який зменшує надійність приводу.

Розглянуті системи на тепер є морально застарілими, а їх виявлені недоліки не сприяють використанню у сучасних конструкціях пральних машин.

Суттєво надійним та відносно доступним варіантом конструкції пральної машини є двошвидкісний асинхронний привід із приєднанням його до барабану через бокову ремінну передачу або електродвигун з'єднаний з баком безпосередньо через головний вал (прямий привід). Такі способи з'єднання робочого двигуна з активатором пральної машини мають найбільшу робастність та надійність системи у цілому. Боковий привід допомагає зменшити пральну машину в довжину, а прямий – у висоту. Кожен із цих способів набув поширення, але, прямий спосіб з'єднання більш надійний, так як не має ремінної передачі, яка з часом втрачає свої властивості.

Двошвидкісні двигуни зазвичай керуються релейно-контакторними схемами. При цьому їх швидкість не стабілізується, що призводить до деякого зниження продуктивності системи. А недотримання встановленої ваги завантаженої білизни може додатково призвести до погіршення енергетики асинхронного двигуна.

Головною ідеєю публікації є обґрунтування переваг системи перетворювач частоти асинхронний двигун у приводі промислової пральної машини. Впровадження такого регулювання дозволяє:

- значно підвищити точність регулювання;
- покращити енергетику системи;
- обмежити пускові струми.

Скалярний метод керування ПЧ дозволяє здійснювати регулювання без наявності датчика швидкості за підтримки точності стабілізації $\pm 2.5\%$ від заданої. Такий стан регулювання у кінцевому рахунку підвищить якість прання, забезпечить кращі енергетичні режими, зменшить у сталому та перехідних режимах нагрів обмоток, і як наслідок збільшить термін безвідмовної експлуатації.

Наукова новизна роботи полягає у наданні обґрунтованої рекомендації щодо переходу конструкції приводу головного руху промислової пральної машини від існуючої до системи частотний перетворювач асинхронний двигун, що на теперішній час є найбільш перспективною.

Висновки. Очевидні переваги системи перетворювач частоти асинхронний двигун в конструкціях пральних машин може призвести до покращення якості відпрацювання технології, форми перехідних процесів у двигуні, зменшить ударне навантаження на валу двигуна та відповідно підвищить надійність системи у цілому.

ПЕРЕЛІК ПОСИЛАНЬ:

1. С.Л. Корякин-Черняк «Стиральные машины от А до Я». 2-е изд. — М.: СОЛОН-Пресс, НіТ, 2005. — 295 с.: ил. — (Библиотека ремонта).
2. Соколовский Г.Г. Электроприводы переменного тока с частотным регулированием: учебник для вузов. — М.: Издательский центр «Академия», 2006.
3. Терехов В. М., Системы управления электроприводов: учебник для вузов. — М.: Издательский центр «Академия», 2005.
4. Колб А.А., Колб А.А. Теорія електроприводу: Навч. посібник. Д.: НГУ. — 2006.

РОЗДІЛ 5

ПРОБЛЕМИ ДИСТАНЦІЙНОЇ ОСВІТИ

УДК 622.6-52

Н.П.Уланова¹, А.Г.Шпорта¹, В.В. Приходько¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ДОСВІД ВИКЛАДАННЯ ВИЩОЇ МАТЕМАТИКИ В НТУ «ДНІПРОВСЬКА ПОЛІТЕХНІКА» В УМОВАХ ПАНДЕМІЇ COVID-19

Анотація. В роботі розглянуто деякі аспекти організації навчального процесу в умовах переходу на електронне навчання студентів-очників у зв'язку з епідеміологічною ситуацією в світі.

Ключові слова: електронне навчання, онлайн-курс, математична підготовка, труднощі переходу, якість освіти, пандемія.

Вступ. Математична підготовка студентів є найважливішою складовою інженерної підготовки студентів технічних навчальних закладів. Формування системи теоретичних знань та практичних навичок, необхідних для вивчення загальнонаукових і спеціальних дисциплін, підвищення загальної математичної культури студентів, розвиток логічного й алгоритмічного мислення, підготовка до застосування отриманих знань та навичок у процесі професійної діяльності – основні задачі якісної математичної освіти.

Вимушений перехід до електронного навчання на основі дистанційних технологій ставить додаткові вимоги до викладання вищої математики, які пов'язані зі змінами у діяльнісній, комунікативній та предметно-структурній складових освітнього процесу.

Постановка задачі. В ситуації, що склалася, виникає необхідність пошуку методів планування, реалізації та моніторингу навчального процесу, організації взаємодії всіх його учасників, вибору освітнього середовища і технічних засобів для забезпечення якісного навчання студентів в умовах пандемії COVID-19.

Основний зміст роботи. Слід зазначити, що на кафедрі вищої математики провадиться систематична робота щодо створення навчальних ресурсів та вдосконалення процедури оцінювання знань. При цьому особлива увага останніми роками приділяється організації самостійної роботи студентів із залученням засобів інформаційних технологій [1, 2, 3]. Реалізацію такого формату навчання забезпечує впроваджена в університеті освітня платформа Moodle, яка дозволяє охоплювати широку аудиторію і вирізняється гнучкістю та надійністю.

При створенні навчального електронного курсу вищої математики використовувався досвід, накопичений під час вивчення матеріалів онлайн-курсів з категорії «математика», розташованих на платформі Coursera.

Розроблений авторами курс, призначений для організації самостійної роботи студентів очної та заочної форм навчання, був розміщений в інструментальному середовищі університету і використовувався ще до початку пандемії. Під час переходу на дистанційний формат цей курс застосовується для забезпечення навчального процесу на всіх етапах вивчення вищої математики.

Кожна частина курсу присвячена певному розділу математики і містить доповнений розв'язанням типових прикладів теоретичний матеріал, а також завдання для самостійної роботи, тести та індивідуальні завдання. Можливості інформаційно-освітнього середовища дозволяють полегшити роботу з текстом, активізувати пізнавальну діяльність. Так, наприклад, у тексті підсвічуються терміни, розміщені в глосарії, можлива миттєва навігація по тексту на базі гіперпосилань, уточнення окремих понять за допомогою “спливаючого вікна”. Частина розміщених в курсі презентацій виконує роль тренажерів, які формують послідовність операцій, наприклад, при знаходженні похідної заданої функції. Рисунки з посиланнями на відповідні сайти ілюструють практичне застосування матеріалу, що викладається. Ряд тем доповнено узагальнюючими відео-лекціями.

Важливу роль на різних етапах організації контролю знань відіграє озміщений в курсі блок тестових завдань. Він дозволяє здійснювати вхідний контроль, проводити поточну перевірку знань під час вивчення кожної теми, діагностувати якість засвоєння взаємозв'язків між окремими елементами навчального матеріалу цілого розділу, сприяє систематизації та узагальненню отриманих знань. Після закінчення поточної перевірки з використанням тестування студенти мають можливість ознайомитися з протоколом результатів та правильних відповідей, а також коментарями викладача. Таким чином, реалізується головна функція такої перевірки – навчальна. При наповненні блоку використовувалися тести різноманітної структури закритого та відкритого типів.

Основна мета математичної складової інженерної підготовки студентів технічних навчальних закладів – навчити логічному міркуванню, вмінню обґрунтовувати твердження, що неможливо в умовах відсутності безпосереднього спілкування з викладачем.

Проблема комунікацій «викладач-студент» частково вирішується за допомогою додатків Zoom та Microsoft Teams. Ці програмні засоби використовуються для проведення занять і консультацій в режимі реального часу та здійснення візуального контакту з викладачем. Наявність інтерактивної дошки, можливість обміну контентом та архівування занять дозволяють активізувати участь студентів під час віддаленого навчання. Моніторинг відвідування занять відбувається шляхом застосування функції учасника.

Виконані домашні та індивідуальні завдання, контрольні роботи надсилаються студентами в розділ «Завдання» робочого простору Teams у вигляді фото сторінок роботи, об'єднаних у файли форматів word або pdf. У вказаному розділі викладач має можливість написати зауваження до окремих

завдань надісланої роботи, виставити оцінку, яка заноситься у протокол, і відправити студенту.

Викладачі кафедри в якості експерименту випробували й інший спосіб перевірки робіт – завантажували файл студента у графічний редактор і вносили виправлення безпосередньо в роботу, як під час традиційної перевірки робіт на папері. Такий спосіб є більш ефективним в плані навчання студента, але потребує багато часу на перевірку кожної роботи, що не узгоджується з існуючими нормативами роботи викладачів у відповідності до їхнього учбового навантаження.

Істотним недоліком віддаленого навчання є недосконалість контролю засвоєних знань. Автори статті застосовували комбінований спосіб, який здійснювався на всіх етапах навчання з використанням відеоконференцій у додатках Teams або Zoom. На першому етапі студенти виконували контрольну роботу або тестові завдання і надсилали їх на перевірку викладачеві. Другий етап приймання завдань відбувався в режимі діалога викладача зі студентом і дозволяв обговорити отримані результати, вказати на помилки і запропонувати студенту індивідуальну траєкторію засвоєння або повторення матеріалів курсу. Високі результати при електронному навчанні демонструють, в першу чергу, мотивовані студенти, які розуміють важливість і необхідність отримуваних знань та володіють такими якостями, як самодисципліна й усвідомленість.

Подальше вдосконалення форм організації контролю, на наш погляд, має відбуватись шляхом створення систем оцінювання, які б враховували не тільки рівень засвоєних знань, а й ступінь розвитку вмінь та навичок, творчих здібностей, аналітичного мислення.

З метою більш глибокого розуміння проблем, які виникають внаслідок вимушеного переходу на електронне навчання, було проведено опитування серед студентів першого курсу. В опитуванні прийняло участь 135 студентів спеціальностей 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія» і 172 «Телекомунікації та радіотехніка».

Навчання студентів в період пандемії відбувалось у повному обсязі за розкладом занять згідно навчального плану.

Проведене опитування показало, що 80% студентів вже мали досвід онлайн-навчання, тому перехід на дистанційну форму відбувався без проблем, з незначними проблемами – у 89% опитуваних. Разом з тим деякі студенти (15%) не мали необхідного обладнання; більше половини студентів скаржились на пов'язані з Інтернетом технічні збої під час навчання.

Майже всі студенти визначили найбільш зручним для навчання додаток Teams; 46% опитуваних активно користуються електронними навчальними матеріалами з вищої математики, розміщеними на платформі Moodle. Як відзначають студенти, відповіді на складні питання щодо матеріалу курсу вони отримують у ході відеоконференцій (63% опитуваних), у відео-чатах з викладачами (45%), на онлайн-консультаціях, які найбільшою мірою наближають навчання до очної форми (32%).

Електронне навчання визнали зручним 42% студентів. В першу чергу відмічається його доступність і відкритість, можливість навчатися там, де є

Інтернет і комп'ютер (78%), гнучкість та індивідуальний характер дистанційної форми (50%). Разом з тим до труднощів, пов'язаних з переходом на новий формат, студенти відносять відсутність очної комунікації з викладачем (34%) та ізолюваність від різнопланового спілкування з однолітками (47%). Перевагу очному навчанню віддали 78% респондентів, які мали значні труднощі при переході на електронне навчання.

Практично всі студенти бачать перспективи використання дистанційних освітніх технологій у своїй професійній діяльності.

Висновки. Враховуючи досвід викладання вищої математики в дистанційному форматі та думку студентів, можна констатувати, що електронне навчання порівняно з традиційним є більш гнучким, доступним, відкритим, йому притаманна більша індивідуальність. Проте ця система в наші дні має ряд недоліків, які порушують єдність процесів навчання, виховання і розвитку студентів. До них належать складності в систематизації знань і набутті практичних навичок, відсутність емоційної та стимулюючої складової у спілкуванні викладача зі студентом. Це і втрата однієї з найважливіших передумов розвитку особистості – змагання між студентами. Такий формат не підходить для розвитку комунікабельності, навичок роботи в команді. Автори вважають, що поєднання очного та електронного навчання більшою мірою дозволить побудувати гнучкий процес навчання з використанням усіх ресурсів інформаційно-освітнього середовища.

ПЕРЕЛІК ПОСИЛАНЬ:

1. В.И.Сулаев. Проектирование и реализация сетевого учебного курса "Высшая математика" /В.И.Сулаев, Л.И.Мещеряков, Н.П.Уланова, В.В.Приходько. //Сб. науч.трудов НГУ.–Д.: НГУ, 2013.–№40.–С.222-228.

2. В.И. Сулаев. Некоторые аспекты методического обеспечения технологий дистанционного обучения/ Сулаев В.И., Мещеряков Л.И., Уланова Н.П. //Сборник научных трудов международной конференции «Современные инновационные технологии подготовки инженерных кадров для горной промышленности и транспорта 2016». – Д.: НГУ, 2016. – С. 496-506.

3. А.Я.Рибалко. Комп'ютерна візуалізація технологій складання електричних машин в умовах дистанційного навчання/ А.Я.Рибалко, Л.І.Мещеряков, Н.П.Уланова, А.Ю.Руссу // Наукові записки Тернопільського національного педагогічного університету ім. В. Гнатюка. Серія Педагогіка.– 2011.– №1.– С.49-54.

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ОСВІТІ

Анотація. Описано найбільш популярні сучасні інформаційні технології, що використовуються в освітньому процесі.

Ключові слова: *інформаційні технології, освіта, Google classroom, Moodle, Zoom, Skype, Google meet, Office 365.*

Вступ. Сучасний період розвитку суспільства характеризується сильним впливом на нього інформаційних технологій, які проникають в усі сфери людської діяльності і утворюють глобальний інформаційний простір. Основні надії покладаються на створення і супровід інформаційно-освітніх середовищ відкритого і дистанційного навчання, на розвиток нових об'єктних технологій створення баз навчальних матеріалів, поряд з розвитком традиційних технологій розробки цифрових освітніх ресурсів.

Невід'ємною і важливою частиною цих процесів є інформатизація освіти. Цей процес супроводжується певними змінами в педагогічній теорії і практиці навчально-виховного процесу. Інформаційні технології покликані стати не додатковим засобом в навчанні, а невід'ємною частиною цілісного освітнього процесу, що значно підвищує його ефективність. Створення дистанційної освіти - це найбільш швидкий і ефективний шлях до підвищення інтелектуального потенціалу суспільства, прискорення процесу переходу студента до інформаційного суспільства.

Основний зміст роботи. Головна мета інформаційних технологій - забезпечення сучасної якості освіти на основі збереження його фундаментальності та відповідності актуальним і перспективним потребам особистості, суспільства, держави.

Завдання інформаційних технологій:

- підвищити мотивацію навчання;
- підвищити ефективність процесу навчання;
- сприяти активізації пізнавальної сфери студентів;
- удосконалювати методики проведення пар;
- своєчасно відстежувати результати навчання і виховання;
- планувати і систематизувати свою роботу;
- використовувати, як засіб самоосвіти;
- якісно і швидко підготувати захід.

Інформаційні технології надають можливість:

- раціонально організувати пізнавальну діяльність здобувачів в ході навчального процесу;
- зробити навчання більш ефективним, залучаючи всі види чуттєвого сприйняття студента в мультимедійний контекст і озброюючи інтелект новим концептуальним інструментарієм;

– побудувати відкриту систему освіти, що забезпечує кожному індивіду власну траєкторію навчання;

– залучити до процесу активного навчання категорії дітей, що відрізняються здібностями і стилем навчання;

– використовувати специфічні властивості комп'ютера, що дозволяють індивідуалізувати навчальний процес і звернутися до принципово нових пізнавальних засобів;

– інтенсифікувати всі рівні навчально-виховного процесу.

Сучасне матеріальне виробництво та інші сфери діяльності все більше потребують інформаційного обслуговування, переробці величезної кількості інформації. Універсальним технічним засобом обробки будь-якої інформації є комп'ютер, який грає роль підсилювача інтелектуальних можливостей людини і суспільства в цілому, а комунікаційні засоби, що використовують комп'ютери, служать для зв'язку і передачі інформації.

Сучасні розробники створили низку платформ та додатків, для зручного користування. Серед них є сервіси, за допомогою яких можна проводити онлайн-заняття, вебінари, створювати конференції, розробляти електронні завдання, створювати таблиці та інше. Ось деякі з них:

– Google classroom.

– Moodle.

– Zoom.

– Skype.

– Google meet.

– Office 365.

Інформаційні технології постали як соціальне явище, що визначає положення людства в світовому масштабі, так як при цій процедурі відбувається сольватація людства і підтримується популярність численної інформації.

Платформи та додатки, які використовують, для зручного організування онлайн-конференцій, зустрічей, вебінарів (таблиця 1).

За результатами опитування (рис.1), великий відсоток людей використовують Microsoft Teams, для проведення онлайн-зустрічей на велику кількість учасників, так як він не має обмежень за часом, доступний на будь-якій платформі, так само має запис екрану, за допомогою якої, люди, які не змогли брати участь в конференції, з легкістю можуть переглянути її пізніше.

За результатами опитування на рахунок того, чому інформаційні технології важливі в освіті, студенти вищих навчальних закладів відповіли таким чином:

– Повинно йти в ногу з часом. Зараз відбувається масова машинізація всього і елементарні знання повинні бути у кожного незалежно від бажання або ж небажання вчити це.

– Вони прискорюють, полегшують і автоматизують процес навчання.

– Світ живе в століття технологій. І це прекрасна можливість поліпшити способи вивчення чогось за допомогою сучасних технологій.

– В наш час, а саме під час пандемії, саме інформаційні технології змогли врятувати процес навчання студентів і школярів. Так як був введений карантин

і навчальні заклади були закриті, на допомогу нам прийшли навчальні платформи (наприклад Microsoft Teams).

Таблиця 1

Найбільш популярні платформи для дистанційного навчання

Інструмент	Zoom	Skype	Microsoft Teams	Google meet	Google Hangouts
Платформа	Mac, Windows, Linux, IOS, Android				
Кількість учасників конференції	100	50	250	100	25
Загальний чат	+	+	+	+	+
Демонстрація екрана	+	+	+	+	+
Передача файлів	+	+	+	+	+
Пряма трансляція	+	+	+	+	+
Запис конференцій	+	+	+	-	-
Додаткові функції	Підміна фону на відео, демонстрація з будь-якої програми з iPhone/iPad	Розмиття фону, автоматичні субтитри	Попередній відео-перегляд, зборів з повністю інтегровані засобами голосового і відеозв'язку	Попередній перегляд екрану, інтеграція з додатком Microsoft Office	Під час пандемії доступ до преміум-функцій безкоштовна, синхронізація чатів, інтеграція з календарями Google
Особистий чат	-	+	+	-	+
Обмеження безкоштовної версії	40 хвилин	-	-	60 хвилин	-

– Сучасні технології мають багато напрямків і допомагають зараз в будь-якій сфері. Ці технології допомагають добути більше достовірної інформації для вивчення. Якраз тільки що дивилася проєкт з історії де багато інформації встановлювали саме за допомогою технологій, що дає можливість дізнатися достовірну інформацію про минуле, а не просто те що придумали і написали в підручнику колись.

– Завдяки платформам і додаткам для онлайн навчання викладачі та здобувачі освіти змогли проводити заняття дистанційно. Причому не проблема

провести як і лекційні, так і практичні заняття. Скажимо більше, у дистанційного навчання є ще великий плюс, а саме: лекційні заняття можуть відвідувати дуже велика кількість людей, припустимо якщо це лекція для всього потоку. Помістити сотні людей в аудиторії важче, ніж в онлайн зборах.

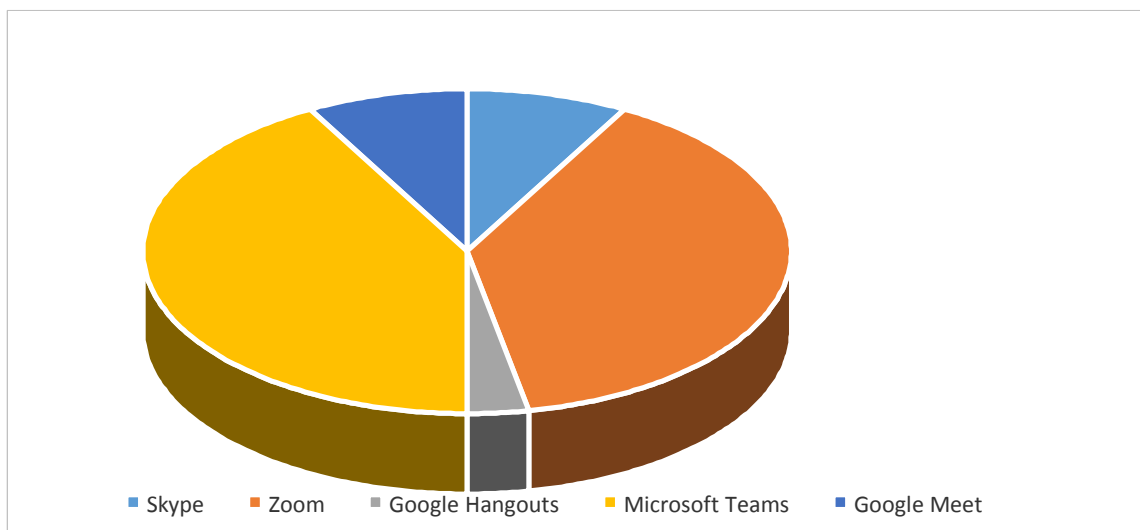


Рис. 1. Результатами опитування використання платформ дистанційного навчання

– Новітні інформаційні технології в навчанні дозволяють активніше використовувати науковий та освітній потенціал провідних університетів та інститутів, залучати на йкращих викладачів до створення курсів дистанційного навчання, розширювати аудиторію студентів.

Висновки. Використання інформаційних технологій допомагає поліпшити освітню діяльність, збільшує якість процесу навчання і підвищення ефективності індивідуальної діяльності студентів. У новому тисячолітті в міру просування до створення інформаційного суспільства дистанційна освіта відіграватиме вельми важливу роль. З кожним днем сучасні освітні технології та платформи будуть вдосконалюватися і наповнюватися новими знаннями. Використання інформаційних технологій створює можливість підвищення ефективності процесів навчання і засвоєння знань.

ПЕРЕЛІК ПОСИЛАНЬ:

1. Горбунова Л.И., Субботина Е.А. Использование информационных технологий в процессе обучения/ Молодой ученый. – 2013. – № 4. – С. 544–547.
2. <https://skyteach.ru/2019/01/14/zoom-platforma-dlya-provedeniya-onlajn-zanyatij/>
3. Пугачев А.С. Дистанционное обучение – способ получения образования // Молодой ученый. – 2012. – № 8. – С. 367–369.
4. <https://www.skype.com/ru/developer/>
5. <https://apps.google.com/intl/ru/meet/>
6. В.П. Демкин, Г.В. Можаява. Информационные технологии дистанционного обучения.

ШТУЧНИЙ ІНТЕЛЕКТ В МИСТЕЦТВІ ТА ТВОРЧОСТІ

Анотація. Представлено огляд використання методів та інструментарію штучного інтелекту в різних галузях мистецтва та вплив на творчість людства.

Ключові слова: *штучний інтелект, інформаційні технології, традиційне мистецтво, освіта, спектр творчих навичок, ILLIAC, AARON.*

Вступ. Штучний інтелект (ШІ) вже не є гіпотетичною технологією, а натомість є технологією, яка затоплює наше повсякденне життя за допомогою персональних помічників, таких як Siri, сугестивних пошукових запитів в Google та самокерованих автомобілів. У світлі зростаючого значення штучного інтелекту, ШІ є темою бурхливих дискусій, де деякі люди розглядають останні тенденції як позитивні, а інші - як негативні. Деякі вважають, що ШІ зробить життя людини безпечнішим і процвітаючим, а інші стверджують, що ШІ може стати неконтрольованим і врешті-решт загрожувати людському суспільству. Творчі зусилля - ще одна сфера, на якій, здається, ШІ перевершує людей. Наприклад, у 2016 році AlphaGo від Google Deepmind (комп'ютерна програма Go) успішно освоїла складну гру і швидко отримала високу оцінку за її «креативність».

Основний зміст роботи. Традиційне мистецтво - ще одна сфера, в якій ШІ продемонстрував швидке вдосконалення. Це має важливі наслідки, оскільки мистецтво, а саме живопис, протягом тисячоліть розглядалося у більшості країн світу як вершину людської творчості. Таким чином, досягнення в галузі штучного інтелекту обов'язково ускладнюють сучасне розуміння творчості та естетичної краси в мистецтві. В даний час проводяться дослідження з побудови штучного інтелекту для створення музики та віршів. Хтось може сказати, що ці творчі продукти, створені ШІ, просто імітують людську працю. Однак слід враховувати, що навіть людські творчі продукти починали з наслідування від інших, оскільки люди також є імітаційними істотами. Творчість видається загадковою, оскільки коли у нас є творчі ідеї, дуже важко пояснити, як ми їх отримали, і ми часто говоримо про нечіткі поняття, такі як „натхнення” та „інтуїція”, коли намагаємось пояснити творчість. Той факт, що ми не усвідомлюємо, як проявляється творча ідея, не обов'язково означає, що наукове пояснення не може існувати. Насправді ми не знаємо, як ми виконуємо інші дії, такі як розуміння мови, розпізнавання образів тощо, але ми маємо все кращі та кращі методи ШІ, здатні відтворити такі дії.

Штучний інтелект відіграв вирішальну роль в історії комп'ютерної музики майже з її початку в 1950-х роках. Однак до недавнього часу найбільше зусиль було спрямовано на композиційні та імпровізаційні системи, і мало зусиль приділялося виразному виконанню. Робота Хіллера та Ісааксона (1958) над комп'ютером ILLIAC - це найвідоміша новаторська робота в галузі

комп'ютерної музики. Їх головним результатом є Iliac Suite, струнний квартет, що складається за принципом вирішення проблем «генерувати та перевіряти». Програма генерувала нотатки псевдовипадково за допомогою ланцюжків Маркова. Сформовані ноти були перевірені за допомогою евристичних композиційних правил класичної гармонії та контрапункту. Зберігалися лише примітки, що відповідають правилам. Якщо жодна із створених нотаток не задовольняла правилам, застосовувалася проста процедура зворотного відстеження, щоб стерти всю композицію до цього моменту, і новий цикл було розпочато знову. Цілі Хіллера та Ісаксона виключали все, що стосується виразності та емоційного змісту.

AARON - це роботизована система, розроблена протягом багатьох років художником і програмістом Гарольдом Коеном, яка може взяти пензлик із роботизованою рукою і самостійно малювати на полотні. Він малює людей в ботанічному саду не просто роблячи копію існуючого малюнка, а створюючи стільки унікальних малюнків на цю тему, скільки від нього може знадобитися. AARON ніколи не бачив людини і не ходив ботанічним садом, але отримав знання про положення тіла та рослини за допомогою правил. Знання AARON і те, як AARON використовує свої знання, не схожі на знання, які ми, люди, маємо і використовуємо, тому що людські знання засновані на переживанні світу, і люди переживають світ своїми тілами, своїми мізками та репродуктивними системами, які комп'ютери не мають. The Painting Fool Саймона Колтона набагато більш автономний, ніж AARON. Незважаючи на те, що програмне забезпечення фізично не застосовує фарбу до полотна, воно імітує багато стилів у цифровому вигляді - від колажу до мазків. За словами Колтона: потрібні лише мінімальні вказівки, і він може придумати власні концепції, завітавши в Інтернет за вихідними матеріалами. Програмне забезпечення здійснює власні веб-пошуки та сканування через веб-сайти соціальних мереж. Ідея полягає в тому, що такий підхід дозволить виробляти мистецтво, яке є значущим для аудиторії, оскільки воно по суті спирається на людський досвід, коли ми діємо, відчуваємо та сперечаємось в Інтернеті. Наприклад, у 2009 р. The Painting Fool створив власну інтерпретацію війни в Афганістані на основі новин.

Чи можемо ми використовувати штучний інтелект для підтримки людської творчості та відкриттів? Нова тенденція, відома як Assisted Creation, має важливі наслідки для творчості: з одного боку, допоміжні системи створення роблять широкий спектр творчих навичок більш доступним. З іншого боку, спільні платформи, такі як розроблена в рамках європейського проекту PRAISE для вивчення музики, полегшують освоєння нових творчих навичок. PRAISE - це навчальна платформа на основі соціальних мереж, що включає людей та інтелектуальні програмні агенти, які надають студенту-музиканту зворотний зв'язок щодо музичної композиції, аранжування та виконання. Студенти завантажують свої рішення до заданого плану уроку, який надає викладач (композиції, аранжування чи вистави). Потім інтелектуальні агенти, а також інші колеги-студенти та викладачі аналізують ці рішення та надають зворотній зв'язок. Наприклад, у випадку композиції агент може сказати: «Ваша модуляція

звучить досить добре, але ви можете спробувати модулювати все на більшу третину для тактів 5 – 8».

Маргарет Боден зазначила, що навіть якщо штучно інтелектуальний комп'ютер був би таким же творчим, як Бах чи Ейнштейн, для багатьох він був би на вигляд творчим, але насправді не творчим. Я повністю погоджуюсь з нею з двох основних причин, таких як: відсутність навмисності та наше небажання відводити місце в нашому суспільстві штучно розумним агентам. Що стосується соціального неприйняття, то причини, через які ми так неохоче приймаємо, що небіологічні агенти можуть проявляти творчість, полягають у тому, що вони не мають природного місця в нашому суспільстві людей, і рішення прийняти їх матиме важливі соціальні наслідки. Отже, набагато простіше сказати, що вони видаються розумними, креативними тощо, замість того, щоб стверджувати, що вони і є такими. Одним словом, це моральне, а не наукове питання. Третя причина відмови у творчості комп'ютерних програм полягає в тому, що вони не усвідомлюють своїх досягнень. Це правда, що машини не мають свідомості і, можливо, ніколи не матимуть свідомого мислення; однак відсутність свідомості не є основною причиною заперечення потенціалу для творчості чи навіть потенціалу інтелекту.

Висновки. Зрештою, комп'ютери не були б першим прикладом несвідомих творців; еволюція є першим прикладом, як блискуче зазначає Стівен Джей Гулд (1996): «Якщо для створення потрібен дальновидний творець, то як сліпій еволюції вдається будувати такі чудові нові речі, як ми самі?»

ПЕРЕЛІК ПОСИЛАНЬ:

1. Arcos, J. L., López de Mántaras, R., and Serra, X. 1998. "Saxex: A case-based reasoning system for generating expressive musical performances." *Journal of New Music Research* 27(3): 194–210.
2. Colton, S., López de Mántaras, R., and Stock, O. 2009. "Computational creativity: Coming of age." *Special issue of AI Magazine* 30(3): 11–14.
3. Franklin, J. A. 2001. "Multi-phase learning for jazz improvisation and interaction." In *Proceedings of the Eighth Biennial Symposium on Art and Technology*. New London, CT: Center for Arts and Technology, Connecticut College.
4. Fry, C. 1984. "Flavors Band: a language for specifying musical style." Reprinted in *Machine Models of Music*, S. M. Schwanauer and D. A. Levitt (eds.). Cambridge, MA: The MIT Press, 1993, 427–451.
5. <https://rb.ru/longread/ai-art/>

ФОРМАЛІЗАЦІЯ ЗАДАЧІ СКЛАДАННЯ НАВЧАЛЬНИХ ПЛАНІВ У ВИЩОМУ НАВЧАЛЬНОМУ ЗАКЛАДІ

Анотація. Розглядаються питання, пов'язані зі створенням навчального плану спеціальності у вищому навчальному закладі. Актуальними проблеми опису завдання складання і модифікації даного нормативного документа. Пропонується використовувати формалізацію основних положень для подання і маніпулювання знаннями з метою визначення пошуку шляхи вирішення.

Ключові слова: навчальний план, вищий навчальний заклад, знання, аудиторне навантаження, інтелектуальна самонавчальна система, нормативні дисципліни.

Вступ. Останнім часом спостерігається широке впровадження комп'ютерної техніки в системи створення і супроводу документів різного типу. Особливий інтерес представляють завдання, пов'язані з розробкою і впровадженням систем електронного документообігу у вищих навчальних закладах. Необхідність використання таких програмних продуктів викликана великим обсягом робіт зі складання і супроводу документів, численними труднощами, що виникають в процесі роботи факультетах тов ВНЗ. Так, у зв'язку зі зміною системи освіти відповідно до Болонського процесу передбачається зростання обсягів документообігу в реформованих вищих навчальних закладах.

Основний зміст роботи. У Дніпропетровському національному університеті розробляється проект «Деканат», який дозволяє автоматизувати роботу його служб. До актуальних відноситься задача створення навчальних планів спеціальностей і розподілу навантаження викладачів кафедр. Рішенням цих питань займається один з основних модулів даної системи, що використовує алгоритми і методи штучного інтелекту.

У зв'язку з цим виникає необхідність опису існуючих знань для складання навчальних планів. У статті пропонується використання формальних систем для вирішення даного завдання.

Основним нормативним документом, що визначає організацію процесу навчання, є навчальний план, який складається з графіка і основного плану навчального процесу.

Навчальний план визначає перелік дисциплін, якими студент повинен оволодіти для досягнення певного освітнього рівня і отримання кваліфікації (бакалавр, магістр), відомості про їхній обсяг (в кредитах і годинах), вигляду індивідуального завдання, характеристики підсумкового контролю знань і послідовність представлення дисциплін.

Для планування процесу навчання з урахуванням певних змін, які повинні бути внесені для виконання інструктивних документів Міністерства освіти і

науки України або Кабінету Міністрів, складається навчальний план року набору. Він розробляється випускаючою кафедрою спеціальності разом з навчально-методичною комісією за напрямом підготовки на основі стандартів освіти держави, галузі та університету.

Загальний обсяг годин з дисципліни повинен включати час на проведення лекцій, практичних, семінарських і лабораторних занять, консультацій, практик, самостійної та індивідуальної роботи і контрольних заходів. На навчальні, виробничі практики, кваліфікаційний іспит, підготовку і захист випускних, дипломних та магістерських робіт кредити відводяться окремо і встановлюється в них співвідношення годин аудиторної, індивідуальної та самостійної робіт залежить від змісту і специфіки окремих видів діяльності студента при їх виконанні.

План навчального процесу визначається відповідною структурно-логічною схемою і містить:

- перелік нормативних дисциплін;
- перелік вибіркових дисциплін;
- розподіл дисциплін за циклами навчання і кафедрами;
- види навчальних занять;
- загальний і річний обсяг дисциплін (час на освоєння) в годиннику і кредитах ЄСПК;
- аудиторна навантаження студента;
- обсяг часу на самостійну роботу;
- форми контролю освоєння модулів.

Час на освоєння дисципліни вимірюється в кредитах згідно з Європейською системою перезасчітвання кредитів. Кредит є уніфікованою одиницею виміру навчального навантаження студента (36 годин), необхідної для освоєння модуля.

Дисципліни загальним обсягом не більше трьох-чотирьох кредитів, як правило, викладають протягом семестру, а курси об'ємом більше чотирьох кредитів - протягом всього навчального року.

Кількість годин, що виділяється на вивчення дисципліни, є вихідним для планування процесу навчання, зокрема, оцінки можливості викладання курсу за семестр. При цьому, повинна дотримуватися тенденція збільшення саме самостійної роботи студента.

Труднощі реалізації процесу складання навчального плану полягають в необхідності вирішення складного завдання, яке враховує безліч різних чинників (облік кількості кредитів, обсягу самостійної роботи студента, розподіл дисциплін за циклами навчання). Цей процес в значній мірі відрізняється елементами нечіткої логіки з множинністю рішень. Тому для пошуку раціонального розподілу навчальних дисциплін бажано використання методів і систем штучного інтелекту.

Розробка і впровадження системи такого типу дозволяє знаходити безліч можливих рішень поставленого завдання за допомогою обраних методів пошуку. Надалі виділити ті, що задовольняють всім заданим обмеженням і умовам. Створення інтелектуальної самонавчальної системи для складання

навчальних планів передбачає скорочення витрат часу працівників кафедр ВНЗ, так як складання кожного наступного навчального плану реалізується з урахуванням попередніх.

Для визначення формальної системи необхідно задати кінцевий алфавіт, визначити процедуру побудови формул, виділити безліч аксіом, задати кінцеве безліч правил виведення (правила продукцій і переписування). У зв'язку з тим, що вхідні дані формулюються на природній мові, при формалізації якісних знань може бути застосована теорія нечітких множин, а саме аспекти, пов'язані з лінгвістичної невизначеністю. Формальний опис предметної області проводиться шляхом використання числення предикатів, для якого розроблені механізми виведення.

Створення такої формальної системи є важливою і трудомістким завданням, в якій повинні бути визначені основні параметри, що впливають на складання плану року набору, і сформульовані правила, що описують даний процес. При цьому необхідно враховувати обмеження, що накладаються на кількість аудиторних годин і кредитів протягом одного семестру, логіку викладання курсів, поділ дисциплін за циклами і особливості різних спеціалізацій. Важливим фактором є логіка викладу дисциплін залежно від курсу навчання і спеціалізації студентів. Кожен крок у процесі роботи з навчальним планом повинен перевірятися на коректність і відповідність всім введеним обмеженням. Крім того, така система повинна володіти значною гнучкістю, що дозволяє оперативно вносити зміни, пов'язані з вимогами Болонського процесу.

Зручним засобом опису і програмної реалізації служить система продукцій, автономність якої робить її гнучкою і адаптується до будь-яких змін проблемної області. Вона відображає закономірності і способи побудови рішень на підставі описів ситуацій. Процедури перетворення опису в асинхронному вигляді дозволяють кожну з продукцій обробляти незалежно один від одного і легко відображаються в системи операторів мов програмування.

Висновки. Функціонування формальної системи зі складання навчальних планів носить цілеспрямований характер. Типовим актом є рішення задачі планування шляху досягнення потрібної мети з деякої фіксованої початкової ситуації. При цьому можливий пошук рішення, як в просторі станів, так і в просторі задач. Формалізація задачі складання навчальних планів є етап побудови експертної системи динамічного типу для створення електронної системи управління навчальним планом.

ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ОЦІНКИ ЯКОСТІ КОДУ ПРОГРАМИ

Анотація. Описано що для створення ефективних навчальних систем у якості тренувального асистента варто створити таку систему штучного інтелекту, яка дозволить аналізувати код, оперативно виявляти помилки та виробляти рекомендації покращення коду.

Ключові слова. експериментальних систем, систему штучного інтелекту, рекурентні типи мереж, згорткову мережу, програмне забезпечення

Вступ. На сьогоднішні освітні установи усе частіше використовують різноманітні технології для автоматизації та удосконалення навчального процесу. Багато аспектів уже було удосконалено, але у даній області ще є невирішені задачі. Однією з таких задач є оцінка якості коду програми. Можна покращити якість навчання та звільнити навчальний персонал від втрати великої кількості часу шляхом автоматизації рутинного тренінгу студентів. Цю задачу краще вирішувати за допомогою створення інструментарію, який дозволить аналізувати код, оперативно виявляти помилки та виробляти рекомендації щодо покращення коду. Створити даний засіб можна за допомогою таких засобів штучного інтелекту як нейронні мережі.

Основний зміст роботи. Оцінка якості коду може здійснюватися як засобами штучного інтелекту, так і стандартними засобами програмування. Звичайні засоби програмування використовуються такими інструментами як літери, синтаксичні та лексичні аналізатори тощо. Але такий аналіз здатен вказувати лише на синтаксичні та лексичні помилки, а логічні помилки рідко входять в область аналізу даних інструментів, бо реалізація такої задачі є досить складною. На противагу, засоби штучного інтелекту, є більш гнучким інструментом та дозволяють вирішити ширший спектр задач, пов'язаний з оцінюванням якості коду завдяки їх здатності навчатися та виділяти закономірності. До того ж існує проблема суб'єктивної думки та різних систем оцінювання викладачів. А штучний інтелект не має власної думки. Його думка – це середнє значення, отримане з навчальної вибірки.

Код програми – це по суті свій текст, а тест – це така структура, яка може бути проаналізована засобами штучного інтелекту. Для аналізу тексту зазвичай використовуються такі типи нейронних мереж як згорткові та рекурентні. Згорткові мережі частіше використовують для знаходження залежностей між словами, а рекурентні для знаходження довгих залежностей, наприклад, в межах одного речення. Також поширеною практикою є комбінування цих двох типів мереж для аналізу тексту.

Виходячи з цього, можна розробити програмне забезпечення, розраховане на багато користувачів, яке зможе використовувати дані типи нейронних мереж та зможе навчатися на прикладах, які вводять різні користувачі.

На сьогодні відомі декілька експериментальних систем, які використовують згорткові або рекурентні типи мереж та виконують аналіз коду. Серед них DeepReview [1] – використовує згорткову мережу, DeepCodeReviewer [2] та система для оцінки коду, описана у статті [3] – використовують рекурентну LSTM мережу. Але ці засоби не ідеальні та ще не набули широкого поширення, до того ж вони більше орієнтовані на роботу з кодом, а не на навчання, хоча дана концепція потенційно може бути успішно застосована з освітньою метою.

Висновки. У висновку можна сказати, що для створення ефективних навчальних систем у якості тренувального асистента варто створити таку систему штучного інтелекту, яка дозволить аналізувати код, оперативно виявляти помилки та виробляти рекомендації щодо покращення коду. Використання штучного інтелекту може зробити навчальний процес більш ефективним та зекономити час навчальному персоналу. Також штучний інтелект може гарантувати достатній рівень точності за умови його якісного навчання і також зменшення упередженості оцінки, спричиненої суб'єктивною думкою оцінювача.

ПЕРЕЛІК ПОСИЛАНЬ:

1. LI, Hengyi; SHI, Shuting; THUNG, Ferdian; HUO, Xuan; XU, Bowen; LI, Ming; and LO, David. DeepReview: Automatic code review using deep multi-instance learning. (2019). *Advances in knowledge discovery and data mining: 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17: Proceedings*. 11440, 318-330. Research Collection School Of Information Systems. URL: https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=5349&context=sis_research (Last accessed: 12.04.2021).

2. Gupta, A., Sundaresan, N. Intelligent code reviews using deep learning. KDD'18 Deep Learning Day, August 2018, London, UK. URL: https://www.kdd.org/kdd2018/files/deep-learning-day/DLDay18_paper_40.pdf (Last accessed: 18.04.2021).

3. Rahman, M.M., Watanobe, Y., Nakamura, K. Source Code Assessment and Classification Based on Estimated Error Probability Using Attentive LSTM Language Model and Its Application in Programming Education. *Appl. Sci.* 2020, 10, 2973. URL: <https://www.mdpi.com/2076-3417/10/8/2973> (Last accessed: 15.04.2021).

ВІДОМОСТІ ПРО АВТОРІВ

Азюковський Олександр Олександрович – к.т.н., професор, перший проректор НТУ «Дніпровська політехніка», м. Дніпро

Алексєєв Михайло Олександрович – д.т.н., професор, декан факультета інформаційних технологій НТУ «Дніпровська політехніка», м. Дніпро

Берднік Михайло Геннадійович – к.т.н., доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Бородай Валерій Анатолійович – к.т.н., доцент кафедри електропривода НТУ «Дніпровська політехніка», м. Дніпро

Бойко Олег Олександрович – старший викладач кафедри кіберфізичних та інформаційно-вимірювальних системи НТУ «Дніпровська політехніка», м. Дніпро

Буслов Дмитро Юрійович – студент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Випанасенко Станіслав Іванович – д.т.н., професор, завідувач кафедри електроенергетики НТУ «Дніпровська політехніка», м. Дніпро

Галушко Олег Михайлович – к.т.н., доцент кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро

Гнатушенко Володимир Володимирович – д.т.н., професор, завідувач кафедри інформаційних технологій та комп'ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро

Гненний Ігор Олексійович – магістр кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Голінько Олександр Васильович – магістр кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Горобець Вячеслав Сергійович – аспірант кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Гресь Наталія Леонідівна – к.т.н., доцент кафедри обліку і аудиту НТУ «Дніпровська політехніка», м. Дніпро

Гусєв Олег Юрійович – к.т.н., професор кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро

Гуліна Ірина Григорівна – к.т.н., доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Дмитрієва Ірина Сергіївна – к.т.н., доцент кафедри інформаційних технологій і систем, Національна металургійна академія України, м. Дніпро

Дмитровська Мар'яна Михайлівна – магістр кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Додатко Кирило Вікторович – магістр кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Дрешпак Наталія Станіславівна – к.т.н., доцент кафедри електротехніки НТУ «Дніпровська політехніка», м. Дніпро

Єровенко Поліна Сергіївна – студентка кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Єфременко Дмитро Костянтинович – магістр кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Желдак Тимур Анатолійович – к.т.н., доцент, завідувач кафедри системного аналізу та управління НТУ «Дніпровська політехніка», м. Дніпро

Іванченко Олег Васильович – к.т.н., доцент кафедри транспортних технологій та міжнародної логістики, Університет митної справи та фінансів, м. Дніпро

Ігнашев Василь Олександрович – магістр кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Іщук Павло Олександрович – асистент кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Кабак Леонід Віталійович – к.т.н., доцент кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Ковальов Олександр Робертович – старший викладач кафедри електроенергетики НТУ «Дніпровська політехніка», м. Дніпро

Кожевніков Антон Вячеславович – к.т.н., доцент кафедри кіберфізичних та інформаційно-вимірювальних систем НТУ «Дніпровська політехніка», м. Дніпро

Корнієнко Валерій Іванович – д.т.н., професор, завідувач кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро

Корнієнко Дмитро Ігорович – магістр кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Куваєв Володимир Михайлович – д.т.н., професор кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Курило Максим Васильович – магістр кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Мартиненко Андрій Анатолійович – старший викладач кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Макуха Деніс Юрійович – студент кафедри електропривода НТУ «Дніпровська політехніка», м. Дніпро

Массалітін Дмитро Дмитрович – магістр кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Мацуй Анатолій Миколайович – к.т.н., доцент кафедри автоматизації виробничих процесів Центральноукраїнський національний технічний університет, м. Кропивницький

Мацюк Сергій Михайлович – асистент кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро

Мединський Антон Геннадійович – магістр кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Мещеряков Леонід Іванович – д.т.н., професор кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Мірошніченко Володимир Олександрович – магістр кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Мороз Борис Іванович – д.т.н., професор кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Моссур Данило Євгенійович – магістр кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Нестерова Ольга Юрійвна – к.п.н., доцент кафедри перекладу НТУ «Дніпровська політехніка», м. Дніпро

Новицький Ігор Валерійович – д.т.н., професор кафедри системного аналізу НТУ «Дніпровська політехніка», м. Дніпро

Новодранова Наталія Олександрівна – аспірант кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Нугуманов Максим Олександрович – студент кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Одновол Микола Миколайович – доцент кафедри системного аналізу НТУ «Дніпровська політехніка», м. Дніпро

Палаш Кшистоф – магістр, Технічний університет «Краківська гірничо-металургійна академія ім. Станіслава Сташиця» (AGH), м. Краків, Польща

Панасейко Ганна Миколаївна – магістр кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Приходько Віра Володимирівна – к.т.н., доцент кафедри вищої математики НТУ «Дніпровська політехніка», м. Дніпро.

Приходченко Сергій Дмитрович – к.т.н., доцент кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Поврожнік Томаш – магістр, Технічний університет «Краківська гірничо-металургійна академія ім. Станіслава Сташиця» (AGH), м. Краків, Польща

Потемпа Михал – магістр-інженер, керівник фірми “GeoBit”, Хшанув, Польща.

Пужь Марта – магістр, Технічний університет «Краківська гірничо-металургійна академія ім. Станіслава Сташиця» (AGH), м. Краків, Польща

Реута Олександр Васильович – к.т.н., доцент кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Рудь Вячеслав Васильович – магістр кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Родна Катерина Станіславівна – асистент кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Рябічев Олег Олегович – магістр кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Рябокоть Надія Іванівна – студентка кафедри системного аналізу і управління НТУ «Дніпровська політехніка», м. Дніпро

Сергієнко Андрій Олександрович – студент кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Сердюк Владислав Юрійович – магістр кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Сироткіна Олена Ігоревна – к.т.н., доцент кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Ситник Роман Сергійович – магістр кафедри програмного забезпечення комп’ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Слесарєв Володимир Вікторович – д.т.н., професор кафедри системного аналізу та управління НТУ «Дніпровська політехніка», м. Дніпро

Спирінцев В'ячеслав Васильович – к.т.н., доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Степанов Олександр Борисович – магістр кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Сябро Іван Володимирович – магістр кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Таргунаков Дмитро Сергійович – магістр кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Тимофєєв Дмитро Сергійович – старший викладач кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро

Тихоню Ілля Вікторович – магістр кафедри інформаційних технологій і систем Національна металургійна академія України, м. Дніпро

Трофименко Денис Олександрович – магістр кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Удовик Ірина Михайлівна – к.т.н., доцент, завідувачка кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Уланова Наталія Петрівна – к.т.н., доцент кафедри вищої математики НТУ «Дніпровська політехніка», м. Дніпро

Циганок Євгеній Дмитрович – магістр кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Харь Альона Тарасівна – асистент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Черечеча Ірина Володимирівна – студентка кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Шевцова Ольга Сергіївна – асистент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Ширін Артем Леонідович – к.т.н., доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Шишацький Олександр Олександрович – магістр кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Шнайдерман Софія Володимирівна – студентка кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Шпорта Анна Григорівна – асистент кафедри вищої математики НТУ «Дніпровська політехніка», м. Дніпро

Яриловець Віктор Ігорович – магістр кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

Якунін Анатолій Олександрович – д.т.н., старший науковий співробітник, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро

ЗМІСТ

Розділ 1 МІЖНАРОДНЕ СПІВРОБІТНИЦТВО У СФЕРІ ОСВІТИ, НАУКИ І ВИРОБНИЦТВА.....	3
1. Алексєєв М.О., Удовик І.М., Дмитровська М.М., Поврожнік Т. МЕТОДИ, АЛГОРИТМИ ТА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПОКРАЩЕННЯ ПРОЦЕСІВ ПЕРЕДАЧІ ДАНИХ РАДІОСИГНАЛАМИ.....	3
2. Ширін А.Л., Потемпа М. ПЕРСПЕКТИВИ РОЗВИТКУ МІЖНАРОДНОЇ СПІВПРАЦІ ВНЗ З МАЛИМ ТА СЕРЕДНІМ БІЗНЕСОМ.....	10
3. Уланова Н.П., Одновол М.М., Приходько В.В. ЯКІСТЬ ОСВІТИ – ГАРАНТІЯ ЕФЕКТИВНОСТІ НАВЧАЛЬНОГО ЗАКЛАДУ.....	13
4. Мещеряков Л.І., Галушко О.М., Курило М.В., Палаш К. УДОСКОНАЛЕННЯ ПРОЦЕСУ ПЕРЕДАЧІ ФАЙЛІВ БЕЗДРОТОВИМ СПОСОБОМ МІЖ ПРИСТРОЯМИ З РІЗНИМИ ОПЕРАЦІЙНИМИ СИСТЕМАМИ НА БАЗІ ТЕХНОЛОГІЇ WEBRTC.....	18
5. Мороз Б.І., Мартиненко А.А., Мірошніченко В.О., Пужьо М. ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ЗНАХОДЖЕННЯ МІСЬКИХ ОСТРОВІВ ТЕПЛА ЗА ДОПОМОГОЮ СУПУТНИКОВИХ ЗНІМКІВ В ІНФРАЧЕРВОНОМУ ТЕПЛОВОМУ ДІАПАЗОНІ.....	22
6. Удовик І.М., Харь А.Т. АНАЛІЗ ВПЛИВУ ДИСТАНЦІЙНОГО НАВЧАННЯ НА ПРОЦЕС ФОРМУВАННЯ МЕТА SKILLS.....	29
Розділ 2 ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У СФЕРІ ОСВІТИ, НАУКИ І УПРАВЛІННЯ ВИРОБНИЦТВОМ.....	34
7. Алексєєв М.О., Голінько О.В. МЕТОДИ, АЛГОРИТМИ ТА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДІАГНОСТИКИ ПРАЦЕЗДАТНОСТІ СТАЦІОНАРНИХ ТЕРМОКАТАЛІТИЧНИХ ГАЗОАНАЛІЗАТОРІВ.....	34
8. Гусєв О.Ю., Кожевніков А.В., Тимофєєв Д. С., Яриловець В.І. ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ЦИФРОВОГО МОДЕЛЮВАННЯ ТА РОЗРАХУНКУ І ВІЗУАЛІЗАЦІЇ ФОРМУЕМИХ В СКЛАДНИХ ОБ'ЄКТАХ ТЕМПЕРАТУРНИХ ПОЛІВ.....	37
9. Куваєв В.М., Спирінцев В.В., Корнієнко Д.І. АЛГОРИТМИ ТА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ РОЗПІЗНАВАННЯ ОБЛИЧЧЯ НА ОСНОВІ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ.....	41
10. Бердник М.Г., Харь А.Т., Іщук П.О. ПРАКТИЧНА ЕФЕКТИВНІСТЬ ВИРІШЕННЯ ЗАДАЧІ ДОСЛІДЖЕННЯ ТЕМПЕРАТУРНИХ ПОЛІВ ПЛАСТИН МЕТОДОМ КІНЦЕВИХ РІЗНИЦЬ.....	45
11. Мороз Б.І., Гуліна І.Г., Сердюк В.Ю. МЕТОДИ ТА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДЛЯ ВИВЧЕННЯ ПРАВИЛ ДОРОЖНЬОГО РУХУ ЗА ДОПОМОГОЮ ВПРОВАДЖЕННЯ ЕЛЕМЕНТІВ ІГРОФІКАЦІЇ.....	49
12. Алексєєв М.О., Шевцова О.С., Моссур Д.Є. ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ОПТИМІЗАЦІЇ СИНХРОНІЗАЦІЇ, КЕРУВАННЯ ТА ВІДТВОРЕННЯ СТАНІВ СПИСКІВ З ПОСТОРІНКОВИМ	

ВІДТВОРЕННЯМ.....	54
13. Мороз Б.І., Мартиненко А.А., Додатко К.В. МЕТОДИ, АЛГОРИТМИ ТА ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДЛЯ ДОСЛІДЖЕННЯ ТА АНАЛІЗУ РИЗИКІВ ОПЕРАЦІЙ НА ФОНДОВОМУ РИНКУ.....	60
14. Kabak L., Buslov D. USING COMPUTER TECHNOLOGY FOR BLIND PEOPLE.....	64
15. Сироткіна О.І., Родна К.С., Мединський А.Г. ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПРОГНОЗУВАННЯ ARIMA ДЛЯ ЧАСОВИХ РЯДІВ....	67
Розділ 3 ПРОГРАМНІ ЗАСОБИ УПРАВЛІННЯ, ЗБОРУ, ОБРОБКИ І ПЕРЕДАЧІ ІНФОРМАЦІЇ.....	73
16. Удовик І.М., Мацюк С.М., Ігнашев В.О. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ОБРОБКИ ЗАПИТІВ ПОШУКОВИХ СИСТЕМ ТА ЇХ ФУНКЦІЇ У КОРПОРАТИВНІЙ РОЗВІДЦІ.....	73
17. Якунін О.А., Кабак Л.В., Циганок Є.Д. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ РОЗШИРЕННЯ ФУНКЦІЙ САПР МОДЕЛЕЙ РЕАЛЬНИХ ОБ'ЄКТІВ В РЕЖИМІ ДОПОВНЕНОЇ РЕАЛЬНОСТІ.....	77
18. Новицький І.В., Слесарєв В.В. ПРОГРАМНА РЕАЛІЗАЦІЯ ОЦІНКИ НАДІЙНОСТІ СИСТЕМИ.....	81
19. В.І. Корнієнко, Мацюк С.М., Рудь В.В. МЕТОДИ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ БІОНІЧНОГО ХАПАННЯ З УНИКНЕННЯМ ПЕРЕШКОД ЗА ДОПОМОГОЮ ДЕРЕВ ОКТАНТІВ ТА ГЛИБОКОГО НАВЧАННЯ.....	85
20. Алексєєв М.О., Реута О.В., Сябро І.В. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ФРЕЙМВОРКА ПІДМІНИ ДАНИХ ДЛЯ ТЕСТУВАННЯ FLUTTER ДОДАТКІВ.....	93
21. Іванченко О.В., Рябічев О.О. ПРОЄКТУВАННЯ ХМАРНОГО АПАРАТНО-ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ 3D-ДРУКУ.....	99
22. Алексєєв М.А., Спирінцев В.В., Ситник Р.С. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЗАСОБУ FLUTTER ПРИ РОЗРОБЦІ МОБІЛЬНИХ ДОДАТКІВ.....	106
23. Удовик І.М., Горобець В.С., Степанов О.Б. МЕТОДИ, АЛГОРИТМИ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ФРЕЙМВОРКІВ ЗГІДНО З ПОТРЕБАМИ ПРОЄКТУ.....	109
24. Гнатушенко В.В., Гненний І.О., Єфременко Д.К. ДОСЛІДЖЕННЯ ВЗАЄМОДІЇ ФРЕЙМВОРКІВ ТА ПРОГРАМНОЇ ПЛАТФОРМИ NODE.JS З МЕТОЮ РОЗРОБКИ ВИСОКО ДИНАМІЧНИХ ВЕБЗАСТОСУНКІВ З ВИКОРИСТАННЯМ JAVASCRIPT.....	114
25. Приходченко С.Д., Шевцова О.С., Нугуманов М.О. ПРОСЛУХОВУВАННЯ ТЕЛЕФОНІВ ДОДАТКАМИ З МЕТОЮ ТАРГЕТУВАННЯ РЕКЛАМИ.....	123
26. Рябокоть Н.І., Родна К.С., Нугуманов М.О. СТВОРЕННЯ	

ПРОГРАМИ ДЛЯ ШВИДКОГО ФОРМУВАННЯ ЗВІТІВ	126
27. Ширін А.Л., Сергієнко А.О. ПЕРСПЕКТИВИ СТВОРЕННЯ ІДЕАЛЬНОЇ МОВИ ПРОГРАМУВАННЯ.....	129
Розділ 4 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ СИСТЕМ ОБРОБКИ ІНФОРМАЦІЇ ДЛЯ ВИРІШЕННЯ ЗАВДАНЬ ОСВІТИ, НАУКИ І УПРАВЛІННЯ ВИРОБНИЦТВОМ.....	132
28. Мороз Б.І., Шишацький О.О., ВИКОРИСТАННЯ ОПТИЧНИХ ДЕТЕКТОРІВ ДОРОЖНЬОГО РУХУ У СИСТЕМАХ АДАПТИВНОГО КЕРУВАННЯ СВІТЛОФОРАМИ.....	132
29. Мещеряков Л.І., Випанасенко С.І., Дрешпак Н.С., Трофименко Д.О. РОЗРОБКА ТА ОПТИМІЗАЦІЯ МЕТОДІВ ТА АЛГОРИТМІВ АВТОМАТИЧНОГО УПРАВЛІННЯ РОЗГОРНУТИМИ КОНТЕЙНЕРІЗОВАНИМИ ВЕБ-ДОДАТКАМИ ТА СЕРВІСАМИ.....	136
30. Куваєв В.М., Бойко О.О., Новодранова Н.О. КЕРУВАННЯ ДОВЖИНОЮ ДРІБНОСОРТНОГО ПРОКАТУ ЗА РАХУНОК ЗБІЛЬШЕННЯ ДОВЖИНИ ЙОГО СЕРЕДНЬОЇ ДІЛЯНКИ.....	142
31. Мацуй А.М. ТЕОРЕТИЧНІ ОСНОВИ ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНОЇ СИСТЕМИ ОЦІНЮВАННЯ РОБОТИ КУЛЬОВИХ МЛИНІВ З МЕТОЮ ЕНЕРГОЗБЕРЕЖЕННЯ.....	144
32. Азюковський О.О., Панасейко Г.М. ПЛАНУВАННЯ ШЛЯХУ ДЛЯ НАВІГАЦІЇ МОБІЛЬНИХ РОБОТІВ З УНИКНЕННЯМ ПЕРЕШКОД ЗА ДОПОМОГОЮ ДЕРЕВА ОКТАНТІВ	151
33. Мещеряков Л.І., Одновол М.М., Гресь Н.Л., Массалітін Д.Д. ЗАСТОСУВАННЯ МЕТОДУ ЛОГІСТИЧНОЇ РЕГРЕСІЇ ДЛЯ КЛАСИФІКАЦІЇ НАУКОВИХ ТЕКСТІВ У НАВЧАЛЬНОМУ АСИСТЕНТІ.....	157
34. Мороз Б.І., Спирінцев В.В., Таргунаков Д.С. ІНФОРМАЦІЙНА ПІДТРИМКА ПРИЙНЯТТЯ РІШЕНЬ В КОМПЛЕКСНИХ ТЕХНОЛОГІЧНИХ ПРОЦЕСАХ ВИРОБНИЦТВА.....	163
35. Бородай В.А., Нестерова О.Ю., Ковальов О.Р. СПОСІБ СИМЕТРУВАННЯ АКТИВНОГО НАВАНТАЖЕННЯ В ЧОТИРИПРОВІДНИХ МЕРЕЖАХ ЗМІННОГО СТРУМУ.....	166
36. Тихоню І.В., Дмитрієва І.С. ДОСЛІДЖЕННЯ СТАНУ ТОВСТОСТІННИХ ВТУЛОК ПІСЛЯ ДОРНУВАННЯ.....	169
37. Макуха Д.Ю., Бородай В.А. ТЕХНІКО-ЕКОНОМІЧНА ЕВОЛЮЦІЯ ГОЛОВНОГО ПРИВОДА ПРОМИСЛОВИХ ПРАЛЬНИХ МАШИН.....	171
РОЗДІЛ 5 ПРОБЛЕМИ ДИСТАНЦІЙНОЇ ОСВІТИ	174
38. Уланова Н.П., Шпорта А.Г., Приходько В.В. ДОСВІД ВИКЛАДАННЯ ВИЩОЇ МАТЕМАТИКИ В НТУ «ДНІПРОВСЬКА ПОЛІТЕХНІКА» В УМОВАХ ПАНДЕМІЇ COVID-19.....	174
39. Гуліна І.Г., Шнайдерман С.В. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ОСВІТІ.....	178
40. Ширін А.Л., Єровенко П.С. ШТУЧНИЙ ІНТЕЛЕКТ В МИСТЕЦТВІ ТА ТВОРЧОСТІ.....	182
41. Приходченко С.Д., Шпорта А.Г. ФОРМАЛІЗАЦІЯ ЗАДАЧІ	

СКЛАДАННЯ НАВЧАЛЬНИХ ПЛАНІВ У ВИЩОМУ НАВЧАЛЬНОМУ ЗАКЛАДІ.....	185
42. Удовик І.М, Желдак Т. А. Черечеча І.В. ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ОЦІНКИ ЯКОСТІ КОДУ ПРОГРАМИ.	188
ВІДОМОСТІ ПРО АВТОРІВ.....	190

Наукове видання

**ПРОБЛЕМИ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
В ОСВІТІ, НАУЦІ ТА ПРОМИСЛОВОСТІ**

XV міжнародна конференція

м. Дніпро
15 – 17 грудня 2020 року

Збірник наукових праць
№ 5

Відповідальний за випуск Л.І. Мещеряков

Видано в редакції авторів публікацій.

Підписано до друку __.__.2021. Формат 30x42/4.
Папір офсетний. Ризографія. Ум. друк. арк. 8,4.
Обл.-вид. арк. 8,4. Тираж 25 пр. Зам. №

Підготовлено до друку та видруковано
в НТУ «Дніпровська політехніка»
Свідоцтво про внесення до Державного реєстру ДК № 1842 від 11.06.2004.

49005, м. Дніпро, просп. Д. Яворницького, 19.