

Міністерство освіти і науки України
Національна академія наук України
Південний науковий центр НАН та МОН України
Чорноморський національний університет імені Петра Могили
Первинна профспілкорова організація ЧНУ імені Петра Могили
Інститут української археографії та джерелознавства ім. М. С. Грушевського НАНУ
Державний архів Миколаївської області
ДУ «Національний науковий центр радіаційної медицини НАМН України»
Державний аграрний університет Молдови (Кишинів)
Університет гуманітарних та природничих наук ім. Яна Длугоша (Польща)
Університет імені Адама Міцкевича (Польща)
Leipzig University of Applied Sciences (Німеччина)
Ca` Foscari University, Venice (Італія)



ОЛЬВІЙСЬКИЙ ФОРУМ – 2022:
стратегії країн Причорноморського регіону
в геополітичному просторі

XVI Міжнародна наукова конференція
23–26 червня 2022 р., м. Миколаїв

ТЕЗИ

Комп'ютерна інженерія.
Інтелектуальні інформаційні системи.
Моделі, методи та засоби програмної інженерії.
Автоматизація та комп'ютерно-інженерні технології



Миколаїв
2022

Ольвійський форум – 2022 : стратегії країн При-чорноморського регіону в геополітичному просторі : XVI Міжнар. наук. конф. 23–26 червня 2022 р., м. Миколаїв : тези доп. : Комп’ютерна інженерія.Інтелектуальні інформаційні системи. Моделі, методи та засоби програмної інженерії. Автоматизація та комп’ютерно-інженерні технології / Чорном. нац. ун-т ім. Петра Могили. – Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2022. – 140 с.

Секція КОМП'ЮТЕРНІ НАУКИ

Підсекція: КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

УДК 62-503.5

Басв В. О.,
студент факультету комп'ютерних наук,
Пузирьов С. В.,
канд. фіз.-мат. наук, доцент,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

РОЗРОБКА МОНІТОРИНГОВОЇ МЕРЕЖИ НА ОСНОВІ МОДУЛІВ ESP8266 WEMOS

Сучасні системи «розумних» будинків можуть містити у собі безліч модулів, які виконують різні функції. Під час проектування такої системи основною проблемою є організація спілкування між модулями та отримання актуальної інформації щодо стану системи в цілому. З цієї причини у межах однієї системи рекомендується використовувати пристрої одного виробника. Розробники пристроїв систем інтернету речей пропонують свої власні рішення для моніторингу показників пристроїв та синхронізації пристроїв між собою. Завдяки унікальності даних рішень та якості програмних розробок (стабільність підключення, зручність користувацького інтерфейсу, тощо), виробники роблять націнку на свою продукцію.

У рамках даної роботи представлений Proof Of Concept (PoC) програмного додатку для моніторингу стану системи модулів інтернету-речей та організації спілкування між модулями. Для моделювання системи інтернету речей були використані 3 модулі ESP8266 Wemos. ESP8266 – мікроконтролер компанії Espressif Systems, який був розроблений у 2014 році. Wi-Fi модуль мікроконтролеру реалізує стандарти IEEE 802.11b/g/n. Наявна підтримка режимів AP та Station. Має 32-бітний процесор Tensilica L106, який працює на частоті 80 МГц. Дана модель була обрана для виконання роботи, оскільки можливостей даного рішення вистачає для навчальних цілей та його ціна є прийнятною. Мікроконтролери були розділені за функціональністю. Типи мікроконтролерів:

- для вимірювання температури;

- для вимірювання рівня освітленості;
- для збереження даних та контролю стану системи.

Розробка прошивки для мікроконтролерів була виконана за допомогою середовища розробки ArduinoIDE. У якості топології локальної мережі було прийнято рішення використати mesh мережу.

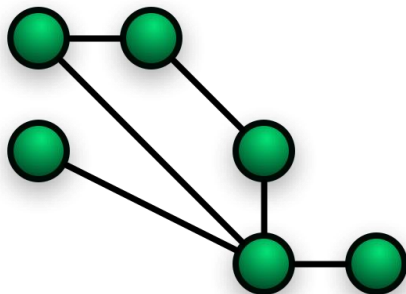


Рисунок 1 – Mesh мережа

Такий тип мереж дозволяє динамічно змінювати топологію чи місце розташування пристроїв. Завдяки тому, що маршрути між пристроями будуються автоматично, є можливість додавати чи відключати пристрої від мережі у процесі роботи та використовувати найбільш оптимальні маршрути для відправки повідомлень між пристроями. Спілкування між контролерами було виконано за допомогою бібліотеки PainlessMesh. PainlessMesh – це бібліотека, яка надає спрощений інтерфейс для розробки простої mesh мережі на основі мікроконтролерів ESP8266 чи ESP32. Завдяки цій бібліотеці розробник не повинен налаштовувати мережу на низькому рівні та піклуватись про стан підключення пристроїв. Для обміну даними у бібліотеці PainlessMesh використовується JSON формат. Передача даних у форматі JSON спрощує процес налагодження зв'язку між пристроями, але може бути занадто важкою для пам'яті пристрою та потужності процесору. Альтернативним рішенням – є обмін даними у вигляді байтів. Для цього прошивка кожного мікроконтролера повинна мати структуру, яка використовується для обміну даними. Таким чином, при отриманні масиву байт від іншого пристрою, байти можна буде представити у вигляді структури. Обмін даними у вигляді байтів є більш продуктивним, оскільки витрачає менше місця на пристроях. Недоліками обміну за допомогою масиву байтів є:

- проблеми з використанням у випадку різних архітектур процесорів (порядок байтів);
- складність відлагодження.

Для передачі даних між пристроями та програмним додатком моніторингу стану мережі був використаний протокол MQTT. MQTT – це легкий, компактний та відкритий протокол обміну даними. Він був створений для передачі даних на відстані, де наявне обмеження пропускнувості каналу. Протокол MQTT працює на прикладному рівні поверх TCP/IP. Обмін повідомленнями у протоколі MQTT здійснюється між клієнтом та брокером повідомлень. Видавець повідомлень відправляє дані на брокер MQTT, вказуючи в повідомленні певну тему (topic). Слухачі можуть отримувати різні дані від багатьох видавців залежно від підписки на відповідні теми.

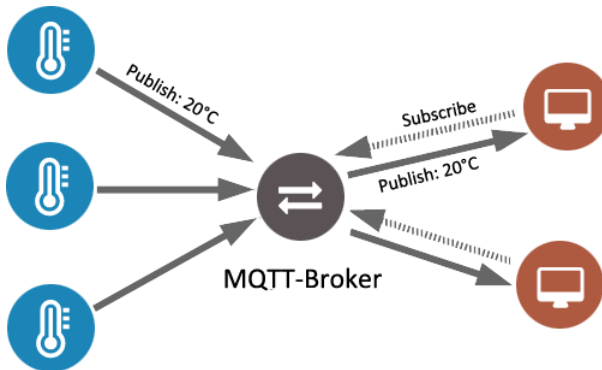


Рисунок 2 – Схема відправки повідомлень протоколом MQTT

У рамках даного проекту пристрої мережі інтернету речей через MQTT канал передають дані з поточними показниками на сервер, де знаходиться додаток на основі SpringBoot. SpringBoot – набір інструментів, які значно спрощують створення серверної частини web-додатку. У набір інструментів входять:

- розробка контролерів;
- підключення до бази даних;
- налаштування журналювання;
- створення системи безпеки та аутентифікації.

Головною перевагою використання SpringBoot у проекті є механізм ін'єкції залежностей. За допомогою даного механізму усі необхідні поля у сервісах автоматично ініціалізуються під час запуску додатка. Для написання серверного додатку була використана мова Java версії 1.8.

Таким чином, була створена система моніторингу та обміну повідомленнями у mesh мережі на основі ESP8266 Wemos та серверного додатку SpringBoot. За допомогою даної системи є можливість переглядати поточний стан системи з будь-якої точки світу з доступом в Інтернет. Стабільність даного рішення забезпечується використанням mesh топології та протоколу обміном повідомленнями MQTT.

УДК 629.1.07

Керекеслер М. О.,
студент факультету комп'ютерних наук,
Бурлаченко І. С.,
старший викладач кафедри КІ,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

АНАЛІЗ ХАРАКТЕРИСТИК КОМПЛЕКСУЮЧИХ ДЛЯ БЕЗПІЛОТНИХ НАЗЕМНИХ ТРАНСПОРТНИХ ЗАСОБІВ

Наземні мобільні роботи найчастіше можуть використовуватися в сегменті логістичних транспортних перевезень. Ефективність використання безпілотних наземних транспортних засобів (БНТЗ) досить велика, так як БНТЗ можуть знизити вартість доставки в порівнянні з використанням людей, а значить, знизиться собівартість надання послуги, що послужить додатковим стимулом до впровадження таких пристроїв. БНТЗ доведеться конкурувати з транспортуванням за допомогою безпілотних літальних транспортних засобів (БЛТЗ), що літають. Перед доставкою літаючими БЛТЗ наземне транспортування має явну перевагу, що ні вантаж, ні транспортний засіб не призведуть до травматичних наслідків для оточуючих, якщо виникне аварійна ситуація. Однак БНТЗ можуть зустрітися з проблемами крадіжки та вандалізму.

Науково-дослідні установи (НДУ) починають випробовувати БНТЗ на вулицях міст. Розпочинається комерційне застосування БНТЗ. Одне з непростих завдань для створення таких БНТЗ, – це розробка алгоритму, який дозволяв би роботу знаходити свій шлях у потоці людей, але так, щоб виключити ризик зіткнення з ними, принаймні з вини робота. БНТЗ оснащені датчиками, у тому числі камерами з машинним зором, радарми та ультразвуковими датчиками, які виявляють тверді об'єкти, такі як бордюри та стіни.

Частіше всього НДУ самі розробляють або замовляють спеціальні компоненти, які необхідні для роботи БНТЗ. Розроблюються власні материнські плати обчислювача, камери та сенсори. Завдяки цьому апаратна частина працюватиме ефективніше та споживати менше енергії. Перед тим як почати збирати БНТЗ необхідно визначитися яким характеристикам та якій ціні мають відповідати комплектуючі. Частіше всього необхідні відмінні комплектуючі дуже дорого коштують, тому доводиться аналізувати дешеві аналоги. Після порівняння аналогів, обирається найбільш підходящий варіант, відповідний усім необхідним критеріям, для конкретного БНТЗ. Для БНТЗ необхідно обрати такі комплектуючі: колеса та мотори для них, драйвер двигуна, мікроконтролер, мікрокомп'ютер, ультразвуковий датчик, камера, gps-модуль.

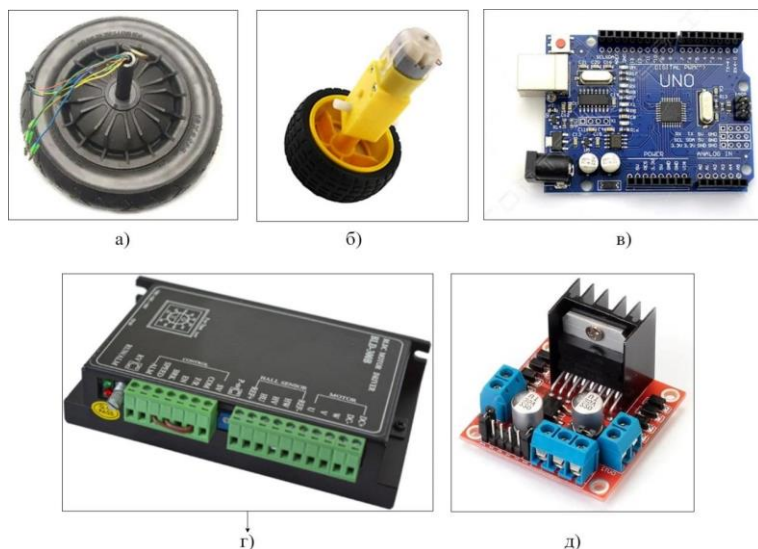


Рисунок 1 – Комплектуючі для безпілотних наземних транспортних засобів

Для руху БНТЗ необхідні колеса та мотори. Враховуючи, що сама платформа с приладами та вантаж матимуть велику вагу, то потрібно обирати колеса та мотори з достатньо великими розмірами, які зможуть витримати велике навантаження. Для цього можуть підійти мотор-колеса для електро-самокатів та гіроскутерів (рис. 1-а), які мають розміри 6,5-10 дюймів, або інші BLDC мотори. Однак такі мотор-колеса досить дорого коштують та споживатимуть багато

енергії. В якості дешевої альтернативи для первинних досліджень БНТЗ рекомендовано звичайні мотори с редуктором від Arduino 4WD Car та колеса до них, що зображені на рис.1-б. Дані мотори-редуктори дуже розповсюджені і їх легко використовувати. Вони не підійдуть для доставки реальних великих вантажів, але їх можна використовувати для перевірки працездатності БНТЗ та тестування гіпотез. Шина має гумове покриття. Діаметр колеса з шиною 6,6 см. Мотор з редуктором має такі характеристики: розміри (64,2мм × 22,5мм × 18,8мм); робоча напруга (3,0 v ~ 12.0v DC); кількість обертів в хвилину (20 ~ 100); Для переміщення БНТЗ знадобиться 4 колеса та мотора. За потребою їх кількість можна збільшити.

Для управління моторами використовуються драйвери двигуна, так як підключати мотори напряму до мікроконтролера не можна. Драйвер буде регулювати швидкість моторів за допомогою ШИМ-сигналу, який подається з мікроконтролера. Якщо б ми обрали великі мотор-колеса, то для їх управління нам знадобиться драйвер BLD-300B (рис. 1-г). Даний драйвер може забезпечувати вихідну потужність 300VA max. Також він забезпечує високий крутний момент, низький рівень шуму, низький рівень вібрації. Працює він при напрузі 18–50V. Однак для мотора-редуктора підійдуть інші моделі, наприклад, можна обрати модуль l298n (рис. 1-д). Схема модуля, що складається з двох H-мостів, дозволяє підключати до нього один біполярний кроковий двигун або одночасно два щіткові двигуни постійного струму. При цьому є можливість змінювати швидкість та напрямок обертання моторів. Управління здійснюється шляхом подання відповідних сигналів на командні входи. Основні характеристики модуля: напруга живлення логіки (5V); споживаний логікою струм (36 mA); напруга живлення двигунів (від 5V до 35V); робочий струм драйвера (2 A); піковий струм драйвера (3 A); максимальна потужність: (20 Вт при температурі 75 ° C); розміри модуля: 43,5 x 43,2 x 29,4 мм;

Модуль має роз'єми OUT1 і OUT2 для одного двигуна, та OUT3 і OUT4 для другого. Також модуль має вхід для живлення та вхід живлення логіки 5V. В модуль вбудований стабілізатор напруги, через який логіка модуля може отримувати додаткове живлення. Контакти IN1, IN2 відповідають за управління першим двигуном, а контакти IN3, IN4 відповідають за управління другим двигуном. Контакти ENA, ENB відповідають за активацію та деактивацію двигунів, також на них можна подати ШИМ-сигнал для регулювання швидкості обертання моторів.

Для управління модулями було обрано один із мікроконтролерів Arduino. Arduino зручна платформа для розробки прототипів БНТЗ.

Основними перевагами Arduino є: доступність, зручність, відкрита архітектура та програмний код, простота мови програмування. За допомогою Arduino можна управляти датчиками, різними модулями, в тому числі і драйвером двигунів. Існує безліч різновидів Arduino, однак частіше використовують Arduino UNO та Arduino Nano. Головна відмінність між ними це розмір. Arduino Nano має менші розміри та використовує кабель mini-USB, тоді як UNO використовує кабель USB. Однак обидві моделі використовують процесор ATmega 328P, тому вони можуть виконувати однакові програми. Для робота, який буде мати і так не маленькі габарити, розмір плати не має значення, тому було обрано Arduino UNO (рис. 1-в). Також Arduino можна підключити до мікрокомп'ютера через UART інтерфейс або через USB. Це дає змогу обмінюватися інформацією між Arduino та мікрокомп'ютером.

Таблиця 1 – Прибутковість від вкладення коштів за проектами

RPI model	CPU	Кількість ядер	ОЗУ	Wi-Fi
Pi 3 A+	Broadcom BCM2837B0	4x1.4 ГГц	512 МБ	802.11ac 4.2
Pi 3 B	Broadcom BCM2837	4x1.2 ГГц	1 ГБ	802.11n 4.1
Pi 3 B+	Broadcom BCM2837B0	4x1.4 ГГц	1 ГБ	802.11ac 4.2
Pi Zero W	Broadcom BCM2835	1x1 ГГц	512 МБ	802.11n 4.0
Pi 4 B	Broadcom BCM2711	4x1.5 ГГц	1 ГБ, 2 ГБ, 4 ГБ	802.11ac 5.0

Для обчислювання маршруту БНТЗ необхідний мікрокомп'ютер з потужним процесором. Для цього було обрано одну із моделей Raspberry Pi з таблиці 1. Raspberry Pi – це родина багатофункціональних мікрокомп'ютерів з компактним розміром. Пристрій має чотириядерний процесор на борту та множину інтерфейсів, починаючи від Bluetooth та Wi-Fi, закінчуючи GPIO. Завдяки інтерфейсам і мережевим протоколам, міні-ПК можна з'єднати з настільними комп'ютерами, ноутбуками та смартфонами, а мережеві протоколи дозволяють здійснювати контроль над відеореєстраторами та БЛТЗ. Raspberry Pi має відносно не велику ціну і може використовуватися для автоматизації будь-яких задач у процесі роботи БНТЗ. Також Raspberry Pi має низький поріг входження при програмуванні ПЗ. Основною мовою програмування Raspberry Pi є Python, для якого написано багато бібліотек для різних задач. Однак писати код можна і на інших мовах програмування: C, C++, C#, Ruby, Java і т.д. Raspberry Pi є доступним через невелику ціну. Raspberry Pi поставляється з 26 контактами GPIO, які дійсно дуже корисні для вбудовуваних проектів та інтерфейсного обладнання. Виходи корисні для реалізації взаємодії компонентів БНТЗ. Можна поєднати потрібні цифрові датчики разом

із великою кількістю контактів GPIO. До Raspberry Pi можна підключати різні дисплеї, модулі, датчики. Raspberry Pi постачається з швидким процесором, що дає хорошу продуктивність. Моделі Raspberry Pi відрізняються процесором та обсягом оперативної пам'яті, кількістю портів USB та наявністю модуля Wi-Fi.

УДК 004.85

Костін О. А.,
студент факультету комп'ютерних наук,
Бурлаченко І. С.,
старший викладач кафедри КІ,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

АНАЛІЗ ТЕХНОЛОГІЙ МАШИННОГО НАВЧАННЯ ДЛЯ ВІДСТЕЖЕННЯ ПОЗ ТІЛА ЛЮДИНИ

Відстеження тіла можна вважати досить зрілою технологією, оскільки є основи в таких областях, як фітнес, реабілітація, доповнена реальність, анімація, ігри, робототехніка і навіть спостереження. Застосунки для тренувань, керовані штучним інтелектом, є одними з найбільш очевидних випадків використання для оцінки пози тіла. Модель, реалізована в мобільному застосунку для смартфона, може використовувати апаратну камеру як датчик, щоб зафіксувати, як хтось виконує вправу, і провести її аналіз. Відстежуючи рух людського тіла, вправу можна розділити на фази ексцентричних і концентричних рухів, щоб проаналізувати різні кути згинання і загальну поставу. Це здійснюється шляхом відстеження ключових моментів і надання аналітики у вигляді підказок або графічного аналізу. Отримані дані можна обробляти в режимі реального часу або після певної затримки, надаючи користувачам аналітику щодо основних моделей рухів і механіки тіла.

Фізіотерапевтична галузь – це ще один випадок використання відстеження людської активності з подібними правилами застосування. В епоху телемедицини домашні консультації стають набагато гнучкішими та різноманітнішими. Технології AI уможливили більш складні способи лікування онлайн. Аналіз реабілітаційних заходів застосовує подібні концепції до застосування фітнесу, за винятком вимог до точності. Оскільки ми маємо справу з відновленням після травми, категорія за стосунків з розпізнаванням поз потрапить до категорії охорони здоров'я. Це означає, що він повинен відповідати

стандартам галузі охорони здоров'я та загальним законам про захист даних певної країни. Оцінка пози – це завдання використання моделі ML для оцінки пози людини з зображення або відео, оцінюючи просторове розташування ключових суглобів тіла (ключових точок).

OpenPose, VIBE та Adobe BodyNet є іншими основними гравцями в цій галузі. За останні п'ять років оцінка пози людини пройшла довгий шлях, але не з'явилася в багатьох програмах, оскільки більшість компаній зосередилися на тому, щоб зробити моделі поз більшими та точнішими, а не виконувати інженерну роботу, щоб зробити їх швидкими та розгорнутими скрізь. Google розробила модель, яка використовує архітектуру SOTA, але зберігає час висновку якомога меншим. Завдяки цьому модель може надавати точні ключові точки в широкому діапазоні поз, середовищ та інфраструктури. Нещодавно Google Research співпрацювала з компанією IncludeHealth, що базується в Огайо, щоб запропонувати пацієнтам дистанційну допомогу. Використовуючи MoveNet, компанія розробила інтерактивний веб застосунок, який керує пацієнтом різними процедурами за допомогою ноутбука, смартфона чи IoT пристрою з web-камерою. Вправи були практично побудовані та призначені фізіотерапевтом для перевірки рівноваги, сили та діапазону рухів.

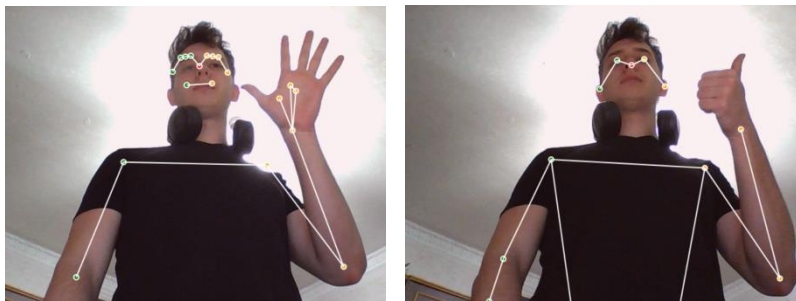


Рисунок 1 – Приклад роботи при середній частоті кадрів: 53 fps BlazePose (зліва) і 78 fps MoveNet (справа)

Оцінка пози відноситься до технік комп'ютерного зору, які визначають людські фігури на зображеннях і відео, щоб можна було визначити, наприклад, де на зображенні з'являється чийсь лікоть. Ці моделі в основному є картою суглобів тіла, які ми відстежуємо під час руху. Це робиться для того, щоб комп'ютер не тільки знаходив різницю між людиною, яка просто сидить і присідає, а й щоб розрахувати кут згинання в конкретному суглобі та визначити, чи правильно виконано рух.

Важливо пам'ятати про те, що оцінка пози лише визначає, де знаходяться ключові суглоби тіла, і не розпізнає, хто на зображенні чи відео. Моделі оцінки пози беруть оброблене зображення камери як вхід і виводять інформацію про ключові точки. Виявлені ключові точки індексуються ідентифікатором деталі з показником достовірності від 0,0 до 1,0. Оцінка впевненості вказує на ймовірність існування ключової точки в цій позиції. Пропонується еталонна реалізація двох моделей оцінки пози TensorFlow Lite: PoseNet та MoveNet: найсучасніші моделі оцінки пози, доступні в двох варіантах Lighting і Thunder. Різні суглоби тіла, виявлені моделлю оцінки пози: nose, eftEye, rightEye, leftEar, rightEar, leftShoulder, rightShoulder, leftElbow, rightElbow, leftWrist, rightWrist, leftHip, rightHip, leftKnee, rightKnee, leftAnkle, rightAnkle.

Версія MoveNet, робота якої зображена на рисунку 1, може розпізнавати одну позу з вхідного зображення. Якщо на зображенні більше однієї людини, модель разом із алгоритмом кадрівання намагатиметься зосередитися на людині, яка знаходиться найближче до центру зображення. Впроваджено розумний алгоритм обрізання, щоб підвищити точність виявлення відео. Зазначимо що, модель збільшить область, де є поза, виявлена в попередньому кадрі, щоб модель могла бачити дрібніші деталі та робити кращі прогнози в поточному кадрі. Хоча модель оцінки пози вказує вам, де знаходяться ключові точки пози, у багатьох фітнес застосунках ви можете піти далі й класифікувати позу, наприклад, чи це поза богині йоги чи поза дошки, щоб надати користувачам відповідну інформацію. Щоб спростити виконання класифікації поз, розроблено інтерфейс service Colab, який навчає вас, як використовувати MoveNet і TensorFlow Lite для навчання спеціальної моделі класифікації поз із вашого набору даних. Це означає, що якщо ви хочете розпізнати пози тренувань, все, що вам потрібно, це зібрати зображення поз, які ви хочете розпізнати, позначити їх, щоб навчити та розгорнути класифікатор поз тіла при тренуваннях. Для навчання класифікатору поз, вам потрібно підготувати зображення поз і помістити їх у структуру папок. Кожне ім'я підпапки – це ім'я класу, який ви хочете розпізнати. Потім ви можете запустити блокнот для навчання спеціального класифікатора поз і конвертувати його у формат TensorFlow Lite.

Класифікація поз моделі TensorFlow Lite дуже мала, всього близько 30 КБ. Беруться вихідні орієнтири з MoveNet, нормалізує координати пози і передає їх через кілька повністю пов'язаних шарів. Результатом моделі є список ймовірностей того, що поза є кожним із відомих типів пози.

Робота системи оцінки пози тіла складається зі збору вихідних даних і завантаження їх для обробки системою. Оскільки ми маємо справу з виявленням руху, то нам потрібно проаналізувати послідовність зображень, а не нерухому фотографію, оскільки нам потрібно витягти, як ключові точки змінюються під час шаблону руху. Після завантаження зображення система виявить та відстежить необхідні ключові моменти для аналізу. ПЗ зробить необхідні розрахунки для оцінки пози. Оцінка пози людини під час виконання вправи – це лише один приклад у фітнес-індустрії. Деякі моделі також можуть виявляти ключові точки на обличчі людини та положенні головки, що можна застосувати для гейміфікації ПЗ.

Оцінка пози людини відповідає за ту частину процесу, де фактична область контакту з продуктом невидима для камери. Наприклад, модель HPE аналізує положення рук і голови спортсменів, щоб зрозуміти, чи взяли вони гантель, чи залишили її на місці. Як навчити модель оцінки пози людини? Оцінка пози людини – це технологія машинного навчання, що означає, що вам знадобляться дані для її навчання. Оскільки оцінка пози людини виконує досить складні завдання виявлення та розпізнавання кількох об'єктів на екрані, то в якості основи використовуються нейронні мережі. Індивідуальний графік занять потрібно розробляти в залежності від цілей і потреб користувача. Також потрібно врахувати його фізичні здібності.

Таблиця 1 – Початкові дані програми спортивних тренувань для навчання моделі

	Тренування 1	Тренування 2	Тренування 3
Підхід 1	60% від максимуму	65% від максимуму	70% від максимуму
Підхід 2	50% від максимуму	60% від максимуму	60% від максимуму
Підхід 3	45% від максимуму	55% від максимуму	55% від максимуму
Підхід 4	40% від максимуму	45% від максимуму	50% від максимуму
Підхід 5	35% від максимуму	40% від максимуму	45% від максимуму

Як ціль взято збільшення кількості виконаних вправ з часом. Фізичні здібності будуть визначатись під час алгоритму. Для цього використовується простий алгоритм:

1) визначається скільки максимально зараз може робити вправ користувач;

2) в залежності від цього формуються підходи та кількість вправ. Їх повинно бути значно менше ніж максимум, щоб користувач міг їх виконати. Та достатньо, щоб давати навантаження на мускулатуру;

3) розробляється програма на три дні;

4) робиться перевірка скільки максимум може робити вправ користувач і алгоритм починається з кроку 2.

У зв'язку з недостатньою кваліфікацією для розробки персональних програм, для початкових даних у таблиці 1 наведена вже готова програма тренувань з застосунок «200 віджимань».

Використовується генетичний алгоритм для селекції оптимальної програми тренувань. Алгоритм роботи застосунок виглядає наступним чином:

1) є початкова популяція (взятий вище алгоритм). Кількість тренувань буде невеликою, 5 варіантів. Вони будуть випадково розподілені серед користувачів;

2) мутація – випадкове збільшення чи зменшення проценту необхідних вправ за підхід;

3) селекція – обираються ті програми у яких:

а. були виконані всі підходи у 95% користувачів. Тобто, якщо сказано виконати 50 віджимань, і вони виконані – значить можна йти далі. Це збереже користувачів від перевантажень;

б. серед програм що залишились, обираються ті, у котрих у середньому кращі результати фінальної перевірки;

4) формується нове покоління з обраної програми;

5) алгоритм починається з 2 шагу.

Таким чином алгоритм буде завжди покращуватись і з часом може стати ефективним алгоритмом збільшення виконаних вправ для користувачів з урахуванням індивідуальних фізіологічних параметрів.

УДК 629.1.07

Матюшок М. М.,

студент факультету комп'ютерних наук

Бурлаченко І. С.,

старший викладач кафедри КІ,

ЧНУ ім. Петра Могили, м. Миколаїв, Україна

СИСТЕМИ НАВІГАЦІЇ АВТОНОМНИХ БЕЗПЛОТНИХ НАЗЕМНИХ ТРАНСПОРТНИХ ЗАСОБІВ

Дослідження ставить за мету розробку програмно-апаратного комплексу, апаратну частину якого складає міні комп'ютер з відеокамерою, а програмну – клієнт-серверне ПЗ. Розглянемо системи навігації, зокрема використання відеокамер для навігації. Більшість транспортних компаній намагаються зробити процес перевезення

максимально автономним та ефективним, використовуючи автономні безпілотні наземні транспортні засоби (АБНТЗ) та створюючи мобільні застосунки для автоматизації процесів перевезень вантажів. Таким чином в процесах задіяна мінімальна кількість людського персоналу, зменшуючи вплив людського фактору. Сучасні технології сприяють розвитку цього напрямку, створено багато різних АБНТЗ, які виконують перевезення, і відрізняються своєю будовою, розмірами та функціями, проте в них є дещо спільне, для забезпечення автономності, а саме для прокладання маршруту та уникнення зіткнень з перешкодами використовуються лідари, радары, відеокамери та GPS – модулі. Проте досягнення максимальної автономності потребує великих фінансових вкладів в розробку. У АБНТЗ використовуються LIDAR (Light Detection and Ranging) та LADAR (Laser Detection and Ranging) модулі, принцип дії яких наступний: випромінювання лазера направляється на об'єкт, відбивається від нього та повертається назад до лідару. Після цього в пристрої визначається відстань до об'єкта. Маючи кут огляду 360° можна отримати сукупність точок навколишнього середовища. Використовуючи спеціальне програмне забезпечення, можна отримати тривимірне зображення, яке відтворює навколишнє середовище навколо системи лідар з точним положенням в просторі.

Крім лідару для забезпечення навігації можливе використання радару. RADAR (Radio Detection And Ranging) випромінює радіохвилі та приймає відбиті від об'єктів радіохвилі, після чого виконується розрахунок відстані. Стисле порівняння характеристик радару та лідару наведено в таблиці 1.

Таблиця 1 – Порівняння характеристик радару та лідару

Назва характеристики	Radar	Lidar
Джерело випромінювання	Антенa	Лазер
Діапазон довжини хвилі	Від 30 см до 3 мм	$\lambda \leq 1,5$ мкм
Переваги	Не має залежності відпогодних факторів	Висока точність детектування (декілька сантиметрів на відстані 100 метрів)
Недоліки	Низька точність детектування (декілька метрів на відстані 100 метрів)	Критичність до пропускнуої можливості атмосфери

Як видно з порівняльної таблиці лідар має значно вищу точність, проте несприятливі погодні умови мають критичне значення, а радар – навпаки, проте цілком доцільно додати до порівняльної таблиці ціну

на радар та лідар. Тому що модулі лідару мають значно вищу ціну ніж модулі радару.

Лідарні комплекси можна поділити на дві категорії, перша – лідари для вимірювання відстані, топографії (скануючі), та друга категорія – лідари для контролю стану атмосфери і забруднень в атмосфері. Блок-схема роботи лідарної системи наведена на рисунку 1.

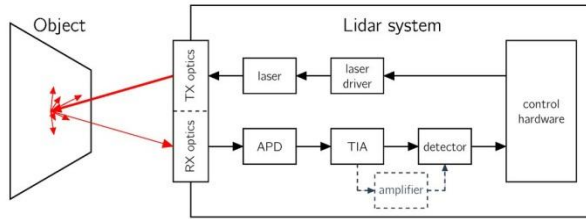


Рисунок 1 – Блок-схема роботи LIDAR системи

У свою чергу, сканувальні системи поділяються на декілька видів, а саме: сканувальна система із поворотним дзеркалом, система із поворотною призмою та фазована антенна решітка. Сканувальна система із поворотним дзеркалом має синусоїдальну розгортку та частота коливання обернено пропорційна заданому куту огляду. Основний недолік сканувальної системи із поворотним дзеркалом це непостійна швидкість руху дзеркала, в результаті чого виникає нерівномірна щільність точок на розгортці, менше точок всередині полоси сканування та більше по краям. Сканувальна система із поворотною призмою, в такій системі призма із дзеркальними гранями постійно обертається навколо своєї осі симетрії. Лазерний промінь переходить із однієї грані призми до іншої, в результаті чого отримується масив точок, який складається із паралельних рядів. Активна фазована антенна решітка – це ряд випромінювачів, які можуть змінювати напрямок лазерного променя, підлаштовуючи відносну фазу сигналу між сусідніми передавачами. В такій системі промінь має перпендикулярний напрямок відносно масиву випромінювачів. Щоб відхилити промінь вліво, передавачі здвигують фазу сигналу і таким чином сигнал від передавачів зліва знаходяться позаду сигналу передавачів, що знаходяться справа.

Найбільш поширені довжини хвиль випромінювання лазеру 850 нм, 905 нм та 1550 нм відповідно. Така довжина хвилі має як свої переваги так и недоліки. Порівняння хвиль випромінювання наведено в таблиці 2.

Таблиця 2 – Порівняння довжин хвиль випромінення лазеру

Назва характеристики	850 нм, 905 нм	1550 нм
Переваги	Недорогі приймачі випромінення; Лазери безпечні для зору	Велика відстань вимірювання – до 1000 м; Лазери також безпечні для зору
Недоліки	Невелика відстань вимірювання – до 100 м	Дорогі приймачі випромінення

$$D = 0.5 * c * t, \quad (1)$$

де c – швидкість світла; t – повний час проходження світлом шляху до точки відбиття і назад; D – відстань до точки відбиття;

У безперервно випромінюючих лазерах вимірюється зсув по фазі φ між випроміненою і хвилею що приймається (2):

$$t = \left(\frac{\varphi}{2\pi}\right) * T, \quad (2)$$

де φ – фазовий зсув; T – період світлових коливань лазерного випромінення;

Щоб виміряти відстань до точки поверхні об'єкта використовується формула (2) з новим значенням t , яке було отримане у формулі (2). Лідари в АБНТЗ можуть використовуватися для автономного планування маршрутів по місту. Принцип використання Lidar-технологій полягає в наступному: дискретне сканування певних ділянок та об'єктів; пристрій фіксує напрямок випромінювання, загальний час проходження променю до об'єкта і назад до приймача; приймач підтримує технології GPS і визначає поточне розміщення сканера; визначаються абсолютні координати відмічених точок відбиття. Стислу порівняльну характеристику описаних моделей АБНТЗ наведено в таблиці 3.

Таблиця 3 – Порівняльна характеристика описаних моделей АБНТЗ

Виробник	Технології для навігації	Швидкість руху, км/год	Максимальний вантаж, кг	Маса робота, кг
Starship	Радари, ультразвукові датчики, GPS, FoT камери.	6	9,1	25
Refraction AI	GPS, камери, гідролокатори	24	82	45
Amazon	GPS, гідролокатори	24	23	45

Очевидно, що всі наведені моделі АБНТЗ мають схожі технології для автономної навігації. У даній роботі, було досліджено модель АБНТЗ, що має шестиколісну базу, та доставляє вантажі масою до 40 кг. Вантажний відсік заблокований і може бути відкритий лише отримувачем. Пересування АБНТЗ можна відстежувати протягом всього часу перевезення, використовуючи мобільний застосунок, який доступний як для IOS так і для Android. Для досягнення автономної навігації та доставки в цілому розробники використали комбінацію GPS, зовнішніх камер та датчиків. Вони сканують вулиці, тротуари, пішохідні переходи на дорогах, використовуючи ультразвукові датчики, радары, стереокамери та ToF камери. Це дозволяє записувати дані зі швидкістю понад 2000 кадрів в секунду, а внутрішня система відображення дозволяє роботу визначити своє місцезнаходження з точністю до дюйма. АБНТЗ орієнтуються оминаючи пішоходів під час дорожнього руху за допомогою машинного навчання (МН). Збираючи великі обсяги даних і обробляючи їх кількома процесами, модуль МН АБНТЗ буде найкращий можливий маршрут. При чому кожне зіткнення покращує побудову маршруту в наступному. Оператори слідкують за роботою АБНТЗ та беруть під контроль якщо виникають несподівані ситуації під час доставки. Після того як ситуація була вирішена, АБНТЗ знову починає працювати в автономному режимі.

УДК 004.93: 621.317.785

Мельник О. Д.,
викладач,

ЧНУ імені Петра Могили, м. Миколаїв, Україна

ПЕРЕРОБКА ПЕТ-ПЛЯШОК У ФІЛАМЕНТ ТА ПОВТОРНЕ ВИКОРИСТАННЯ ВІДПРАЦЬОВАНОГО ПЛАСТИКУ ДЛЯ 3D-ПРИНТЕРІВ.

На сьогоднішній день використання 3D-принтерів дуже поширено. За допомогою них створюються як прототипи різних інструментів та винаходів, так і звичайні декоративні або предмети домашнього вжитку. Але, після створення моделей на таких принтерах, майже завжди залишаються відходи пластику від підтримки та невдалих спроб. Зазвичай їх просто викидають, але їх можна використати знову, після переробки у філамент. Повторна переробка пластику дозволить зменшити витрати на матеріал а також позитивно вплине на екологію. Адже весь пластик, який люди викидають, є досить токсичним для

природи а також довго розкладається. Також для виготовлення філаменту можна використати звичайні пластикові пляшки, які зустрічаються всюди у домашньому ужитку.

Для того, щоб переробити пластик потрібно знати всі його властивості. І якщо для невдалих моделей та інших відходів, які були надруковані на принтері, ми можемо дізнатись тип пластику на самій катушці, то для пластикових пляшок потрібно орієнтуватись на колір. Для 3D-друку розглядаються такі види пляшок, виготовлених з поліетилентерефталату:

- Блакитні. Такі пляшки мають найжорсткіший пластик, але в розплавленому вигляді він буде найбільш рідким з усіх.
- Білі та зелені. Пластик у цих пляшках м'якший, ніж блакитний. Матеріал густіший у розплавленому вигляді.
- Коричневі. Ці пляшки мають м'який пластик. У розплавленому вигляді він характеризується великою в'язкістю (подібно до ABS-філаменту).

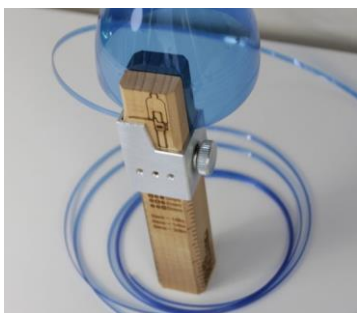


Рисунок 1 – Зовнішній вигляд пляшкорізу.

Для створення філаменту з пляшок потрібно зробити декілька кроків:

- Нарізати їх за допомогою пляшкорізу (рис.1);
- Поставити в духову шафу до повного плавлення;
- Після плавлення пластик від пляшок повинен повністю охолонути. Вся зайва волога випарується з матеріалу, і він кристалізується. На вигляд полімер нагадує скло;
- Повторно подрібнити матеріал;
- Далі подрібнений пластик засипається у вертикальний екструдер. Шнек, що обертається, переміщає мелений полімер в нижню частину пристрою до сопла з нагрівачами. Пластик починає плавитися;
- Пластик з екструдера виходитиме до ємності з холодною водою, щоб запобігти злипанню.

Перед використанням саморобного філаменту для друку слід добре просушити. Так як за наявності великої кількості вологи нитка розплавленого пластику виходитиме з екструдера у вигляді піни. Для висушування можна використовувати духовку або просто розташувати тару на батареї опалення.

Для створення філаменту з відходів після друку потрібно зробити менше кроків, а саме виключити перші кроки і відразу ж почати з подрібнення пластику та засипанням в екструдер.

При використанні готової нитки з пластикових пляшок для 3D-друку на принтері слід врахувати кілька нюансів:

- Друкувати виріб потрібно на холодному столі, температура якого не перевищує 35 °С. Таке обмеження температурного режиму пов'язане з тим, що пластик повинен швидко охолонути до температури нижче 70 °С. В іншому випадку матеріал може практично повністю втратити свої властивості міцності;

- Температура екструдера має бути близько 265 °С. Але вона може відрізнятись залежно від швидкості друку та типу пластику, з якого виготовлений філамент;

- Якщо механізм 3D-принтера, що подає, складається з латуні, то на ролик краще надіти тонкостінний шланг з гуми. Він не дасть нитці з такого матеріалу ковзати.

Звісно при повторному використанні відпрацьованого пластику для принтерів нюансів немає, адже це просто переплавлений пластик в зручну для принтера форму. Але пластик з пляшок має свої переваги та недоліки при використанні.

Основною перевагою переробки пляшок на пластикові нитки для 3D-друку є те, що можна заощадити на покупці промислових філаментів та переробити непотрібну тару, яка при вивезенні на звалища сильно забруднює довкілля.

Серед недоліків використання саморобного філаменту слід виділити таке:

- можна друкувати вироби тільки на низькій швидкості, тому що при її підвищенні нитка рветься;

- потрібний дробарка для помелу матеріалу та окремий екструдер для переправлення пластику;

- неможливо надрукувати великі за розміром вироби, оскільки довжина нитки обмежена.

Самостійне виробництво ниток для 3D-друку – це чудовий спосіб заощадити на витратних матеріалах та переробити непотрібні пляшки. Саморобні філаменти близькі за властивостями до покупних. Тому за їх допомогою можна друкувати невеликі вироби відмінної якості.

Пузирьов С. В.,
канд. фіз.-мат. наук, доцент,
Кім В. М.,
студент факультету комп'ютерних наук,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

СИСТЕМА ОПОВІЩЕННЯ НА БАЗІ ESP-32 CAM, ДАТЧИКУ РУХУ ТА TELEGRAM BOT API

В інформаційних технологіях система оповіщення – це комбінація програмного та апаратного забезпечення, що забезпечує засіб доставки повідомлення набору одержувачів. Зазвичай він показує діяльність, пов'язану з обліковим записом. Такі системи є важливим аспектом сучасних веб-додатків.

Важливою складовою Інтернету речей (Internet of things, IoT) є засоби та інструменти інтеграції IoT-пристроїв з мобільними пристроями, зокрема із месенджерами. Сучасні функціональні можливості дозволяють використовувати месенджер як складову IoT. Завдяки цьому користувачі можуть оперативно отримувати дані з підключених до месенджеру пристроїв, при чому місце та час не мають значення, за умови доступу до Інтернету.

В даній роботі представлено концепт безпроводної камери з датчиком руху, інтегрованої із заданим месенджером через його API (прикладний програмний інтерфейс). За допомогою такого пристрою (або мережі пристроїв) можна:

- вести відеоспостереження за певною територією у полі зору камери;
- оперативно передавати відеодані або статичні зображення користувачу;
- виконувати попередню обробку зображення або відеопотоку;
- надсилати повідомлення користувачу про спрацювання певних датчиків;

В якості апаратного модуля використовується Wi-Fi-камера ESP32-CAM, яка може передавати дані до Telegram-месенджеру за допомогою спеціально створеного бота, а також піроелектричний інфрачервоний (PIR) датчик руху, який дозволяє вловлювати рух. Цей датчик малий за габаритами, недорогий, споживає мало енергії, легкий в експлуатації, практично не схильний до зносу.

Відеодані відправляються до бота за допомогою протокола MQTT, який побудований поверх TCP/IP та орієнтований для обміну інформацією між пристроями за принципом «Producer-Consumer».

Перевагами протоколу MQTT є:

1. Простота використання. Це програмний блок без зайвої функціональності, що може бути легко вбудований в будь-яку складну систему.

2. Зручність для більшості рішень з датчиками. Дає можливість пристроям виходити на зв'язок і публікувати повідомлення, які не були заздалегідь відомі або визначені.

3. Легкість адміністрування.

4. Низьке навантаження на канал зв'язку.

5. Надійність роботи в умовах ненадійного зв'язку.

6. Гнучкі формати передачі контенту.

Модуль ESP32-CAM – це модуль, який поєднує в собі плату для розробки ESP32 та камеру OV2640 2MP. Плата відрізняється скромним споживанням електроенергії, а також компактними розмірами.

Завдяки тому, що в плату вбудована камера, такий пристрій можна використовувати в якості камери спостереження, систем розпізнавання осіб й жестів, сканування QR кодів та багатьох інших цілей. Налаштування плати ESP також потрібні для тестування й вдосконалення різних програм. За допомогою такого пристрою Ви можете знайти помилки в роботі, спробувати різні режими роботи пристрою, спостерігати за реакціями та налаштувати робочий процес.

ESP32-CAM-OV-2604 не має USB-роз'єму, тому знадобиться програматор FTDI для завантаження коду через контакти U0R і U0T (послідовну шину). Також на платі присутній модуль Wi-Fi BT SoC 802.11 b/g/n та 32-бітний процесор з низьким енергоспоживанням

Макетна схема камери з датчиком руху та програматором представлена на рис. 1

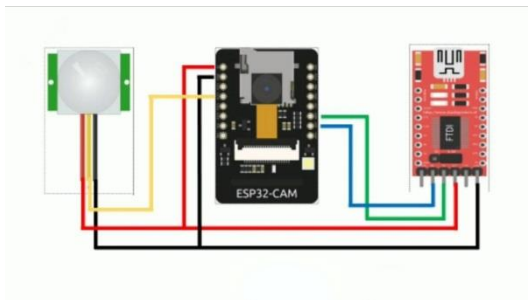


Рисунок 1 – Схема підключення до датчику руху та програматора

Структурна схема взаємодії камери та месенджеру представлена на рис. 2

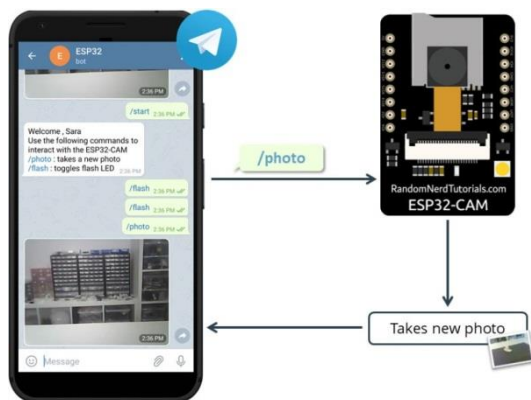


Рисунок 2 – Структурна схема взаємодії з Telegram-месенджером

УДК 004.45 (076.5)

Савінов В. Ю.,
канд. техн. наук, доцент кафедри КІ,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

ЗАХИСТ ЛИСТУВАННЯ ЗА ТЕХНОЛОГІЄЮ PGP

PGP (Pretty Good Privacy) – комп'ютерна програма, а також бібліотека функцій, що дозволяє виконувати операції шифрування та цифрового підпису повідомлень, файлів та іншої інформації, представлена в електронному вигляді, у тому числі прозоре шифрування даних на пристрої, наприклад, на жорсткому диску.

PGP має безліч реалізацій, сумісних між собою та іншими програмами (GnuPG, FileCrypt та ін.) завдяки стандарту OpenPGP (RFC 4880), які у той же час мають різний набір функціональних можливостей. Існують реалізації PGP для всіх найпоширеніших операційних систем. Окрім реалізацій, що вільно поширюються, є ще й комерційні.

Механізм роботи PGP полягає у тому, що шифрування PGP здійснюється послідовно хешуванням, стиском даних, шифруванням з симетричним ключем, і, нарешті, шифруванням з відкритим ключем.

Причому, кожен етап може здійснюватися одним з декількох відповідних алгоритмів.

Для реалізації поштового сеансу спочатку необхідно встановити GPG (GNU Privacy Guard) – безкоштовний інструменту, що реалізує технологію PGP. Після цього встановлюється поштовий клієнт Thunderbird та налаштовується обліковий запис.

Додавання плагіну EnigMail через пункт меню Add-ons у Thunderbird дозволяє зашифрувати вміст листів та ставити цифровий підпис (рис. 1).

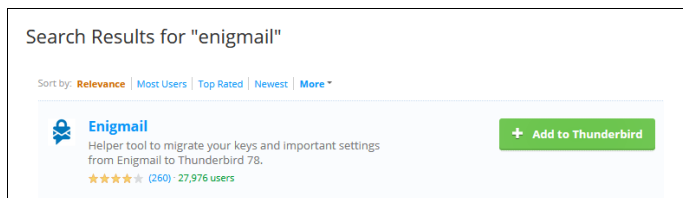


Рисунок 1 – Плагін *Enigmail*

Після цього користувачем PGP здійснюється генерація публічного та приватного ключів (рис. 2).

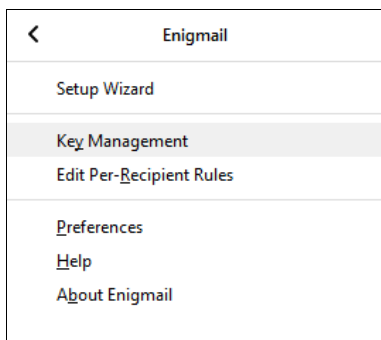


Рисунок 2 – Перехід до керування ключами

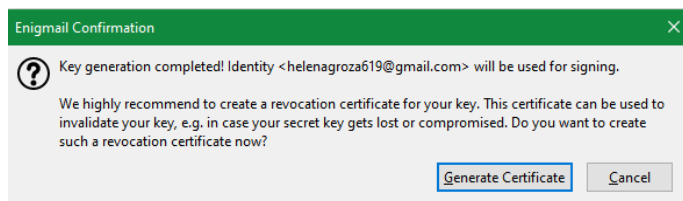


Рисунок 3 – Генерація сертифікату для скасування ключів

При генерації ключів задаються їхній власник (ім'я та адреса електронної пошти), тип ключа, довжина ключа та термін його дії. Відкритий (публічний або *Public key*) ключ використовується для шифрування та перевірки цифрового підпису. Відкритий ключ можна надавати кому завгодно, тому що він дає можливість отримувачу прочитати зашифрований лист, а також використовується для декодування та створення цифрового підпису. Закритий (приватний або Private key) ключ не можна показувати нікому. Закритий ключ дозволяє зашифрувати листи, що надсилаються. Крім того, для захисту приватного ключа використовується пароль – Passphrase. Введення паролю виконується перед написанням листа, щоб можна було скористатися функціями шифрування та цифрового підпису.

При генерації ключів варто також згенерувати сертифікат (рис. 3), за допомогою якого можна зробити ключ недійсним в разі, якщо він буде загублений або розсекречений. Сертифікат становить собою файл, який треба зберігати у надійному місці.

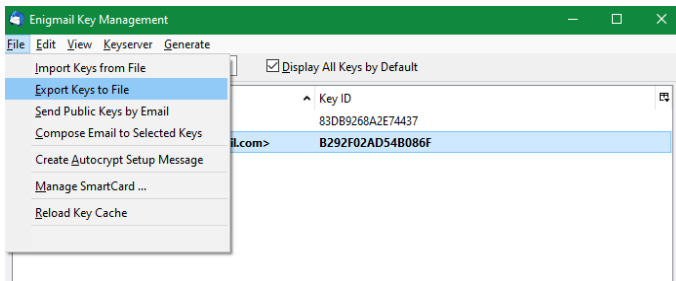


Рисунок 4 – Експорт ключа (лише публічного!) для переказу другій особі

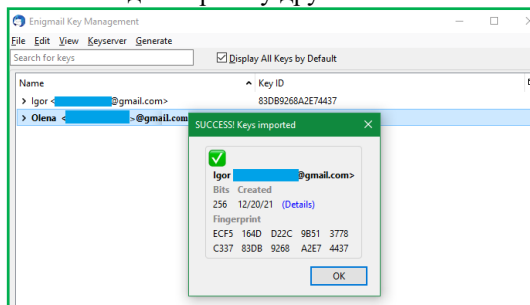


Рисунок 5 – Імпорт *Public Key*, отриманого від другого кореспондента

Щоб мати можливість надсилати та отримувати листи, необхідно обмінятися з другим кореспондентом публічними ключами. Для цього треба здійснити їх експорт, наприклад, до файлу (рис. 4) та передати другій особі для імпорту (рис. 5). Інший спосіб – відправити публічний ключ у листі.

Альтернативний варіант – завантажити свій публічний ключ на сервер ключів (**Key Server**), щоб він автоматично був доступний одержувачу листа (рис. 6).

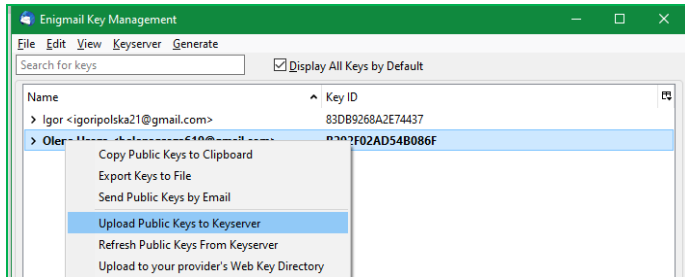


Рисунок 6 – Завантаження публічного ключа на **Key Server**

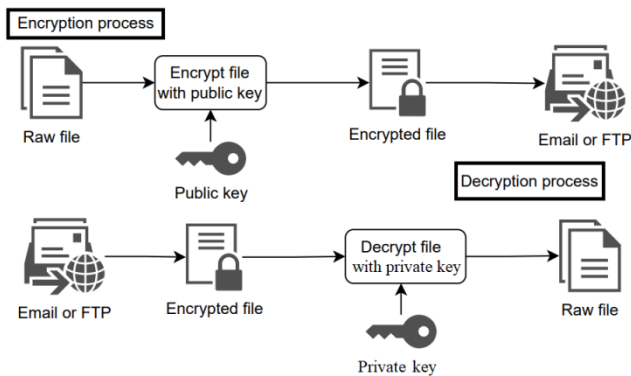


Рисунок 7 – Механізм листування з використанням PGP

Таким чином, технологію PGP можна ефективно використовувати для захисту електронних листів та додання до них електронного підпису. Інструменти, які реалізують дану технологію, застосовують асинхронне шифрування за допомогою публічного та приватного ключів. В результаті успішно реалізується поштовий сеанс з іншою людиною-адресатом (рис. 7).

Як при шифруванні повідомлень, так і під час перевірки цифрового підпису необхідно, щоб прийнятий одержувачем відкритий ключ дійсно належав відправнику. При простому завантаженні відкритого ключа він може бути замінений. Однак недостатньо просто створити сертифікат, захищений від модифікації, оскільки таким чином гарантується лише цілісність сертифіката після його створення. Користувачі також повинні у будь-який спосіб перевірити, що відкритий ключ у сертифікаті дійсно належить відправнику.

УДК 004.8

Степанов К. М.,
студент факультету комп'ютерних наук,
Крайник Я. М.,
канд. техн. наук, доцент, завідувач кафедри КІ,
ЧНУ ім. Петра Могили, Миколаїв, Україна

СИСТЕМА МОДЕЛЮВАННЯ РОБОТИ ДЕКОДУВАННЯ LDPC-КОДІВ З ВИКОРИСТАННЯМ БІБЛІОТЕК JAVASCRIPT

З метою досягнення поставленої задачі, а саме розробки системи моделювання роботи декодування LDPC-кодів з використанням бібліотек JavaScript було проведено детальний аналіз предметної області і визначення поняття «сервіс» як динамічна система дій, спрямованих на отримання заданих результатів у багатокритеріальному полі протягом встановленого строку та в рамках виділених ресурсів із залученням виконавців, які володіють необхідними навичками та знаннями.

На основі проведеного аналізу було здійснено вибір технічних та програмних засобів для реалізації системи. Систему було вирішено спроектувати на базі «JavaScript» фреймворку «React.js».

У рамках проведеного опису концепції порівняння та дослідження базових принципів та відмінностей технологій декодування, було обрано для забезпечення відповідного програмного середовища, а також проаналізовано основні підходи до налаштування процесів постійної інтеграції. Також було наведено основні питання які стосуються роботи алгоритмів та методів декодування в системі, а також підходів до масштабування додатку.

Аналіз архітектури платформи «React.js» та дослідження процесів роботи з бібліотеками також мали місце в процесі виконання

дослідження та надали можливість сформувати програмні вимоги до системи, серед яких є використання наступних технічних і програмних засобів: «Npm», «React.js», «Webpack», «VirtualBox».

При розробці програмної частини системи було досліджено великий спектр бібліотек, та створено сайт для моделювання роботи декодування LDPC-кодів, сам процес розробки програмного забезпечення детально описано.

УДК 681.518.3

Солобута Л. В.,

канд. техн. наук, доцент кафедри КІ,

Полівода Д. В.,

студент факультету комп'ютерних наук,
ЧНУ ім. Петра Могили, Миколаїв, Україна.

СИСТЕМА ЗБОРУ ІНФОРМАЦІЇ ДЛЯ ВОДІЯ

В сучасному світі більшість повсякденних завдань спрощується або зовсім автоматизується і кожного дня ця тенденція зростає. В зв'язку з тим, що кількість автомобілів зростає з кожним роком, все складніше стає переміщатися у великому місті і дуже непростим завданням стає паркування автомобіля. Саме під час паркування часто відбуваються пошкодження автомобілів з-за великого скупчення автівок та обмеженого простору в місцях паркування. Тому пристрої, що полегшують маневрування автомобіля під час паркування є актуальними.

Найбільш розповсюдженою є система парктроніку. На бортах автомобіля (зазвичай на передньому та задньому бампері, іноді і з боків) кріпляться датчики, вони сигналізують про відстань до перешкоди, інформація про це видається на дисплеї або у вигляді звукового сигналу (рисунок 1).

Принцип роботи паркувальних систем.

Алгоритм роботи паркувальної системи досить простий – коли водій починає рухатися заднім ходом, паркувальний радар включається автоматично. У корпус бампера встановлюються ультразвукові датчики.

Визначальними при виборі є кількість датчиків. Для досвідченого водія буде достатньо 4 датчиків. Парктроніки з 6 датчиками добре визначає кут переднього бампера: 2 попередньо

встановлювати в передній бампер і 4 в задній. Паркувальні системи на 8 датчиків для тих, хто недавно за кермом або відчуває себе невпевнено.

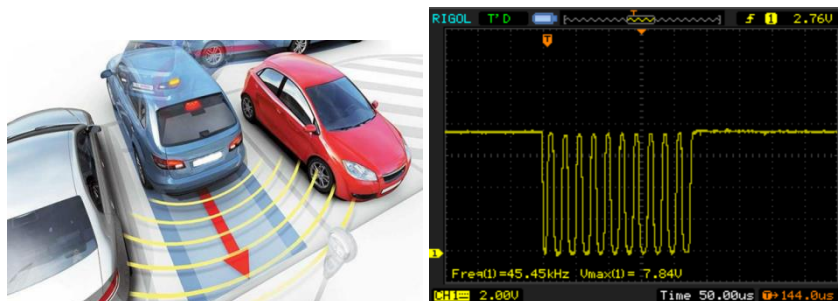


Рисунок 1 – Робота системи парктроніку

Парктроніки також бувають з дисплеями і без. При виборі дисплея, потрібно керуватися по-перше, тим в який час доби використовується автомобіль і по-друге, ціною.

Крім того, в паркувальних системах може бути встановлено багато інших функцій – голосових попереджень, діагностика неполадок, функція запам'ятовування парктроніком виступаючих частин автомобіля (фаркоп, запасне колесо), це дозволить уникнути помилкових спрацьовувань пристрою.

Тобто, будь-яка паркувальна система складається з трьох складових:

- ультразвукові датчики;
- блок управління;
- системи виведення даних (це може бути екран або маленькі динаміки, вбудовані в блок управління).

Кількість датчиків може змінюватись в залежності від моделі від двох до восьми та більше. Деякі моделі із двома датчиками дозволяють підключити додаткові.

Блок генерує пакет імпульсів, частота яких знаходиться в ультразвуковому діапазоні.

Оскільки існує величезна кількість людей, яка користується старими та бюджетними машинами, де відсутній бортовий комп'ютер, а також враховуючи високу ціну на стандартні системи парктроніку, є необхідність розробити систему з ультразвуковими датчиками та смартфоном.

Система має містити: УЗ датчики, блок керування на Ардуино та смартфон для виводу даних.

Діючий макет представлено на рисунку 2.

Модель має два датчики і «мертва зона» між ними може бути значною, тому діюча система має містити мінімум 4 датчики.

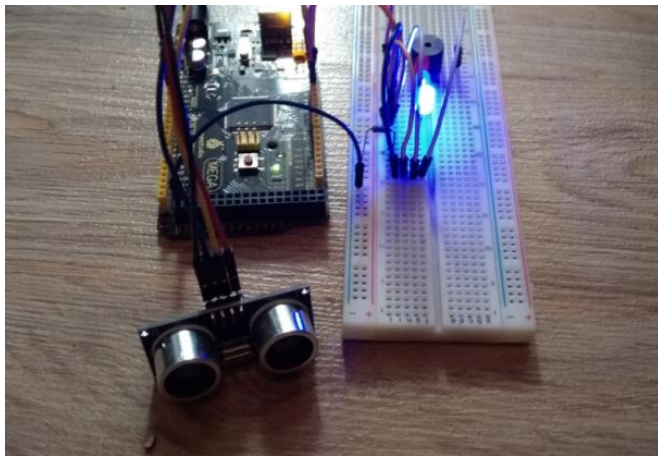


Рисунок 2 – Макет парктроніку

Висновок. Система паркування з використанням смартфона є сучасним та дешевим рішенням проблеми для користувачів машин будь-якого класу.

УДК 004

Сизоненко О. М.,
д-р техн. наук, професор,
Христо О. І.,
канд. техн. наук., старший науковий співробітник,
ІІІТ НАНУ, м. Миколаїв, Україна

СИСТЕМА МОНІТОРИНГУ ПРОЦЕСУ СПІКАННЯ ДИСПЕРСНИХ МАТЕРІАЛІВ ЕЛЕКТРИЧНИМ СТРУМОМ

У сучасній літературі зазначено, що спікання це процес консолідації порошкових матеріалів, який відбувається нижче точки їхнього плавлення за умови високої сили їх стиснення, спрямований на

зміну структури матеріалу. Існують багато різних методів спікання, з яких можна виділити метод іскро - плазмового спікання (ІПС), що відбувається одночасно за рахунок пропускання через зразок імпульсного модульованого струму та під дією зовнішньої одноосової сили при низькому атмосферному тиску з метою високошвидкісного ущільнення порошкового матеріалу. Лабораторні установки, які забезпечують відзначений процес спікання, мають задовольняти таким його параметрам, як швидкості нагріву та охолодження, сили навантаження, швидкості стиснення, глибини вакууму та потужності введення енергії до матеріалу. Важливими елементами такої установки є вакуумна камера з системою створення механічного навантаження, потужний генератор випрямленого пульсуючого струму, системи водоохолодження та вакуумування камери, а також системи керування та моніторингу процесу спікання. Керування процесом спікання може відбуватися, як вручну так й в автоматичному режимі по заданому алгоритму зміни його визначальних характеристик. Система моніторингу у складі ІПС установки відіграє важливу роль у тому сенсі, що повинна забезпечити якісну обробку інформаційних сигналів з метою їх збереження та подальшому впровадженні у математичних розрахунках.

Основне завдання вимірювального комплексу полягає в обробці аналогових сигналів, які надходять з датчиків вмонтованих у технологічних вузлах ІПС установки (генератор струму, вакуумна камера, механічний прес) для визначення наступних характеристик процесу: температура спікання зразка, тиск механічного навантаження, середнє значення імпульсного струму через зразок, швидкість стиснення, тиск залишкового вакууму у камері, середній рівень напруги на зразку й виділену потужність. Передача отриманих даних до персонального комп'ютера з можливістю зміни кроку дискретизації процесу теж є одним із вагомих завдань системи, структурна схема якої відображена на рис.1.

Можна зазначити, що кожен вузол відображений на структурній схемі приєднаний до зовнішніх портів мікроконтролера (МК) представляє собою відокремлений модуль розширення на якому окрім датчика можуть розміщуватися такі елементи, як диференційний посилювач сигналу, інтегруюча або диференційна ланка, перетворювач рівнів сигналу, елемент гальванічної розв'язки з лінійним коефіцієнтом передачі, джерело живлення та інші обов'язувальні елементи. Крім того, система повинна забезпечити стійкість до електромагнітних перешкод, що надходять у першу чергу з напівпровідникового генератора

імпульсного струму та з інших пристроїв автоматизованого керування процесом ППС.

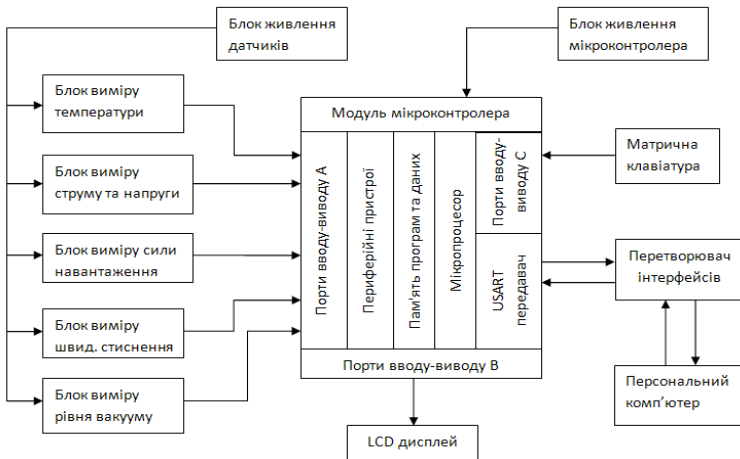


Рисунок 1 – Структурна схема системи моніторингу ППС

Технологія спікання порошкового матеріалу, який розміщено у графітовій матриці стиснутого з обох боків пуансонами під дією, як струму високої щільності, так й зовнішнього механізму навантаження, відбувається за температурними показниками до 1500 °С. Вимірювання температури нагрітого зразка прийнято проводити хромель-алюмельовою термопарою. Сигнал із термопари на рівні від 5 мВ до 50 мВ подається на вхід інструментального диференціального підсилювача AD627 та підтягується до рівня, достатнього для оцифрування, й направляється на один із входів вбудованого модуля АЦП МК Atmega2560. Крім того, для усунення попадання високої напруги на порти МК в схемі передбачено гальванічну розв'язку силової частини від вимірювального тракту на основі лінійної оптопари IL300. Таку обережність пов'язано з можливістю виникнення величезних значень електрорушійної сили (ЕРУС) самоіндукції на зразку, що спікається, в результаті короткочасного обриву струму, що проходить через нього величиною порядку 1 кА і більше. Коефіцієнт посилення підсилювача AD627 визначається за формулою та задається резистором на його вході:

$$K_{\text{п}} = 5 + 200k\Omega/R_g, \quad (1)$$

де R_g – опір резистора на вході підсилювача AD627, Ом.

Значення цього коефіцієнту задається з урахуванням напруги насичення обраної термопари, а результуючий сигнал не повинен перевищувати опорної напруги АЦП МК. Максимальне значення хромель-алюмельової термопари при температурі 1350 °С сягає 50 мВ. Якщо коефіцієнт посилення дорівнює $K_n=45$, амплітуда сигналу буде сягати 2,43 В, що є прийнятною напругою на вході АЦП. Лінійну оптопару ІЛ300 може бути налаштовано, як для фотодіодного, так і для фотогальванічного режиму підсилювання. Як відомо, фотогальванічна топологія забезпечує кращу лінійність, найнижчий рівень шуму та дрейфові характеристики. Якщо лінійний оптрон побудувати з ОП LM358, то лінійність передачі можна отримати на частоті сигналу приблизно до 2 кГц. Так як процес нагріву або охолодження зразку в матриці відбувається достатньо повільно, то отримана частотна смуга пропускання оптрона вважається прийнятною.

Для виміру струму у навантаженні використано датчик струму компенсаційного типу на основі ефекту Холла. Модель CSNR161 дозволяє вимірювати змінний, постійний або імпульсний струми амплітудою ± 200 А з частотою до 150 кГц. Генератор імпульсного струму працює на частоті до 50 кГц, але амплітуда імпульсів може сягати 1200 А. Струмознімальний контакт генератора виконано у вигляді широкої мідної шини розміром 70 мм \times 50 мм. Переріз вікна обраного датчика струму складає всього 10 мм \times 13,5 мм, тому його вбудовано у шину таким чином, щоб охоплювати тільки деяку частину її поверхневої площі. Якщо зробити припущення, що вимірювальний струм розподіляється рівномірно по перерізу шини, то частину струму, що проходить через вікно датчика, можна визначити за відношенням їх перетинів $S_d/S_{ш} = 1/7$. Живлення датчика здійснюється від біполярного джерела ± 12 В. Вихід датчика в своєму роді є джерелом струму, який навантажено на опір порядку від 30 Ом до 40 Ом. Внаслідок цього сигнал напруги на резисторі виявляється пропорційним струму в первинному колі, поділеному на кількість витків додаткової обмотки. Значення струму можна обчислити за формулою:

$$I = V_H \cdot N / R_L, \quad (2)$$

де V_H – напруга на опорі навантаження, В; N – кількість витків компенсаційної обмотки, од.; R_L – навантажувальний опір, Ом. Для обраної моделі датчика $N=1000$, $R_L=30$ Ом. Результуюче значення струму розраховується з урахуванням коефіцієнту відношення перетинів.

Оскільки сигнал струму у своєму спектрі може містити високочастотну складову, то перед подачею на вхід АЦП його

потрібно попередньо проінтегрувати (згладити) для отримання середнього значення. Треба також зазначити, про необхідність введення випрямляючої ланки сигналу датчика струму на випадок, якщо спікання буде відбуватися від генератора змінного струму. Випрямляч зібраний на двох ОП, тому потребує біполярного джерела для їх живлення. Так як конструкція датчика забезпечує гальванічну розв'язку з силовим контуром, то усереднений сигнал струму без будь-яких додаткових ланцюгів оптичної розв'язки подається безпосередньо на один із входів АЦП МК для подальшої обробки. Вимірювання напруги на навантаженні теж відбувається за подібним принципом як і для струму, але з тою відмінністю, що перед подачею на АЦП МК усереднений сигнал напруги подається на лінійний оптрон PL300 для забезпечення гальванічного розв'язування з силовим контуром. Сигнал напруги з вузлів навантаження подається на вхід блоку напруги через коаксіальну скрановану лінію. Під'єднання лінії з обох її кінців як до навантаження, так і до блоку здійснюється через байонетні роз'єми. Зовнішній вигляд розглянутих блоків представлено на рис. 2.



Рисунок 2 – Блоки вимірювання напруги, струму й температури

Силу навантаження порошкової суміші у графітовій матриці та швидкість її стиснення вирішено аналізувати на основі кута повороту відповідного обертового механізму машини Р-0,5. У ролі кутового вимірювача використовується інкрементальний магнітний енкодер на основі ефекту Холла AS5040, що має роздільну здатність з дозволом $0,35^\circ$ або 1024 імпульсів за оборот. Дія датчика запускається від

діаметрально намагніченого двополюсного магніту, який повинен розміщуватися паралельно торцевій поверхні мікросхеми на висоті від 0,5 мм до 1,8 мм. Матеріал магніту – неодим-залізо-бор у формі таблетки діаметром 6 мм і висотою 2,5 мм. Зусилля, що передається порошковій суміші під дією реверсивного механізму, прямопропорційне куту повороту шестерні силовимірювача, тому для його визначення потрібне знання про абсолютне кутове положення обертового механізму, яке надається за ШІМ сигналом магнітного енкодера. Робочий цикл ШІМ прямопропорційний абсолютному кутовому положенню, яке визначається за формулою:

$$\Phi_{\text{pos}} = \frac{t_{\text{on}} \cdot 1025}{t_{\text{on}} + t_{\text{off}}} - 1, \quad (3)$$

де t_{on} – тривалість імпульсу, c ; t_{off} – тривалість паузи, c .

Аналоговий сигнал зручніше вимірювати, ніж тривалість імпульсу, тому перед подачею на МК сигнал ШІМ усереднюється за допомогою RC фільтра нижніх частот. Параметри фільтра задаються такими, щоб не перевантажувати вихід ШІМ та забезпечувати меншу пульсацію. Для таких значень фільтра, як $R_{\phi}=10$ кОм, $C_{\phi}=47$ мкФ постійна часу ланцюга буде становити біля 0,47 с. Збільшення постійної часу RC ланки зменшує пульсації, але водночас зменшує час відповіді системи, який буде помітно відчутним при високих значеннях швидкостях зростання сили стиснення. На відміну від сили навантаження для визначення швидкості стиснення знання абсолютного кутового положення обертового механізму не є обов'язковою умовою, тому для цього більш доцільно використовувати інкрементальний вихід енкодера.

Усі перетворення над двобайтовими числами з урахуванням розмірності (одиниці виміру) кожного параметра виконуються програмою комп'ютера. Цифрові дані таких параметрів, як напруга, струм, температура та тиск спочатку приводяться до значення сигналу напруги відповідно для кожного вимірювального блоку, а потім розраховуються остаточні значення цих параметрів з урахуванням апроксимаційних коефіцієнтів. З урахуванням 10-бітної роздільної здатності АЦП та опорної напруги його живлення формула перетворення зазначених параметрів до відповідного значення електричних сигналів має вигляд:

$$U, I, T, P = \frac{5 \cdot (X_{U,I,T,P} + 1)}{1024} \quad (4)$$

Струм розраховується за формулою (2), але з урахуванням коефіцієнту підсилення вихідного каскаду $K_{\text{п}}$ блока вимірювального блоку та коефіцієнту відношення перетинів $k_{\text{пер}}$:

$$I = k_{\text{п}} \cdot k_{\text{пер}} \cdot U_{\text{ДТ}} \cdot N / R_{\text{н}} \quad (5)$$

Поклавши що $K_n=2/3$, $k_{пер}=7$ та з урахуванням попередніх коефіцієнтів, остаточна формула для розрахунку струму приймає вигляд:

$$I_p = \frac{700}{6} \cdot I_d, \quad (6)$$

де I_d – сила струму дискрету на виході з магнітного енкодера, A .

Для обчислення сили навантаження спочатку потрібно визначити дискрет сигналу напруги з виходу магнітного енкодера, що припадає на $1 \text{ кс} \cdot \text{с}$. Практично встановлено, що при обертанні шестерні шкали циферблата зусиль на кут, що відповідає значенню $500 \text{ кг} \cdot \text{с}$, вихідна напруга з усередненого ШІМ виходу магнітного енкодера становить $4,54 \text{ В}$. Тому на $1 \text{ кг} \cdot \text{с}$ припадає приблизно 9 мВ . З урахуванням аналогового кроку дискретизації АЦП $5/1024 \approx 4,8 \text{ мВ}$, поправочний коефіцієнт перетворення цифрових значень сили навантаження до параметричних (кілограм-сили) дорівнює $9/4,88 \approx 1,86$.

Швидкість стиснення визначається на основі підрахунку кількості імпульсів інкрементального виходу магнітного енкодера у інтервал часу 10 с за формулою:

$$\vartheta = 6 \cdot \frac{n \cdot \Delta\varphi^0 \cdot h}{360^0}, \quad (7)$$

де n – кількість імпульсів, од; $\Delta\varphi^0$ – роздільна здатність енкодера; h – крок різьби ходового гвинта, мм .

Прийнявши, що $\Delta\varphi^0 = 360^0/1024 \approx 0,35^0$ та $h = 6 \text{ мм}$, отримуємо остаточну формулу розрахунку швидкості, виражену у міліметрах за:

$$v = \frac{36 \cdot n}{1024}, \quad (8)$$

Вихідний сигнал блоку перетворення температури може бути представлено виразом:

$$U_{Tp} = U_T \cdot k_{п1} \cdot k_{п2}, \quad (9)$$

де U_T – напруга термопар, B ; $k_{п1}$ – коефіцієнт підсилення диференційного підсилювача термопар, безрозм.; $k_{п2}$ – коефіцієнт підсилення лінійної оптопар, безрозм.

Для хромель-алюмельової термопар побудовано графік оберненої функціональної залежності температури від термо-ЕРУС та за допомогою вбудованого математичного апарату отримано її апроксимаційне наближення на основі поліному третього порядку точності. З урахуванням коефіцієнтів посилення сигналу термопар $k_{п1}=45$ та $k_{п2}=2$ апроксимаційна крива температури визначається за формулою:

$$T = 17.917 \cdot U_T^3 - 50.267 \cdot U_T^2 + 570.95 \cdot U_T - 2.014 \quad (10)$$

Виходячи з потреби знаходження більш раціонального визначення температури з урахуванням можливостей 8-ми розрядного МК вирішено апроксимаційну криву температур (форм. 10) розділити на

декілька ділянок в межах яких закон зміни температури залишається достатньо лінійним. У програмі вибір тої чи іншої ділянки організується операторами розгалуження на підставі значення регістру даних перетворення АЦП.

Висновки. Проаналізовано принципи роботи вимірювальних блоків системи моніторингу процесу ІПС та розроблена її структурна схема з врахуванням інтерфейсу їх під'єднання до обчислювального пристрою та одночасним забезпеченням гальванічної розв'язки логічних елементів системи, за умови підтримання лінійності передачі інформаційних сигналів, від технологічних вузлів силової установки де можлива поява струмів високого потенціалу. Отримано аналітичні вирази параметрів процесу спікання за якими програма мікроконтролера здійснює перетворення вхідних сигналів до їх числового зображення відповідного формату з врахуванням системи команд його вбудованого арифметично-логічного пристрою для досягнення найбільшої ефективності у витраті часу та точності розрахунку.

УДК 004.8

Шевченко О. В.,
студент факультету комп'ютерних наук,
Крайник Я. М.,
канд. техн. наук, доцент, завідувач кафедри КІ,
ЧНУ ім. Петра Могили, Миколаїв, Україна

СИСТЕМА ДОСЛІДЖЕННЯ ТА МОДЕЛЮВАННЯ РОБОТИ БІБЛІОТЕК СТИСНЕННЯ

З метою досягнення поставленої задачі, а саме розробки системи рекомендації бібліотек стиснення було проведено детальний аналіз предметної області та відібрано найпопулярніші бібліотеки, які використовуються у різних прикладних застосуваннях.

Проведено дослідження бібліотек та визначення їх швидкості кодування та декодування. Проведений аналіз дає можливість зробити висновок про те, що більшість систем поділяються на два типи, а саме стиснення з втратами та без.

На основі проведеного аналізу було здійснено вибір технічних та програмних засобів для реалізації системи. Систему було вирішено спроектувати на базі JavaScript-фреймворку «Node.js».

У рамках проведеного опису концепції порівняння та дослідження базових принципів та відмінностей технологій стиснення, було обрано для забезпечення відповідного програмного середовища, а також проаналізовано основні підходи до налаштування процесів постійної інтеграції. Також було наведено основні питання, що стосуються роботи алгоритмів та методів стиснення в системі, а також підходів до масштабування застосунку.

Аналіз архітектури платформи «Node.js» та дослідження процесів роботи з бібліотеками також мали місце під час виконання дослідження та надали можливості сформулювати програмні вимоги до системи, серед яких – використання таких технічних і програмних засобів, як: «Npm», «Node.js», «Webpack», «Express.js», «MySQL», «VirtualBox».

При розробці програмної частини системи було досліджено великий спектр бібліотек стиснення та створено сайт для підбору необхідної бібліотеки під конкретний випадок, детально описано сам процес розробки програмного забезпечення.

Секція ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ У РОЗВИТКУ СУСПІЛЬСТВА

Підсекція: ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ СИСТЕМИ

УДК 004.942

Асєєв В. Д.,
аспірант кафедри ІІС,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

СОЦІАЛЬНІ МЕРЕЖІ ЧЕРЕЗ МЕТРИКИ ТА ОГЛЯД ПЗ ВІЗУАЛІЗАЦІЇ ГРАФІВ

Аналіз соціальних мереж (АСМ або «social network analysis / SNA» в англомовній літературі) є набір методів, що дозволяють отримувати цінну інформацію про особливості соціальних мереж за їх характеристиками. В аспекті АСМ соціальні зв'язки, такі як «друзі», «колеги», «родичі», є ребрами, що зв'язують окремих людей, що служать вузлами в мережі.

АСМ має два основні завдання:

- 1) визначення кількісних та якісних метрик, що дозволяють провести всеосяжний аналіз соціальної мережі;
- 2) візуалізація її у зручному для дослідника вигляді.

Існує три типи значущих метрик соціальної мережі.

Перший включає **п'ять видів** «характеристик зв'язку»: *гомогенність* – схильність учасників організовувати зв'язки й з іншими учасниками, мають схожі властивості, *множинність* – кількість різних видів зв'язків між учасниками, *взаємність/двосторонність* – ступінь, у якому двоє учасників відповідають одне одному взаємністю у соціальних зв'язках, *закритість* – ймовірність учасників мати прямий зв'язок один з одним усередині мережі, *сусідство* – схильність учасників утворювати зв'язки Польщі з географічно близькими собі.

Другий тип складається з **шести** «характеристик розподілу»: *міст – індивід*, що служить єдиним зв'язком між окремими кластерами мережі, *центральність* – набір параметрів, що описують «рівень впливу» конкретного учасника, *щільність* – відношення між реальним

числом зв'язків та максимально можливим при даній кількості вершин, *відстань* – мінімальна кількість зв'язків між двома учасниками мережі, *структурні прогалини* – відсутність шляху між двома кластерами мережі, *сила зв'язку* – кількісна характеристика зв'язку.

Третій набір – це **три** «характеристики сегментації»: *кількість клік* (соціальних груп), *коефіцієнт кластеризації* – ймовірність, що два вузли, пов'язані з третім, будуть пов'язані один з одним, *згуртованість* – ступінь, у якому вузли утворюють прямі зв'язки один з одним.

Перелік характеристик, які можуть бути розраховані для конкретної мережі, повністю залежить від кількості відомої мережі інформації. Другий компонент аналізу соціальних мереж – візуалізація графів – буде докладно розглянутий у таблиці. На сьогоднішній день існує кілька програмних продуктів для аналізу соціальних мереж.

Порівняльний аналіз проводиться за такими критеріями: функціональність програми або бібліотеки, кількість підтримуваних вхідних та вихідних форматів, підтримувані платформи та умови розповсюдження.

У результаті порівняльного аналізу було виявлено такі закономірності:

- переважна більшість програм призначена для виконання на персональному комп'ютері;
- вони вимагають заздалегідь підготовлені дані для дослідження (за винятком Wolfram Alpha, який використовує Facebook API);
- більшість систем уніфікована та не має спеціальних опцій для аналізу саме соціальних мереж;
- менше половини систем поєднують у собі можливості як для візуалізації, так і для аналізу графів.

Візуалізацією графа називається побудова двовірного (рідше для специфічних вихідних даних – тривимірного) уявлення його ребер та вершин. Для візуалізації графів є різні алгоритми. Вони відрізняються один від одного вимогами до графа (зважені/незважені ребра, обмеження на максимальну кількість вершин), швидкістю отримання кінцевого результату, видом представлення графа, яке буде побудоване після закінчення роботи, а головне - метою, з якою вони застосовуються.

Таблиця 1 – Зведена таблиця програмних систем та бібліотек для проведення аналізу соціальних мереж

Назва	Функціонал	Вхідний формат	Вихідний формат	Платформа	Умови розповсюдження
AllegroGraph	База графів. Візуалізація RDF	RDF	EDF	Linux, Mac, Windows	Free i Commercial
EgoNet	Аналіз егоцентричних мереж	XML	CSV	Довільна система с Java	Open Source
Gephi	Дослідження та зміни графів	DOT, GML, GDF, GRAPHML, NET, GEXF, CSV, Database	GDF, GEXF, SVG, PNG	Довільна система с Java 1.6 и OpenGL	Open Source (GPL3)
GraphStream	Бібліотека роботи з статичними та динамічними графами	DGS, DOT, GML, Edge list	DGS, DOT, GML, Images	Довільна система с Java	Open Source
Graph-tool	Python-модуль для аналізу та візуалізації графів	DOT, GraphML	DOT, GraphML, BML, CANON, CMAP, EPS, FIG, GD, GD2, GIF, GTK, ICO, IMAP, CMAPX, ISMAP, JPEG, PDF, PLAIN, PNG, PS, PS2, SVG, SVGZ, TIF, VML, VMLZ, VRML, WBMP, XLIB	GNU/Linux, MAC	Free (GPL3)
Graphviz	Візуалізація графів	DOT	BMP, CANON, CMAP, EPS, FIG, GD, GD2, GIF, GTK, ICO, IMAP, CMAPX, ISMAP, JPEG, PDF, PLAIN, PNG, PS, PS2, SVG, SVGZ, TIF, VML, VMLZ, VRML, WBMP, XLIB	Linux, Mac, Windows	Open Source (CPL)
sigma.js	Бібліотека для візуалізації графів	GEXF, JSON, XML	GEXF, XML	Підтримка JavaScript, HTML5 и WebGL	Open Source (MITL)
Mathematica	Аналіз графів, розрахунок статистичних даних, візуалізація даних, оптимізація, розпізнавання зображення	Більш 50 форматів	Більш 50 форматів	Windows, Mac, Linux	Commercial
Wolfram Alpha	Аналіз графів и часових вибірок	Facebook API	Більш 20 форматів	Сучасні браузерери	Free i Commercial

Візуалізація графів знаходить застосування у різних практичних галузях: програмна інженерія (діаграми сутностей, потоків даних, маршрутів), управління проектами, представлення структурованих знань та даних, проектування мікросхем, соціологічні дослідження та інше. Програма, яка задовольняє всім вищезгаданим вимогам, може представляти практичну цінність для дослідників та фахівців з роботи з соціальними мережами, оскільки дозволить значно знизити час на збирання, обробку та підготовку даних до аналізу.

УДК 517.95

Брагінець О. В.,

канд. фіз.-мат. наук, доцент кафедри ІС,

Воробйова А. І.,

канд. фіз.-мат. наук, доцент кафедри ІС,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

ЗАСТОСУВАННЯ МАТЕМАТИЧНОГО ПАКЕТА MAPLE ДО ГРУПОВОГО АНАЛІЗУ ДРЧП

Абстракт. Розглянуто застосування математичного пакету Maple до відшукування симетрій систем ДРЧП на прикладі моделі Фрьоліха-Пайерлса у нерівноважному стані в одновимірному випадку.

Вступ. Груповий аналіз є ефективним інструментом при дослідженні диференціальних рівнянь. Основні ідеї групового аналізу були сформульовані Софусом Лі ще у ХІХ столітті. Група симетрій системи диференціальних рівнянь – це група, яка перетворює розв’язки цієї системи в інші її розв’язки. Дослідження симетрій диференціальних рівнянь є процес алгоритмізований, який використовується у багатьох пакетах прикладних програм.

Рівняння для рівноважних та нерівноважних станів моделі Фрьоліха-Пайерлса для електронів, що взаємодіють з фононами тільки при певних дискретних модах отримані в роботі. В одновимірному випадку нерівноважні стани описуються системою зачеплених рівнянь:

$$\frac{-\partial^2}{\partial t^2} W(t, x) + \omega_0^2 \frac{\partial^2}{\partial x^2} W(t, x) = -4\alpha \frac{\partial^2}{\partial x^2} |\Psi(t, x)|^2, \quad (1)$$

$$i \frac{\partial}{\partial t} \Psi(t, x) = \left(\frac{-1}{2m} \frac{\partial^2}{\partial x^2} - \mu \right) \Psi(t, x) + W(t, x) \Psi(t, x), \quad (2)$$

де ω_0 , α , m , μ – дійсні параметри.

Хвильова функція $\Psi(t, x)$ задовольняє рівняння Шрьодінгера з залежним від часу потенціалом, що задається дійсною функцією

$W(t, x)$, яка є розв'язком неоднорідного рівняння Даламбера з правою частиною $-4\alpha \frac{\partial^2}{\partial x^2} |\Psi(t, x)|^2$.

Очевидно, що система рівнянь (1)-(2) не буде мати дуже широкі симетрії, оскільки ліва частина рівняння (1) інваріантна відносно перетворення Лоренца, а ліва частина рівняння (2) інваріантна відносно перетворення Галілея. Перед тим, як шукати симетрії системи рівнянь (1)-(2) віднормуємо коефіцієнти, зробивши заміну:

$$W(t, x) = \mathcal{W}(t, \varkappa), \quad \Psi(t, x) = \frac{\omega_0}{2\sqrt{\alpha}} e^{i\mu t} \Psi(t, \varkappa), \quad \varkappa = \frac{x}{\omega_0}.$$

Тоді маємо систему:

$$\frac{-\partial^2}{\partial t^2} \mathcal{W}(t, \varkappa) + \frac{\partial^2}{\partial \varkappa^2} \mathcal{W}(t, \varkappa) = \frac{-\partial^2}{\partial \varkappa^2} |\Psi(t, \varkappa)|^2, \quad (3)$$

$$i \frac{\partial}{\partial t} \Psi(t, \varkappa) = \frac{-1}{2m\omega_0^2} \frac{\partial^2}{\partial \varkappa^2} \Psi(t, \varkappa) + \mathcal{W}(t, \varkappa) \Psi(t, \varkappa), \quad (4)$$

Представивши функцію $\Psi(t, \varkappa)$ через амплітуду і фазу $\Psi(t, \varkappa) = \rho(t, \varkappa) e^{i\phi(t, \varkappa)}$ зводимо систему (3)-(4) до системи вигляду:

$$F_1 = \mathcal{W}_{tt} - \mathcal{W}_{\varkappa\varkappa} - 2(\rho_x^2 + \rho\rho_{\varkappa\varkappa}) = 0, \quad (5)$$

$$F_2 = \rho_t + \frac{1}{2m\omega_0^2} (2\rho_x\phi_x + \rho\phi_{\varkappa\varkappa}) = 0, \quad (6)$$

$$F_3 = \rho\phi_t - \frac{1}{2m\omega_0^2} (\rho_{\varkappa\varkappa} - \rho\phi_{\varkappa}^2) + \rho\mathcal{W} = 0. \quad (7)$$

Будемо говорити, що система рівнянь (5)-(7) задає многовид $[F]$ у просторі V , де $V = \langle t, \varkappa, \mathcal{W}, \rho, \phi \rangle$, а F – вектор з компонентами F_1, F_2, F_3 .

Обчислення симетрій системи (5)-(7) за допомогою Maple. Розглянемо тепер, як за допомогою математичного пакету Maple можна дуже швидко знайти інфінітезимальні оператори групи інваріантності системи ДРЧП на прикладі системи (5)-(7).

Спочатку завантажуюмо пакет **PDEtools**:

> restart : with(PDEtools) :

Визначаємо залежні та незалежні змінні:

> DepVars := [w, ρ, φ](x, t);

$$\text{DepVars} := [w(x, t), \rho(x, t), \phi(x, t)]$$

Використовуємо команди diff_table та declare для компактності вводу та виводу на екран результатів:

declare ((w, ρ, φ)(t, x));

W, P, Φ := diff_table(w(t, x)), diff_table(ρ(t, x)), diff_table(φ(t, x)) :

>

w(t, x) will now be displayed as w

ρ(t, x) will now be displayed as ρ

φ(t, x) will now be displayed as φ

Вводимо самі рівняння та об'єднуємо їх в систему:

$$\begin{aligned} pde1 &:= \text{diff}(W[t], t) - \text{diff}(W[x], x) + 2 \cdot \text{diff}(P[\] \cdot P[x], x); \\ pde2 &:= P[t] + \frac{1}{2 \cdot m \cdot \omega^2} \cdot (2 \cdot P[x] \cdot \Phi[x] + P[\] \cdot \text{diff}(\Phi[x], x)); \\ pde3 &:= P[\] \cdot \Phi[t] - \frac{\text{diff}(P[x], x) - P[\] \cdot \Phi[x]^2}{2 \cdot m \cdot \omega^2} + W[\] \cdot P[\]; \end{aligned}$$

>

$$\begin{aligned} pde1 &:= w_{t,t} - w_{x,x} + 2 \rho_x^2 + 2 \rho \rho_{x,x} \\ pde2 &:= \rho_t + \frac{1}{2} \frac{2 \rho_x \phi_x + \rho \phi_{x,x}}{m \omega^2} \\ pde3 &:= \rho \phi_t - \frac{1}{2} \frac{\rho_{x,x} - \rho \phi_x^2}{m \omega^2} + w \rho \end{aligned}$$

> PDESYS := [pde1, pde2, pde3];

$$\begin{aligned} PDESYS &:= \left[w_{t,t} - w_{x,x} + 2 \rho_x^2 + 2 \rho \rho_{x,x}, \rho_t \right. \\ &\quad \left. + \frac{1}{2} \frac{2 \rho_x \phi_x + \rho \phi_{x,x}}{m \omega^2}, \rho \phi_t - \frac{1}{2} \frac{\rho_{x,x} - \rho \phi_x^2}{m \omega^2} + w \rho \right] \end{aligned}$$

Інфінітезимальні оператори для системи ДРЧП знаходимо за допомогою команди **Infinitesimals**:

> G := Infinitesimals(PDESYS);

$$\begin{aligned} G &:= [-\xi_1(t, x, w, \rho, \phi) = 1, -\xi_2(t, x, w, \rho, \phi) = 0, _ \eta_1(t, x, w, \rho, \phi) \\ &= 0, _ \eta_2(t, x, w, \rho, \phi) = 0, _ \eta_3(t, x, w, \rho, \phi) = 0], [-\xi_1(t, x, w, \rho, \\ &\phi) = 0, -\xi_2(t, x, w, \rho, \phi) = 1, _ \eta_1(t, x, w, \rho, \phi) = 0, _ \eta_2(t, x, w, \rho, \\ &\phi) = 0, _ \eta_3(t, x, w, \rho, \phi) = 0], [-\xi_1(t, x, w, \rho, \phi) = 0, -\xi_2(t, x, w, \\ &\rho, \phi) = 0, _ \eta_1(t, x, w, \rho, \phi) = 0, _ \eta_2(t, x, w, \rho, \phi) = 0, _ \eta_3(t, x, w, \\ &\rho, \phi) = 1], [-\xi_1(t, x, w, \rho, \phi) = 0, -\xi_2(t, x, w, \rho, \phi) = 0, _ \eta_1(t, x, \\ &w, \rho, \phi) = -1, _ \eta_2(t, x, w, \rho, \phi) = 0, _ \eta_3(t, x, w, \rho, \phi) = t], \\ &[-\xi_1(t, x, w, \rho, \phi) = 0, -\xi_2(t, x, w, \rho, \phi) = 0, _ \eta_1(t, x, w, \rho, \phi) = \\ &-t, _ \eta_2(t, x, w, \rho, \phi) = 0, _ \eta_3(t, x, w, \rho, \phi) = \frac{1}{2} t^2] \end{aligned}$$

Перевірка:

> *map(SymmetryTest, [G], PDESYS);*

[{0}, {0}, {0}, {0}, {0}]

Отже, за допомогою пакету Maple були знайдені такі ж самі інфінітезимальні оператори групи симетрій системи (5)-(7).

Використовуючи команду *SymmetryTransformation* можна, для кожного із п'яти отриманих операторів, знайти перетворення змінних t, x, W, ρ, ϕ , які залишають систему (5)-(7) інваріантною.

> *NewVars := [f, g, h](r, s);*

NewVars := [f(r, s), g(r, s), h(r, s)]

> *SymmetryTransformation (G[5], DepVars, NewVars);*

$$\left\{ \begin{array}{l} r = x, s = t, f(r, s) = w(x, t) - \varepsilon t, h(r, s) = \frac{1}{2} \varepsilon t^2 + \phi(x, t), g(r, \\ s) = \rho(x, t) \end{array} \right\}$$

Тут визначені нові залежні f, g, h та незалежні r, s змінні, ε – груповий параметр.

Висновки. Знання групи симетрій диференціального рівняння або системи диференціальних рівнянь дозволяє використовувати метод симетрійної редукції для побудови точних розв'язків цього рівняння (системи) [1,4]. Для багатовимірних рівнянь – це інколи єдиний ефективний спосіб отримання класів точних розв'язків. Сучасний математичний пакет Maple дає можливість дуже швидко обчислити симетрії системи ДРЧП та перейти до наступного етапу побудови точних розв'язків для цієї системи.

УДК:62-049.5(043,2)

Донченко М. В.,

канд. техн. наук, доцент кафедри ІС,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

ВРАХУВАННЯ НАДІЙНОСТІ ПРИ ОЦІНЦІ БЕЗПЕКИ СИСТЕМ

Є дві базові оцінки якості систем – надійність і безпека. Ці дві оцінки мають багато спільного і, при цьому, можуть суттєво відрізнятися. В обох напрямках проведено багато досліджень. Наведемо їхні

визначення. *«Надійність системи – це її здатність до безвідмовної роботи протягом заданого проміжку часу у визначених умовах».* *«Безпека – це такі умови, в яких перебуває складна система, коли дія зовнішніх факторів і внутрішніх чинників не призводить до процесів, що вважаються негативними по відношенню до даної складної системи у відповідності до наявних, на даному етапі, потреб, знань та уявлень»* (Вікіпедія). При всьому тому, що безпеці присвячено дуже багато розробок, питання оцінки безпеки ще до кінця не вирішене і це не тому, що воно не важливе. Скоріше, навпаки, але тут слід визнати специфіку і складність отримання такої оцінки в силу багатофакторності впливів, які приводять до негативних ситуацій (їх частіше називають: аварійних ситуацій (АС)). Якщо методи оцінки надійності детально розроблені і вже багато років використовуються, то відносно безпеки таких глибоких напрацювань, на жаль, не знайдено. Тому дане дослідження має на меті поєднати ці дві важливі категорії якості, визначити їх взаємний «вплив» і отримати можливість, якоюсь мірою, оцінки безпеки через надійність.

Як уже було відмічено вище, ці характеристики оцінюють властивості системи, якби в одному напрямку, але з різних точок зору. Обидві характеризують функціонування системи і його припинення, але там, де появляється відмова – надійність припиняється, а небезпека тільки розвивається. Надійність характеризується «появами відмов», точніше їх не появою. Відмова – це неможливість нормального функціонування в заданих умовах. А безпека – це відсутність АС в реальних умовах. Точніше – це відсутність негативного впливу на життя і здоров'я людей, деструктивного впливу на саму систему і на довкілля. Розглянемо ці характеристики детальніше.

Відмова – це ситуація, в якій система не може нормально функціонувати в заданих умовах. Ця подія може виникнути в результаті накопичення негативних змін в самій системі (старіння, знос і т.п.). Цей процес базується на фізичних процесах змін в матеріалах деталей системи (старіння) чи змін геометрії механічних з'єднань між елементами системи (знос). Ці процеси досить складні, але існує достатньо великий об'єм досліджень у цьому напрямку, що надає можливість прогнозувати, з певною ймовірністю, появу критичних змін, які можуть привести до відмови. Відмови, які появляються в результаті таких змін називають поступовими. Інший вид відмов – це відмови, які виникають в результаті випадкової появи зовнішніх екстремальних впливів на систему, що, в результаті, може призвести до деструктивних змін в конструкції системи і неможливості нормального функціонування і, навіть, до АС. Появу таких впливів досить складно спрогнозувати,

тому ймовірність їх появи визначають на основі статистичних даних експлуатації системи чи спеціальних випробувань.

Що стосується безпеки, то тут все набагато складніше. В нашому дослідженні ми будемо розглядати тільки ті системи, для яких характерна можливість появи АС після відмови системи. Безпека забезпечується відсутністю загроз для людей, самої системи і довкілля. В залежності від призначення системи, її конструктивних особливостей і умов використання можна виділити загрози і напрямки їх можливого впливу та рівня негативних змін при появі відмови:

- може призвести до деструктивних змін в самій системі;
- провокує до негативних змін оточуючих систем;
- може бути небезпечна для операторів і людей навколо;
- може призвести до катастрофічних змін у системі і довкіллі.

Звичайно, така градація достатньо умовна, але її визначення ставить вимоги як до проектування системи, та і до оцінки її безпеки.

За аналогією з надійністю оцінку безпеки зручніше виконувати за явною появою небезпечних подій (або АС). Властивість надійності системи втрачається при появі відмови. Поява відмови може бути причиною появи негативних впливів на людей, довкілля і на саму систему. В нашому випадку, відмова стає очевидною загрозою як цілісності системи, так і для здоров'я і життя людей та негативних змін довкілля. Тому, маючи параметри надійності (ймовірність відмови), можна говорити, що є передумови появи АС.

Розглянемо події:

А – відмова не настала;

Б – поява відмови;

В – настала АС;

С – АС не настала.

Для нашого випадку, події **А** і **С** нас не цікавлять. Подія **Б** розглядається при умові, що вона передує події **В**. Тоді, події **Б** і **В** несумісні тому, що спочатку появляється подія **Б**, а потім **В**. Більше того, подія **В** залежна від того, чи появилася подія **Б**. Подія **Б** не залежить від події **В** тому, що при появі події **В** розгляд появи події **Б** втрачає смисл.

Тоді подія поява АС може настати у випадку відмови і аварійної ситуації в результаті її появи. Оскільки ці події взаємопов'язані і часовий проміжок між ними практично нульовий, то можна говорити про спільну появу обох подій.

$$N=BB$$

Якщо перейти до ймовірностей появи подій, то ми отримаємо:

$$P(N)=P(B)*P(B/B),$$

де $P(B) = Q(t)$ – ймовірність появи відмови системи, яка може привести до АС;

$P(B/B)$ – ймовірність настання аварійної ситуації в результаті відмови системи.

Настання АС край небажана подія і, як правило, вона фіксується. За статистикою АС маємо можливість отримати розподілення щільності ймовірності і ймовірності настання АС. Маючи розподіл ймовірності відмови системи нескладно отримати ймовірність появи АС.

$$N(t) = Q(t) * P_{АС}(t).$$

Відповідно, для оцінки безпеки зручніше мати ймовірність протилежної події – ймовірність не настання АС $S(t)$

$$S(t) = 1 - N(t) = 1 - [Q(t) * P_{АС}(t)].$$

Оскільки настання АС пов'язано в великими втратами в, тому числі, і серед людей, ведуться серйозні дослідження в напрямку зменшення чи обмеження ризику появи АС. Це – цілеспрямоване проектування, організації експлуатації, врахування «людського фактору», удосконалення систем захисту.

Сучасні САПР дозволяють значно підвищити ефективність проектування з метою покращення показників надійності, зменшення можливості появи загроз АС, введення конструктивного, експлуатаційного і організаційного систем захисту від можливих загроз та врахування «людського фактору».

Крім цього, існують умови, які сприяють чи провокують появу АС. Вони можуть бути як зовнішні, так і внутрішні. Врахування таких умов дозволить точніше оцінювати рівень безпеки і направляти дослідження і проектування на зменшення рівня їх впливу. Очевидно, щоб настала АС необхідно спільне настання відмови, можливості настання АС і умов для її появи.

$$S(t) = 1 - N(t) = 1 - [Q(t) * P_{АС}(t) * U(t)],$$

де $U(t)$ – ймовірність появи сприятливих умов для настання АС.

Логічно було б врахувати й ефективність систем захисту, якщо вони є. В цьому випадку нас цікавить ймовірність зниження можливості настання АС системами захисту $\{Z(t)=0\}$ – захисту немає чи не спрацьовує; $\{Z(t)=1\}$ – повний захист. Тоді для спільного настання перерахованих подій формула прийме вигляд:

$$1 - N(t) = 1 - \{Q(t) * P_{АС}(t) * U(t) * [1 - Z(t)]\},$$

де $1 - Z(t)$ – ймовірність зниження загрози АС при наявності систем захисту.

Висновки. Запропонована можливість оцінки безпеки систем за показником її надійності. Метод може бути корисним при проектуванні й аналізі безпеки складних систем.

ПОРІВНЯЛЬНИЙ АНАЛІЗ ПРОГНОСТИЧНИХ МОДЕЛЕЙ НА ОСНОВІ ДЕРЕВ РІШЕНЬ

При розв'язанні завдань числового прогнозування частіше всього використовують традиційні регресійні методи, проте числові дерева рішень мають явні переваги. Дерева рішень краще підходять для завдань із великою кількістю ознак або з великою кількістю складних нелінійних залежностей між ознаками та результатом. Дерева для вирішення завдань числового прогнозування поділяються на дві категорії: регресійні дерева та дерева моделей. Регресійне моделювання також робить припущення про дані, які практично часто порушуються, але у випадку з деревами цього не відбувається.

Для вирішення завдань прогнозування використовується методологія, яка складається з наступних етапів.

Етап 1. Збір даних. На цьому етапі здійснюється збір початкових даних. В даному випадку для створення прогностичної моделі за допомогою моделей різного типу, що базуються на деревах, використовується попередній набір даних *Combined Cycle Power Plant*.

Етап 2. Дослідження та підготовка даних. У порівнянні з іншими типами моделей машинного навчання, однією з переваг дерев є те, що вони здатні обробляти різні типи даних без попередньої обробки. Це означає, що не потрібно нормалізувати чи стандартизувати ознаки. Незважаючи на те, що дерева дуже стійкі до ненадійних даних, завжди доцільно перевіряти дані на наявність серйозних проблем. На основі попереднього аналізу набору даних передбачається, що дані є надійними.

Етап 3. Навчання моделі на даних. Першим кроком до етапу навчання моделі на даних є підготовка даних до створення моделей шляхом поділу набору на навчальний та тестовий. Для кожного вузла в дереві вказано кількість прикладів, що досягають точки ухвалення рішення. Більш детальний опис відповідності дерева, включаючи середньоквадратичну помилку для кожного з вузлів та загальну міру важливості ознак, отримують з повного звіту моделі. Усі предиктори, що використовуються в моделі, згідно зі звітом, ранжовані за ступенем важливості.

Для того щоб зрозуміти, як дерево працює, простіше використовувати візуалізацію. Для побудованої моделі регресійного дерева деревоподібна діаграма представлена на рисунку 1. Подібні візуалізації допомагають розповсюджувати результати побудови регресійних дерев, оскільки їх легко зрозуміти. Числа, показані в кінцевих вузлах, є прогнозованими значеннями для прикладів, що досягають вузла. Таким чином, за діаграмою можна визначити ключові фактори, які дозволяють набувати високих значень чистого погодинного виходу електроенергії (PE) (для набору *Combined Cycle Power Plant*).

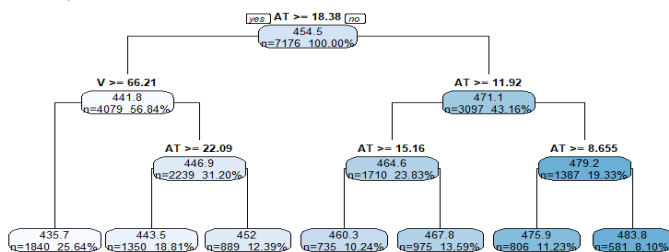


Рисунок 1 – Візуалізація моделі регресійного дерева

Етап 4. Визначення ефективності моделі. Побудована модель регресійного дерева використовується для прогнозування на основі тестових даних. Для перевірки якості прогнозів для моделі регресійного дерева обраховані значення метрик якості. Кореляція між прогнозованими та реальними значеннями забезпечує простий спосіб вимірювання ефективності моделі. Значення кореляції 0.95 для даного набору дуже гарний показник. Однак воно лише показує, наскільки сильно прогнози пов'язані з реальними значеннями, але не те, наскільки далекі прогнози справжніх значень.

Ще один спосіб оцінити ефективність моделі – визначити, наскільки далекий у середньому прогноз від істинного значення. Відповідь це питання може дати обчислення середньої абсолютної помилки (*Mean Absolute Error, MAE*). Для набору даних *Combined Cycle Power Plant* обчислення показують, що в середньому різниця між прогнозами моделі регресійного дерева та фактичними вихідними значеннями чистого погодинного виходу електроенергії становить приблизно 4,2 МВт. При різниці між максимальним і мінімальним значенням у тестовому наборі даних 75 МВт це говорить про те, що модель працює досить добре.

Етап 5. Підвищення ефективності моделі. Для покращення якості прогнозів виконується процедура скорочення дерева. У цьому випадку

перехресна перевірка вибирає найскладніше дерево. Однак для створення альтернативної моделі дерево скорочене до 5 кінцевих вузлів. Візуалізація моделі скороченого регресійного дерева представлена на рисунку 2а.

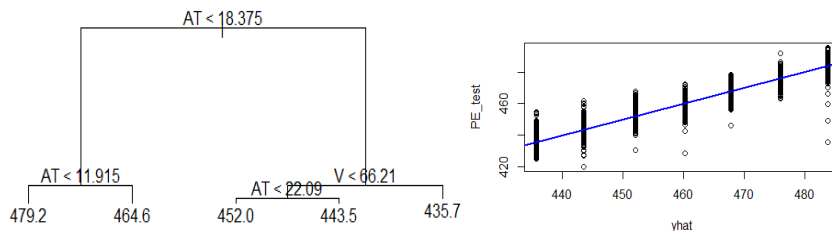


Рисунок 2 – а) Візуалізація моделі скороченого регресійного дерева;
 б) Прогнозовані значення погодинного виходу електроенергії
 з використанням моделі регресійного дерева

Відповідно до результатів перехресної перевірки для отримання передбачень використовується нескорочене дерево. Прогноз виконується на тестовій вибірці даних. На рис.2б представлені прогнозовані значення чистого погодинного виходу електроенергії на тестовому наборі даних з використанням моделі регресійного дерева

Бегінг та випадковий ліс. Бутстреп-агрегування, або *бегінг*, є процедурою загального призначення, що дозволяє знизити дисперсію статистичної моделі. Ця процедура особливо корисна у контексті дерев рішень. Відомо, що усереднення кількох спостережень знижує дисперсію. Отже, природним способом зниження дисперсії і як результат, підвищення точності прогнозів того чи іншого методу статистичного навчання є взяття великої кількості навчальних вибірок з генеральної сукупності, побудова прогновної моделі по кожній навчальній вибірці та усереднення отриманих прогнозів. Отже, метод бегінгу має на увазі створення великої кількості копій вихідного набору даних шляхом застосування *бутстрепу*, побудова окремого дерева рішень для кожної з цих копій та наступне поєднання всіх дерев для створення однієї прогновної моделі. Важливо відзначити, що кожне дерево будується за певною бутстреп-вибіркою незалежно від інших дерев.

Звіт по бегінг-моделі для набору даних *Combined Cycle Power Plant* свідчить, що середнє квадратів залишків моделі на навчальній вибірці даних – 10,97. Аргумент $mtry=4$ показує, що з створення кожного вузла дерева розгляду підлягають все 4 предиктора, тобто необхідно виконати бегінг. У бегінг-моделі на контрольних даних MSE (*mean*

squared error – середньоквадратична помилка прогнозу) становить 11,97 – це трохи гірше, ніж на навчальній вибірці даних, але на 2,5 краще ніж помилка, отримана для обрізаного дерева. Прогнозовані значення чистого погодинного виходу електроенергії на тестовому наборі даних із використанням бегінг-моделі наведено на рисунку 3.

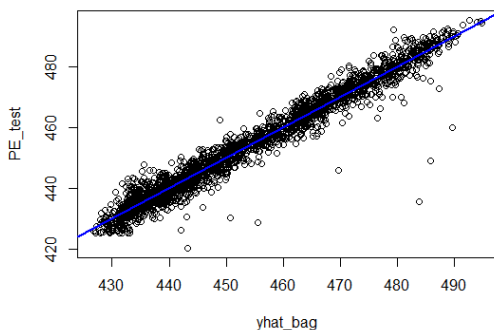


Рисунок 3 – Прогнозовані значення чистого погодинного виходу електроенергії на тестовому наборі даних із використанням бегінг-моделі

З метою покращення якості моделі змінюється кількість дерев, що створювались у бегінг-моделі функцією «випадкового лісу». Побудова моделі випадкового лісу (*Random Forest*) виконано аналогічним чином.

Бустінг. Подібно до бегінга, бустінг є загальним підходом, який можна застосовувати до багатьох статистичних методів регресії та класифікації. Розглянемо бустінг у контексті дерев рішень. Алгоритм бустінгу нагадує алгоритм бегінга, проте дерева будуються послідовно: кожне дерево вирощується з використанням інформації щодо раніше вирощених дерев. Бутстреп-вибірки в ході реалізації бустінгу не створюються, а натомість кожне дерево будується за певним чином модифікованим вихідним набором даних.

На відміну від побудови одного великого дерева рішень, яке зводиться до тісної апроксимації даних та потенційно може призвести до перенавчання, бустінг виконує *повільне навчання*. Графік відносної важливості змінних та відповідні обраховані значення показника важливості представлені нижче (рис.4).

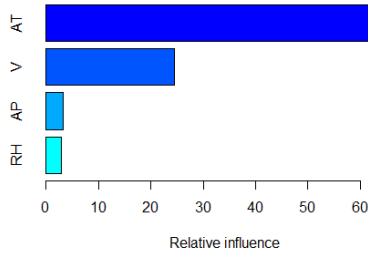


Рисунок 4 – Графік відносної важливості змінних набору даних *Combined Cycle Power Plant*

Візуально на гістограмі видно, що *AT* і *V* значно перевершують за важливістю всі інші змінні. Подаємо також графіки приватної залежності («*partial dependence plot*») для цих двох змінних. Графіки цього типу ілюструють приватні ефекти тих чи інших змінних відгуку після фіксації ефектів інших змінних постійному рівні (рис. 5).

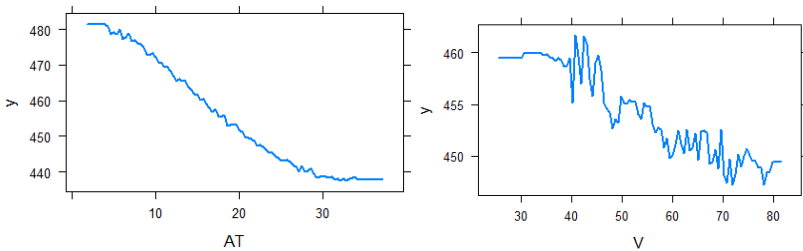


Рисунок 5 – Графіки приватної залежності для змінних *AT* та *V* набору даних *Combined Cycle Power Plant*

За графіками видно, що величина чистого погодинного виходу електроенергії очікувано зменшується зі збільшенням *AT*. У разі збільшення *V* спостерігається загальна тенденція зниження чистого погодинного виходу електроенергії. Розроблена бустінг-модель застосовується для обчислення прогнозів за контрольними даними та показників їх якості.

Вибір кращої моделі, що базується на деревах, виконується на основі порівняння показників якості прогнозу (табл.1). В таблиці 1 наведено порівняльний аналіз показників якості прогнозу на різних прогнозних моделях. Для оцінки якості прогнозів були використані різні показники якості. Кращою за сукупністю показників якості є бустінг-модель. Особливістю прогнозних моделей побудованих на

деревах рішень є їх простота та ефективність, та можливість підвищувати їх ефективність. Отримані результати вказують на добру якість прогнозних значень з використанням регресійної моделі.

Таблиця 1 – Порівняльна таблиця показників якості прогнозу

Тип моделі	Середня абсолютна похибка (mean absolute error, MAE)	Середньоквадратична похибка (mean squared error, MSE)	Квадратний корінь із середньоквадратичної похибки (root mean squared error, RMSE)	Кореляція, Cor
Регресійне дерево	4,2	29,3	5,41	0,95
Бегінг	2,36	11,97	3,46	0,98
Випадковий ліс	2,35	11,90	3,45	0,98
Бустінг	2,28	11,02	3,32	0,98

УДК 004.942

Кулаковська І. В.,

канд. фіз.-мат. наук, доцент кафедри ІС,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

АНАЛІЗ РОЗВИТКУ СЛАБКОСТРУКТУРОВАНИХ СИСТЕМ ЗА ДОПОМОГОЮ МЕТОДОЛОГІЇ КОГНІТИВНИХ КАРТ

Дослідники, які цікавляться когнітивними процесами людини, вже давно використовують комп'ютерне моделювання, щоб спробувати визначити принципи пізнання і планування. Стратегія полягає в тому, щоб побудувати обчислювальні моделі, які втілюють набір принципів, а потім вивчити, наскільки добре ці моделі відображають продуктивність людини в когнітивних завданнях. Обчислювальне моделювання має як сильні, так і слабкі сторони в його корисності для сприяння розвитку когнітивних теорій. Робоча реалізація моделі дає можливість перевірити повноту та внутрішню узгодженість основних теоретичних засад, а також створити детальні, кількісні прогнози моделі при тестуванні в нових умовах:

– намагаючись використовувати інформаційні технології для прийняття рішень у соціально-економічних, організаційних, політичних та інших соціально значимих сферах, ми часто стикаємося з так званими погано структурованими дисциплінарними сферами, які мають такі специфічні характеристики;

- описує систему понять (факторів, понять) предметної області та їх взаємозв'язки, але визначення є неповними – багато факторів, зв'язків і параметрів виявляються лише при подальшому аналізі моделі;

- предметна область динамічна, тобто її параметри залежать від часу внаслідок взаємодії факторів, що її породжують;

- коли основні параметри системи задаються (значення факторів, ступінь впливу одних факторів на інші фактори), вони є якісними і неточними, а також неоднозначними змінними, або дають лінійну вербальну (мовну) шкалу оцінок;

- значення системних параметрів в основному отримують шляхом експертних інтерв'ю, тому це суб'єктивні оцінки; навіть ті фактори та установки, які можуть бути якісно виражені в результаті обробки економічної статистики, опитувань тощо, можуть бути оцінені різними експертами; отже, кінцеві значення параметрів моделі участі експерта є результатом його приватної суб'єктивної обробки цих даних, у тому числі вибору одного з розрахункових значень (або їх зваженої згортки) з урахуванням достовірності даних, джерела репутації тощо;

- наперед встановлених альтернатив стану системи небагато, і вони, як правило, очевидні, неочевидні альтернативи, найкращі з яких, зазвичай, є єдиними альтернативами, що виникають під час аналізу ситуації.

При оцінці виявлених альтернатив необхідно враховувати динаміку ситуації, тобто прогноз її розвитку під дією різних факторів, появу побічних наслідків тощо.

Когнітивне моделювання – зазвичай комп'ютерні системи використовуються для визначення найбільш ефективних управлінських рішень та/або сценаріїв розвитку системи на основі вибору концепцій (факторів), які характеризують стан системи кількісно чи якісно, а також для оцінки взаємодії кожного фактору з іншими. Як правило, моделювання на основі когнітивних методів виконується командами людей (менеджерів, експертів), відповідальних за розробку якоїсь погано структурованої системи.

Аксельрод запропонував підхід когнітивного моделювання для аналізу та подальшого прийняття рішень у погано визначених ситуаціях. Він заснований на моделюванні суб'єктивного погляду кожного експерта на систему, включаючи:

- структурування системи: модель представлення знань експерта-аналітика у вигляді знакового орграфа (когнітивної карти) (F, W) , де F – множина концептів ситуації, W – множина причинно-наслідкових зв'язків між концептами системи;

- методи аналізу системи.

В наш час методологія когнітивного моделювання стрімко розвивається в напрямку покращення апарату аналізу та моделювання системи.

Основні етапи когнітивного моделювання включають:

ЕТАП 1 Обґрунтування мети і завдань дослідження.

ЕТАП 2 Вивчення рівня складності ситуації з позицій поставленої мети: збирання, систематизація, аналіз наявної статистичної та якісної інформації про об'єкти управління та їх зовнішнє середовище, визначення притаманних досліджуваній ситуації вимог, умов і обмежень

ЕТАП 3 Виділення основних концептів, що впливають на розвиток економіко- організаційної ситуації.

ЕТАП 4 Визначення взаємозв'язків між концептами шляхом розгляду їх причинно-наслідкових зв'язків (побудова когнітивної карти у вигляді орієнтованого графа).

ЕТАП 5 Дослідження сили взаємодії різних концептів. Для цього можуть бути використані як математичні моделі, які описують деякі точно визначені кількісні залежності між концептами, так і суб'єктивні уявлення експерта щодо формалізації якісних взаємозв'язків між концептами. На етапах 3-5 будується остаточна когнітивна модель системи. В подальшому вона подається у вигляді функціонального графа.

ЕТАП 6 Перевірка адекватності когнітивної моделі на реальній ситуації (верифікація когнітивної моделі).

ЕТАП 7 Визначення на основі когнітивної моделі можливих варіантів подальшого розвитку системи, виявлення шляхів та способів впливу на систему з метою досягнення бажаних результатів, запобігання небажаним наслідкам, тобто вироблення стратегії управління. Завдання цільових, бажаних напрямків і сили зміни тенденцій процесів у ситуації. Вибір комплексу заходів (сукупності керуючих факторів), визначення їх можливої і бажаної сили і спрямованості впливу на ситуацію (конкретно-практичне застосування когнітивної моделі).

Використання когнітивного підходу під час дослідження складних слабкоструктурованих систем, що в даному випадку розглядається в контексті безперервної «взаємодії» результатів дослідження суб'єктом складного об'єкта і самого об'єкта, необхідно розглядати щонайменше з трьох позицій:

- позиція «корисності» («шкоди») наслідків прийнятих рішень для об'єкта;
- позиція отримання нових знань про об'єкт;
- корисність отриманих знань для самого суб'єкта.

У першому випадку особливо важливо розуміти «ризик людського фактору». З огляду на слабкоструктурованість проблем великих систем, ускладненість «правильного» збору та аналізу інформації про них, «правильність» поставленої мети дослідження (тлумачення механізму явищ прогнозування, управління, генерації управлінських рішень, їх оцінювання, прийняття рішень тощо), слід замислитися про необхідність зручного інструмента для проведення досліджень і підтримки прийняття рішень.

Когнітивний аналіз розглядається як один із найбільш потужних інструментів дослідження нестабільної і слабо структурованої системи. Він сприяє кращому розумінню існуючих в середовищі проблем, виявленню суперечностей та якісному аналізу процесів, що протікають. Суть когнітивного (пізнавального) моделювання – ключового моменту когнітивного аналізу полягає у відображенні найскладніших проблем і тенденцій розвитку системи в більш простому вигляді в моделі, дослідженні можливостей виникнення кризових ситуацій та знаходженні способів і умов їх вирішення в модельній ситуації. Використання когнітивних моделей якісно підвищує обґрунтованість прийняття управлінських рішень у складній і швидкоплинній обстановці, позбавляє експерта від «інтуїтивного блукання», економить час на осмислення та інтерпретацію подій, що відбуваються в системі.

УДК 004.82

Сіденко Є. В.,

канд. техн. наук, доцент, доцент кафедри ІС,

Кондратенко Г. В.,

канд. техн. наук, доцент, доцент кафедри ІС,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

Трухов А. С.,

software engineer, middle.net developer,
Globallogic, м. Миколаїв, Україна

МАШИННЕ НАВЧАННЯ ДЛЯ МАРШРУТИЗАЦІЇ ДРОНУ

У роботі розглянуто застосування машинного навчання для ефективної маршрутизації безпілотного літального апарату (БПЛА, дрон) у просторі, що дозволить з більшою точністю переміщатися визначеним маршрутом на пересіченій місцевості.

Сучасні дрони мають широкую сферу застосувань: фотографування різних ділянок або об'єктів, збір параметрів стану повітряних мас на

висоті, аналіз ситуації на місцевості, доставка вантажів до певного місця, пошукові роботи та інше. Для програмування дронів необхідно зареєструвати обліковий запис розробника. Після чого, завантажити SDK з сайту виробника. Наприклад, у компанії DJI передбачені різні SDK в залежності від того, що планується програмувати, зокрема управління корисним навантаженням або самим польотом. Важливо вибрати правильний SDK і переконатися, що підтримується наявне обладнання, наприклад, додаткові датчики дрона або мобільний пристрій.

Застосування машинного навчання включає в себе створення моделі, яка навчається на деяких навчальних даних, а потім може обробляти дані для складання прогнозів польоту і маршрутизації. Для систем машинного навчання використовуються різні типи моделей. Навчання з підкріпленням є областю машинного навчання та визначає, які заходи агенти повинні вживати в середовищі з метою максимізації сукупної винагороди. Навчання з підкріпленням є однією з трьох основних парадигм машинного навчання, поряд із навчанням з вчителем і навчанням без вчителя. Навчання з підкріпленням відрізняється від навчання з вчителем тим, що не вимагає подання вхідних та вихідних пар і не вимагає явного виправлення неоптимальних дій. Замість цього основна увага приділяється пошуку балансу між дослідженням території та експлуатацією поточних знань.

Основними бібліотеками для створення моделей штучного інтелекту, які підтримуються ігровим двигуном Unity є Acord.Net, CNTK, Tensorflow та Unity ML-Agents.

Перш ніж розпочати вбудовування штучного інтелекту слід створити звичайну сцену, на якій будуть розміщені агенти та інші об'єкти. Для вирішення задачі маршрутизації дрону було створено тренувальне середовище, яке змусить виконувати рухи літального апарату по всіх осях (рисунок 1).

Розроблений агент має складну будову, було розроблено прототип, який підходить для доставки та має складний каркас для підвищення стабільності (рисунок 2).

Правильне визначення нагород та покарання має велике значення при використанні методу машинного навчання з підкріпленням. Була розроблена власна система, яка враховує огинання перешкод, надаючи нагороду кожного кадру рівну шляху, який подолав агент за останній кадр. За допомогою векторів, агент буде спостерігати за середовищем, всього використано 17 векторів: 8 діагональних під кутом 20 та 45 градусів, вперед під кутом 45, 65, 90, 110 та 155, вгору, вниз, вліво та вправо.

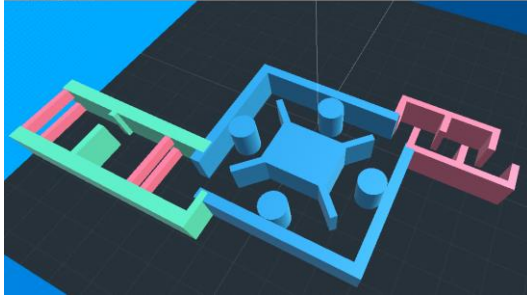


Рисунок 1 – Тренувальне середовище



Рисунок 2 – Візуальне відображення агенту

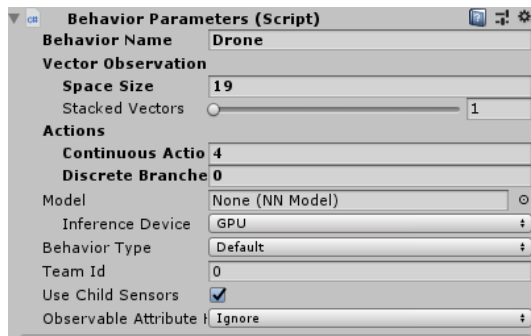


Рисунок 3 – Налаштування параметрів моделі агенту

У процесі інтеграції штучного інтелекту необхідно створити екземпляр мозку для навчання та налаштувати його, а саме вказати кількість вхідних параметрів, описати характер вихідних параметрів, їх кількість та вказати пристрій для навчання: відеодро чи ядро

процесора. Посилання на мозок має бути у кожного агента та у академії (рисунок 3).

Згідно заданим параметрам була розроблена архітектура нейронної мережі, яка містить вхідний шар, два приховані шари та вихідний шар (рисунок 4).

Одним з важливих етапів є тестування поведінки в новому середовищі. Додатково до перевірки правильної поведінки агента також тестується його здатність досягати заданої цілі. Щоб перейти до більш реального середовища було створено прототип міста з центральним парком посередині (рисунок 5).

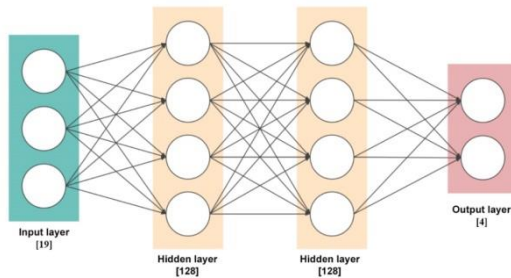


Рисунок 4 – Архітектура створеної нейронної мережі

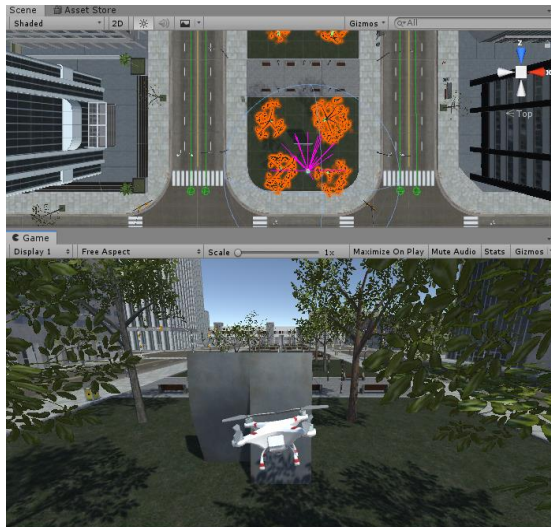


Рисунок 5 – Маршрутизація дрону в тестовому середовищі

В тестовому середовищі агент успішно досягнув кінцевої точки, не зіштовхнувшись з жодною з перешкод. В ході тестування було перевірено здатність агента орієнтуватися у просторі, виконувати всі базові рухи по 4 осям, досягти кінцевої точки та огинати перешкоди в різних середовищах.

УДК 681.5

Скакодуб О. С.,
аспірант кафедри ІС,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

АВТОМАТИЗАЦІЯ ПРОЦЕСІВ КЕРУВАННЯ БЕЗПІЛОТНИМИ ЛІТАЛЬНИМИ АПАРАТАМИ

В доповіді розглядаються питання розробки SCADA-системи для керування польотом безпілотних літальних апаратів (БПЛА). Виявлені основні керовані змінні системи керування БПЛА, а також розроблена функціональна структура її SCADA-системи.

В наш час набуває розвитку розробка БПЛА на основі комп'ютерних систем керування для моніторингових та транспортних операцій. Керування польотом БПЛА є складним технічним завданням, оскільки воно потребує контролю швидкості обертання двигунами, кутом нахилу та вхідними даними про координати перешкод. В процесі руху БПЛА на його політ можуть впливати інерційні моменти, сила тертя об повітря та високі або низькі температури. Надійність і точність системи керування обумовлена її перешкодозахищеністю по відношенню до шумів в вимірювальній апаратурі з урахуванням похибок і помилок вимірювання фазових координат БПЛА. Підвищений рівень складності та багатофункціональності сучасних БПЛА вимагає створення нового класу систем керування, що базуються не на статичній інформації баз даних, а на поточних даних, які формуються в масштабі реального часу в ході протікання відповідних технологічних процесів. Сучасними системами такого типу є системи SCADA (Supervisory Control and Data Acquisition), що мають ієрархічну розгалужену структуру та включають в себе промислові комп'ютери, технологічні контролери, автоматизовані пости оператора, програматори, засоби, що реалізують промислові мережі. Застосування SCADA-системи забезпечить реалізацію керування польотом БПЛА без залучення оператора по заданій

траєкторії, що, в свою чергу, дозволить значно підвищити ефективність роботи та загальні економічні показники БПЛА.

Розроблена функціональна структура SCADA-системи керування польотом БПЛА представлена на рис. 1, де прийнято наступні позначення: ПК – промисловий комп’ютер; ПЛК – програмований логічний контролер; ТПЛК – технологічний ПЛК; МАВ – модуль аналогового виводу; МЗД – модуль збору даних; ГП – гіроскоп для визначення положення БПЛА в просторі; БР – барометр для визначення висоти польоту БПЛА; 4G – бездротовий зв’язок для обміну інформацією між БПЛА та ПК; ЕПТ1, ЕПТ2, ЕПТ3, ЕПТ4 – електроприводи гвинтів БПЛА.

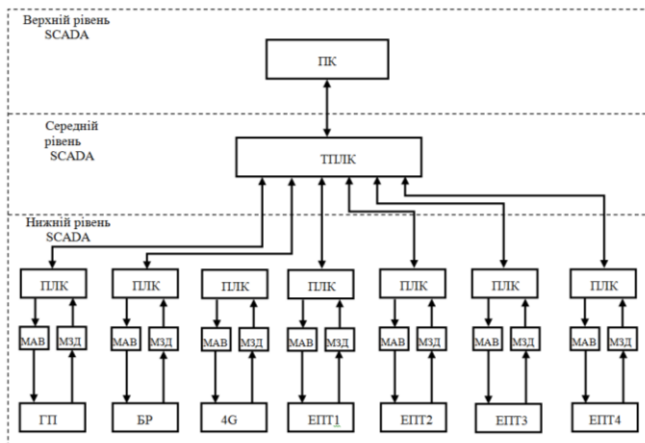


Рисунок 1 – Функціональна структура SCADA-системи керування польотом БПЛА

Основними керованими координатами даної системи є: координати положення БПЛА в просторі x , y , z , що отримані з гіроскопа, висота польоту БПЛА h , що визначається барометром, швидкості польоту за основними координатами V_x , V_y , V_z , відстані до перешкод у тривимірному просторі. Крім того, компоненти БПЛА є складними об’єктами керування, що функціонують в умовах дії невизначених параметричних та координатних збурень, таких як температура, швидкість вітру, вологість повітря, щільність повітря та атмосферний тиск.

Аналіз алгоритмів та схемотехнічних рішень для проектування і реалізації керуючих пристроїв наведеної SCADA-системи керування польотом БПЛА показує доцільність використання принципів інтелектуального керування, що спираються на теорії нечітких множин, штучних нейронних мереж та еволюційних обчислень. Керуючі пристрої, що розроблені на базі нечіткої логіки, штучних нейронних мереж та еволюційних алгоритмів, на даний час успішно використовуються в сучасних системах автоматизації різнотипних технологічних об'єктів і процесів.

УДК 004.9:004.1

Таранов М. О.,
викладач кафедри ІС,
Кулаковська І. В.,
канд. фіз-мат. наук, доцент кафедри ІС,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

НАВЧАЛЬНІ ЗАНЯТТЯ ДЛЯ СТУДЕНТІВ ПІД ЧАС ВОЄННОГО СТАНУ ЗА ДОПОМОГОЮ ВІДЕО З YOUTUBE, КОНФЕРЕНЦІЙ GOOGLE MEET ТА СИСТЕМИ ЕЛЕКТРОННОГО НАВЧАННЯ MOODLE

Із перших днів повномасштабного вторгнення Росії в Україну, навчання в Миколаївській області було призупинено. Через місяць після початку нового етапу війни, в університеті ЧНУ імені Петра Могили відновився освітній процес. Це дало можливість продовжувати виконувати свою основну сферу діяльності – готувати майбутніх професіоналів, які зможуть стати рушійним фактором для розвитку й процвітання України.

На думку ряду науковців Заходу (F. Bodendorf, H. Dichanz, A. Ernst, H. Lobin, M. Stührenberg, G. Rehm) та вітчизняних науковців (Н. Морзе, В. Кухаренко, О. Рибалко, Н. Сиротенко, Б. Шуневич), метою будь-якого проекту електронного навчання є забезпечення якісного здобування знань студентами за допомогою інформаційних технологій.

Уже більше року (карантин) ми працюємо в умовах, що негативно позначаються на навчальному процесі. Процес треба змінювати, підлаштовуючись під виклики часу. Дистанційне навчання у тому вигляді, що є зараз, не завжди ефективне та якісне. Перед нами стоїть

виклик забезпечити достойний рівень освіти, не зважаючи на період дії воєнного стану. Нам варто розраховувати, що тенденції дистанційної освіти вже не повернуть нас до формату освіти п'ятирічної чи десятирічної давності. Ми повинні пам'ятати, що є нормативна база, яка вимагає збереження особистих даних в межах країни, всі сервіси цього не забезпечують, але їх можна використовувати як доповнення до навчання. Важливо зберігати свої напрацювання не на якомусь далекому сервері, а на сервері в межах закладу. Важливо використовувати безкоштовні ресурси з відкритим кодом, таку платформу яка дасть змогу користуватись нею безкоштовно. Тому ми обрали Moodle, адже вона створювалась як система для навчання та дозволяє організувати освітній процес, проте цього разу ми доповнили її засобами відео трансляції з Youtube.

```
<iframe width=«560» height=«315»  
src=«https://www.youtube.com/embed/FGDgAv15VM0» frameborder=«0»  
allow=«accelerometer; autoplay; clipboard-write; encrypted-media;  
gyroscope; picture-in-picture» allowfullscreen=«>></iframe>
```

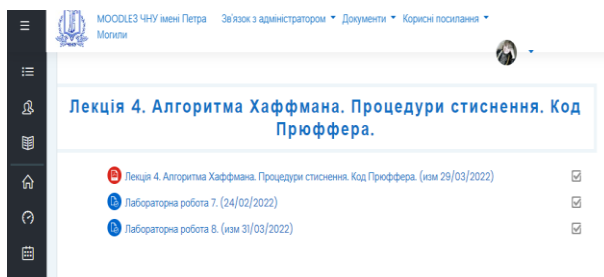
Обставини склалися так, що ми були змушеними перейти на дистанційні технології. Цей формат працює в університеті злагоджено, навчальні заняття для студентів проводяться за допомогою відеоконференцій Google Meet, використовується система електронного навчання Moodle, яка наповнена необхідними навчальними матеріалами. Оскільки це не перший наш досвід проведення освітнього процесу онлайн, ми зуміли швидко адаптуватись. У розкладах занять ми зараз вказуємо не аудиторію, а посилання, за яким можуть приєднатись до занять не лише студенти та викладачі, а й співробітники деканатів та ректорату, з метою контролю за освітнім процесом.

На рисунку 1 наведено приклад вивчення алгоритму Прюфера на занятті з Дискретної математики.

Для зручності сприйняття розроблено візуалізацію алгоритмів, які вивчаються (рис. 2). Студенти мають змогу переглянути відео зі стислим поясненням матеріалу в зручний для них час, бо, на жаль, не скрізь надійний інтернет, час від часу в Миколаєві повітряні тривоги та обстріли, під час яких бажано знаходитись в захищеному місці та інші питання, такі як робота студентів у волонтерських організаціях.

Нині навчання в дистанційному режимі є для нас напрацьованим механізмом – у нас вже є база тестових завдань у системі Moodle, сформовані відеопрезентації, які необхідні студентам комп'ютерного

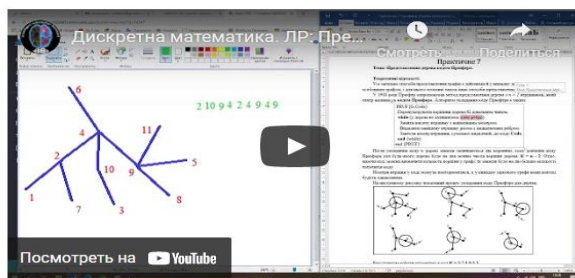
факультету для засвоєння курсу дискретної математики. Тому, звичайно, викладачам зараз працювати онлайн легше, ніж два роки тому. У студентів важкості у засвоєнні курсу ми не помітили ні при очному навчанні, ні при дистанційному. У будь-якій студентській підгрупі є студенти, які вчаться систематично й сумлінно та отримують за це високі бали, а є й ті, які можуть без поважної причини не відвідати заняття чи не підключитися до нього. Звісно, у перші дні воєнного стану відчувалося на заняттях високе емоційне напруження серед усіх нас, але зараз його рівень вже понизився. Напевно, ми вже звикаємо до цих обставин. Радує, що відповідальність студентів у нас досить висока, майже 90%.



Лабораторна робота 7. (24/02/2022)

Тема: Представлення дерева кодом Прюффера.

Мета: опрацювання алгоритму створення коду та антикоду Прюффера



Практичне 7 Прюффер 2022 з var32.docx | 5 лютий 2022, 10:45

Рисунок 1 – Приклади оформлення занять у Moodle

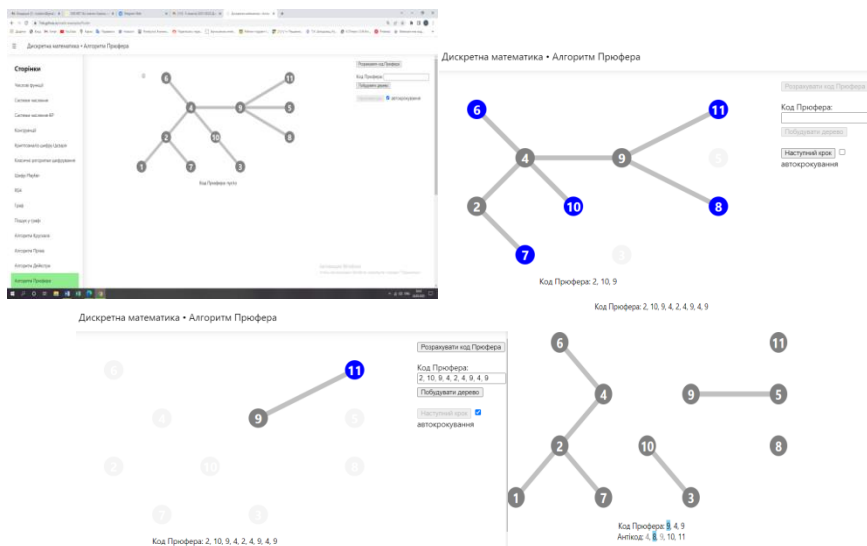


Рисунок 2 – Візуалізація роботи алгоритму Прюфера

УДК 519.65

Хомченко А. Н.,
 д-р фіз.-мат. наук, професор кафедри ІС,
 ЧНУ ім. П. Могили, Миколаїв, Україна

ІЗОСПЕКТРАЛЬНІ КЛАСИ СЕРЕНДИПОВИХ МОДЕЛЕЙ БІКУБІЧНИХ ЕЛЕМЕНТІВ

Мета роботи – класифікувати серендипові поверхні бікубічної інтерполяції функції двох аргументів з точністю до спектральної еквівалентності. Спектр – це сукупність 4-х чисел, що характеризують «у середньому» серендипову поверхню елемента.

Якщо спростувати обмеження Лагранжа на степінь інтерполяційного поліному $P(x, y)$, розв'язок задачі інтерполяції не буде однозначним. Але метод конструювання базису виділяє із множини можливих поліномів один (єдиний). Ситуація нагадує парадокс Бертрана (1889) з теорії ймовірностей. Цей парадокс не випадково називають парадоксом нечітких формулювань математичних задач. Йдеться про нечіткі початкові припущення.

Ми розглядаємо найбільш поширений на практиці елемент Q12 (рис. 1). На рис. 2 зображено відповідний (стандартний) фрагмент схеми Паскаля.

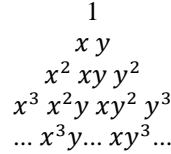
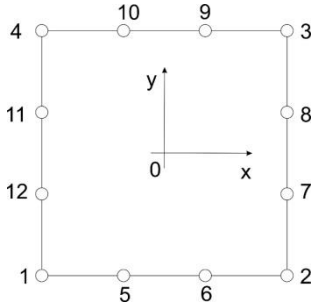


Рисунок 1 – Скінченний елемент Q12

Рисунок 2 – Схема Паскаля

Загальний вигляд інтерполянта:

$$P(x, y) = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 xy + \alpha_6 y^2 + \alpha_7 x^3 + \alpha_8 x^2 y + \alpha_9 xy^2 + \alpha_{10} y^3 + \alpha_{11} x^3 y + \alpha_{12} xy^3 \quad (1)$$

На практиці, як правило, працюють з «канонічним» носієм $-1 \leq x, y \leq 1$, а інтерполянт записують у формі Лагранжа: $P(x, y) = \sum_{i=1}^{12} N_i(x, y) \cdot f_i$, де f_i – відповідні аплікати.

Базисні функції $N_i(x, y)$ асоціюються з відповідними вузлами і мають властивості:

$$N_i(x_k, y_k) = \begin{cases} 1, & i = k, \\ 0, & i \neq k, \end{cases} \sum_{i=1}^{12} N_i(x, y) = 1 \quad (2)$$

де i – номер функції, k – номер вузла

Розв'язування задачі бікубичної інтерполяції – це пошук (моделювання) базисних функцій $N_i(x, y)$. Цілком достатньо знайти лише дві функції, наприклад, «кутову» $N_1(x, y)$ і «проміжну» $N_5(x, y)$. Поведінку серендипової поверхні досить інформативно і переконливо характеризують два числа: інтегральне середнє аплікати і локальна апліката в середній точці носія (в барицентрі). На моделях Q12 спектр містить 4 числа і утворює інваріантну правильність.

Важливе зауваження: ізоспектри з'являються лише після відмови від стандартного набору мономів (формула 1). Треба замінити лише два доданки: 11-й і 12-й. Теоретично, можливі два варіанти: $x^3 y^2, x^2 y^3$, або $x^2 y^2, x^3 y^3$, замість підкреслених у формулі (1). В англійських джерелах ці мономи 5-го і 6-го ступеня називають ієрархічними формами. Спроби замінити підкреслені мономи в (1) на ієрархічні форми були невдалими. Якщо визначник СЛАР обертається в нуль,

труднощі стають принциповими. Це стимулює пошуки нематричних методів конструювання моделей Q12. Деякі нематричні методи читач відшукає в монографії: Хомченко А. Н., Литвиненко О. І., Астіоненко І. О. Когнітивно-графічний аналіз ієрархічних базисів скінчених елементів. - Херсон: ОЛДІ-плюс, 2019. – 260 с.

Наведені нижче моделі з інваріантним спектром отримані переважно нематричними методами. Автор свідомо обрав стандартну модель Q12 (Standard), яка з 1968 року і до тепер (що дивує) вважається єдиною з неповторним спектром.

Стандартний спектр бікубічного елемента містить наступні 4 характеристики: локальна апліката «кутової» поверхні $N_1(0,0) = -\frac{5}{16}$, середня апліката «кутової» поверхні $\gamma_1 = -\frac{1}{8}$; локальна апліката «проміжної» поверхні в барицентрі елемента $N_5(0;0) = \frac{9}{32}$, середня апліката проміжної поверхні $\gamma_5 = \frac{3}{16}$.

Інтегральне усереднення аплікат виконується за формулою Ньютона-Котеса $\gamma_i = \frac{1}{S} \iint_D N_i(x,y) dx dy$, S – площа носія.

Нагадаємо, як виглядає стандартний базис Q12:

$$\begin{aligned} N_1(x,y) &= \frac{1}{32}(1-x)(1-y)(9(x^2+y^2)-10), \text{ аналогічно у} \\ &\text{вузлах } i=2,3,4. \\ N_5(x,y) &= \frac{9}{32}(1-x^2)(1-y)(1-3x), \text{ аналогічно для проміжних} \\ &\text{вузлів } i=6,7,8,\dots,12. \end{aligned} \quad (3)$$

Альтернативний базис зі стандартним спектром:

$$\begin{aligned} N_1(x,y) &= \frac{1}{128}(1-x)(1-y)(1-3x)(1-3y)(-40-39x- \\ &\quad 39y-36xy), \\ &\text{аналогічно для кутів } i=2,3,4; \\ N_5(x,y) &= \frac{9}{128}(1-x^2)(1-3x)(1-y)(1-3y)(4+3y), \\ &\text{аналогічно для } i=\overline{6,12}. \end{aligned} \quad (4)$$

Привертає увагу поява ієрархічних форм. Інваріантність спектра дозволяє комбінувати стандартні поверхні з альтернативними. Насправді, ми вже маємо 4 моделі. Зрозуміло, що зважене усереднення будь-якої пари базисів генерує безліч моделей з однаковим спектром.

Секція ТЕХНІЧНІ НАУКИ

**Підсекція: МОДЕЛІ, МЕТОДИ ТА ЗАСОБИ
ПРОГРАМНОЇ ІНЖЕНЕРІЇ**

УДК 004.032.26

Антіпова К. О.,
доктор філософії, викладач кафедри ІПЗ,
ЧНУ ім. Петра Могили, Миколаїв, Україна

ОГЛЯД НАЙБІЛЬШ ЗАГАЛЬНОВЖИВАНИХ АРХІТЕКТУР ГРАФОВИХ НЕЙРОННИХ МЕРЕЖ

У системах штучного інтелекту, які використовують глибоке навчання, набори даних зазвичай представлені в евклідовому просторі. Проте останнім часом збільшується кількість неевклідових даних, які представлені у вигляді графів. Графові нейронні мережі (ГНН) базуються саме на застосуванні графів у методах глибокого навчання. Під ГНН зазвичай мається на увазі не одна архітектура, а безліч різноманітних архітектур, які й були розглянуті.

В основі більшості архітектур ГНН лежить згортка графів, які складаються з набору вузлів і ребер. І вузли, і ребра можуть мати певні властивості, представлені векторами: h_i - вектор властивостей вузла, де i - індекс вузла, e_{ij} - вектор властивостей ребра, де i, j - вузли, які пов'язані цим ребром. Згортка використовує функцію властивостей суміжних вузлів для прогнозування властивостей вузла в наступному шарі нейронної мережі. Вектори властивостей використовуються для класифікації вузлів, ребер та графів.

Якщо застосувати функцію f до кожного з векторів h_i , то можна робити прогнози для кожного вузла. Тобто таким чином можна класифікувати вузли за їх ознаками: $Z_i = f(h_i)$. Аналогічно можна виконувати класифікацію ребер на основі їх властивостей. Для цього потрібні як вектори суміжних вузлів, так і вектори ребер: $Z_{ij} = f(h_i, h_j, e_{ij})$. Можна також прогнозувати деякі атрибути для всього графу, об'єднавши всі функції вузлів та застосувавши відповідну функцію f : $Z_G = f(\sum_i h_i)$.

Спектральні методи використовуються для представлення графу в спектральній області. Ці методи базуються на обробці сигналів графу і визначають оператор згортки в спектральній області за допомогою перетворення Фур'є. Спочатку сигнал x перетворюється в спектральну область, потім виконується операція згортки шляхом поелементного множення. Після згортки до отриманого сигналу застосовується зворотнє перетворення Фур'є:

$$\begin{aligned} F(x) &= U^T x, \\ F^{-1}(x) &= Ux, \\ g \cdot x &= F^{-1}(F(g) \cdot F(x)) = U(U^T g \cdot U^T x), \end{aligned} \quad (1)$$

де U – матриця, що визначається власними векторами L , де L – нормалізована матриця Лапласа, яка визначається наступним чином:

$$L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}, \quad (2)$$

де $U^T g$ – фільтр у спектральній області, D – степенева матриця, A – матриця суміжності графу.

Спектральні нейронні мережі в якості фільтру в спектральній області використовують діагональну матрицю g_w , де w – навчальні параметри нейронної мережі. Отже, можна побудувати мережу, яка вивчає згорткові фільтри для класифікації графів. Недоліками при цьому є те, що фільтр застосовується до всього графу, і, крім того, цей підхід є малоефективним в обчислювальному плані, особливо для великих графів.

Нейронна мережа ChebNet включає в себе визначення згорткових нейронних мереж в контексті теорії спектральних графів, що забезпечує необхідну математичну базу для проектування швидких локалізованих згорткових фільтрів на графах. Щоб вирішити проблему локальності, в ChebNets запропоновано, щоб на представлення властивостей у векторі деякого вузла v могли впливати лише вектори вузлів, які віддалені від вузла v на радіус, який менше або дорівнює k .

Ця згортка обчислюється за допомогою поліномів Чебишева. Використовуючи розширення поля K Чебишева, можна визначити K -локалізовану згортку, яка використовується для формування згорткової нейронної мережі. Таким чином зменшується складність обчислень, оскільки не потрібно обчислювати власні вектори нормалізованої матриці Лапласа.

Графові згорткові мережі вважаються найбільш розповсюдженою архітектурою у системах з глибоким навчанням. У цій мережі K –

локалізована згортка, запропонована в ChebNets, спрощена до $K = 1$. Також ця нейронна мережа відрізняється від інших спектральних наступним чином:

1. Використання одиничної матриці I для матриці суміжності A : $\tilde{A} = A + I$.

2. Використання симетричної нормалізації для матриці Лапласа:

$$L_{norm} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}.$$

3. Використання перенормування для вирішення проблеми зникання/вибуху градієнта:

$$I + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \rightarrow \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}, \quad (3)$$

де $\tilde{D}ij$ – ступенева матриця графу: $\tilde{D}ij = \sum_j \tilde{A}ij$.

Правило оновлення шару нейронної мережі розраховується наступним чином:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}), \quad (4)$$

де H – матриця властивостей, W – зважена матриця властивостей.

Відповідно, правило оновлення вузла розраховується наступним чином:

$$h_i^{(l)} = \sigma\left(\sum_{i \in N_j} c_{ij} W h_j\right), \quad (5)$$

де $c_{ij} = \frac{1}{\sqrt{|N_i| |N_j|}}$, N_i, N_j – кількість суміжних вузлів.

Графові згорткові мережі простіші в кодуванні і набагато ефективніші в обчислювальному відношенні, ніж мережі, що використовувалися до них. Проте такі мережі мають декілька обмежень: в них не підтримуються властивості ребер і повідомлення у графах. Зазвичай, вузли повинні мати можливість надсилати повідомлення (числові вектори) через ребра графа.

Просторові методи виконують згортки безпосередньо на графі на основі його топології. Зазвичай, алгоритм наступний:

1. Вектори властивостей вузла трансформуються за допомогою певної проекції.

2. Потім вони агрегуються за допомогою функції інваріантної перестановки.

3. Вектори властивостей розраховуються на основі поточних значень кожного вузла і агрегованого представлення його суміжних вузлів.

Нейронні мережі передачі повідомлень використовують поняття повідомлень у ГНН. Повідомлення m_{ij} відправляється через ребра i та j і обчислюється за допомогою функції f_e :

$$m_{ij} = f_e(h_i, h_j, e_{ij}). \quad (6)$$

Повідомлення, що надходять на кожен вузол, агрегуються за допомогою функції інваріантної перестановки. Для розрахунку оновлених векторів властивостей вузлів агреговані значення поєднуються з властивостями вузлів через функцію f_v :

$$h_i = f_v(h_i, \sum_{j \in N_i} m_{ji}). \quad (7)$$

Нейронні мережі передачі повідомлень є потужним фреймворком і вважаються однією з найбільш узагальнених архітектур ГНН. Однак їх слабким місцем є масштабованість через необхідність зберігання та обробки повідомлень, а також властивостей вузлів. Тому на практиці цю мережу застосовують лише для невеликих графів.

Графові мережі уваги використовують модель, яка у своєму найзагальнішому формулюванні дозволяє кожному вузлу брати до уваги всі інші вузли, відкидаючи всю інформацію про структуру графу. Розрахунок коефіцієнтів уваги використовується для визначення важливості властивостей вузла j для вузла i .

Основна ідея графових мереж уваги полягає в тому, щоб обчислити коефіцієнт уваги неявно, на відміну від графових згорткових мереж. Автори цієї нейронної мережі запропонували обчислювати коефіцієнт уваги a_{ij} на основі властивостей вузлів та ребер, які потім передаються у функцію уваги: $a_{ij} = \text{attention}(h_i, h_j)$. Після чого на вагових коефіцієнтах уваги a_{ij} застосовується функція softmax, щоб отримати розподіл ймовірностей:

$$a_{ij} = \text{softmax}_j(a_{ij}) = \frac{\exp(a_{ij})}{\sum_{k \in N_i} \exp(a_{ik})}. \quad (8)$$

Відповідно, правило оновлення вузла розраховується наступним чином:

$$h_i^{(l)} = \sigma(\sum_{i \in N_j} a_{ij} W h_j). \quad (9)$$

Отже, графові мережі уваги є достатньо масштабованими та ефективними з точки зору обчислень. Окрім того, вибір функції уваги не має великого значення: достатньо використовувати адитивну функцію оцінки.

Одним із основних недоліків більшості архітектур ГНН є масштабованість через те, що вектор властивостей кожного вузла формується на основі властивостей суміжних вузлів. Це може бути досить неефективно для величезних графів, де у кожного вузла багато суміжних вузлів. Для вирішення цієї проблеми використовуються методи вибірки. **Основна ідея методів вибірки** полягає в тому, що замість використання інформації про всі суміжні вузли, вибирається їх підмножина для проведення поширення. Ця ідея була реалізована у фреймворку **GraphSage** з наступною структурою:

1. Рівномірний вибір підмножини вузлів, суміжних з поточним. На кожному шарі нейронної мережі збільшується радіус суміжності k .
2. Агрегація властивостей вузлів із вибірки.
3. Класифікація графів або вузлів на основі агрегації, проведеної на попередньому етапі.

З точки зору складності обчислень цей підхід більш ефективний порівняно з використанням інформації від всіх суміжних вузлів. Модель може навчатися двома способами:

1. Навчання без учителя. Це можна зробити за допомогою функції втрат, яка змушує суміжні вузли мати подібні представлення, а незв'язані вузли - різні представлення.
2. Навчання з учителем. Це можна зробити за допомогою перехресної ентропії для визначення представлення вузлів.

PinSAGE є розширенням фреймворку GraphSAGE. PinSAGE базується на такому ж самому принципі, але застосовується на дуже великому графі (3 мільярди вузлів і 18 мільярдів ребер). Фреймворк використовується в системі рекомендацій Pinterest. Основні принципи цієї архітектури:

1. Для поточного вузла суміжні вузли визначаються за допомогою механізму випадкових блукань. Симулюючи випадкові блукання, які починаються з цільових вузлів, можна вибрати вузли з найбільшою кількістю відвідувань. також кожному вузлу буде присвоєна оцінка важливості, яка вказує, наскільки він важливий для поточного вузла.
2. Агрегація виконується за допомогою «вибірки важливості»: нормалізуються та підсумовуються оцінки важливості, отримані за допомогою випадкових блукань.
3. Модель навчається з учителем на вибірці вузлів, зв'язаних на основі даних з історії користувачів Pinterest.

Отже, можна зробити висновок, що в сучасні ГНН здебільшого є ефективними в обчислювальному плані та більш-менш простими з точки зору кодування. Слабка масштабованість все ще є недоліком деяких мереж, таких як спектральні нейронні мережі та нейронні

мережі повідомлень. У зв'язку з чим, активно використовуються такі фреймворки, як GraphSAGE та PinSAGE, які застосовують певні механізми вибору даних для проведення поширення.

УДК 004.4'23

Боровльова С. Ю.,
старший викладач канд. техн. наук, доцент кафедри ІПЗ,
Жлуктарьов А. А.,
студент факультету комп'ютерних наук,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

ПРИНЦИПИ ВИКОРИСТАННЯ CLEAN ARCHITECTURE

Для створення простих для розуміння та масштабованих інформаційних систем бажано обрати та дотримуватись певної архітектурної стратегії, налаштованої на потреби конкретного проекту. Використання архітектурних принципів дозволяє визначити такі положення для всієї команди розробників:

- які концептуальні шари наявні в застосунку та за що вони відповідають;
- як ці шари взаємодіють між собою та їх залежності;
- які зовнішні сутності передбачені та яким чином вони взаємодіють із застосунком.

Одним з таких архітектурних стилів є Clean Architecture (рис. 1), який був створений в 2010-х роках та активно використовується і сьогодні. Він поєднав в собі декілька інших підходів, які сходяться у таких положеннях:

- Архітектура має бути незалежною від UI частини проекту: вона може бути як консольним інтерфейсом, так й графічним, чи навіть просто «спілкуватись» через REST/SOAP API.
- Архітектура має бути незалежною від обраного інструментарію для інфраструктури: структура застосунку має передбачати зміну обраної бази даних, поштового серверу тощо.
- Має бути можливість тестувати кожен компонент програми легко та з мінімальною кількістю об'єктів-заглушок.

Такі вимоги досягаються завдяки абстракції класів у інтерфейси і використання їх у якості контрактів та дотримання принципу інверсії залежності: оскільки шари не залежать від конкретних реалізацій контрактів, програміст може просто замінити одну реалізацію іншою чи створити нову дотримуючись спільного інтерфейсу.

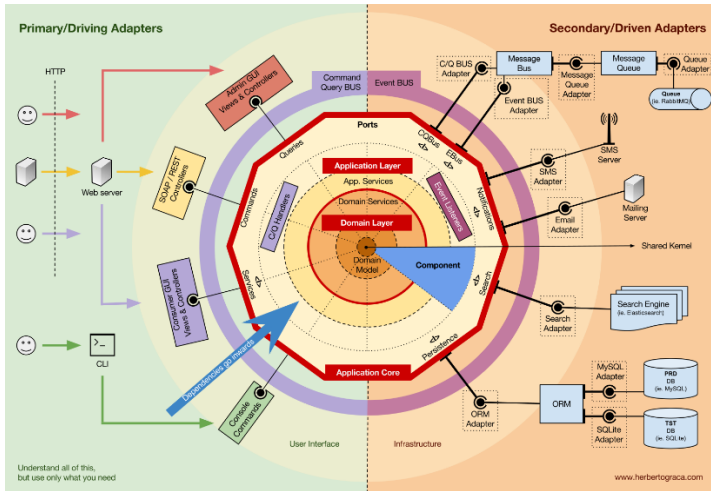


Рисунок 1 – Схема роботи застосунку при використанні Clean Architecture

Є декілька назв запропонованих шарів, але вони всі описують одні й ті ж сутності. Найбільш популярним розподілом є поділ на 4 частини:

- *Домен*. У ньому зберігаються сутності прикладного домену, які описують моделі, створені для конкретного рішення. Також сюди можуть додаватись Доменні Сервіси, якщо функціонал обробки моделі доволі складний та має бути окремо протестований.

- *Застосунок*. Цей шар втілює сценарії використання застосунку, використовуючи Домен та Інфраструктуру. Він також містить Порти – контракти, через які «спілкуються» шари Користувацького Інтерфейсу й Інфраструктури з Застосунком, та використовуються цим шаром для реєстрації залежностей.

- *Користувацький Інтерфейс*. Описує, яким чином користувач може взаємодіяти з програмою: через графічний чи консольний інтерфейс або за допомогою REST/SOAP.

- *Інфраструктура*. Категорія Інфраструктура містить посилання на конкретні реалізації взаємодії із зовнішніми сервісами: будь-то черга повідомлень чи ORM. Оскільки інтерфейси для використання конкретних реалізацій не завжди збігаються з тим, що зручно використовувати в контексті Застосунку, то часто їх абстрагують за допомогою шаблону проектування Адаптер. Він дозволяє поєднати

Порт та інфраструктурний сервіс завдяки імплементації потрібного контракту функціями обраної бібліотеки чи фреймворку.

Таким чином, потік залежностей направлений всередину, тому моделі та сценарії використання нічого не знають про оточуючі їх шари окрім їх контрактів, і тому можна, наприклад, змінювати користувацький інтерфейс та обрану базу даних. Кожен шар відокремлений та містить в собі все потрібне для тестування свого функціоналу.

УДК 004.4

Горбань Г. В.,
канд. техн. наук, доцент кафедри ІПЗ,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

ВИКОРИСТАННЯ ЗАСОБІВ МОВИ PYTHON ДЛЯ АНАЛІЗУ ДАНИХ РЕЗУЛЬТАТІВ ТРЕНУВАНЬ З АКАДЕМІЧНОГО ВЕСЛУВАННЯ

Штучний інтелект, у свою чергу, походить від наслідування людської поведінки та здібностей, таких як мислення та навчання. Він заснований на ідеї розробки так званих інтелектуальних агентів або машин із здатністю здобувати, моделювати та використовувати знання, аналітичні навички та досвід для досягнення спільних цілей. Розробка та впровадження інноваційних систем, заснованих на новітніх інформаційних технологіях і складних методах обробки даних, стають все більш важливими для негайного збору, передачі, зберігання та аналізу даних датчиків у спорті. Штучний інтелект заснований на концепціях машинного навчання та Data Mining. Інтеграція машинного інтелекту при розробці сучасних спортивних інформаційних систем дозволяє швидко й автоматично оцінювати значення параметрів спортивних даних.

Сучасні методи штучного інтелекту для спорту включають кластерний аналіз, пошук асоціативних правил, нейронні мережі, опорні вектори, нечітку логіку, які можна використовувати для кластеризації, класифікації, розпізнавання образів і прогнозування конкретних спортивних даних. Зокрема, аналіз даних з використанням алгоритмів самонавчання, таких як штучні нейронні мережі, все частіше обговорюється як перспективна галузь застосування у спортивній науці. Комп'ютерні системи з концепціями нечіткої логіки, які використо-

вуються в спорті, включають дані, зібрані з пристроїв із датчиками, а також рекомендації та стандарти для відповідних вправ.

Зарубіжні фахівці використовують елементи штучного інтелекту для вирішення завдань спортивної біомеханіки, спортивної орієнтації та відбору в системі аналізу для оптимізації тренувального процесу. Крім чисельних і статистичних методів, методи нейронних мереж, інтелектуальний аналіз даних, нечітка логіка та розпізнавання образів виявилися перспективними методами для оцінки та отримання інформації про спорт.

Метою дослідження є підвищення ефективності системи фізичного виховання в ЧНУ імені Петра Могили шляхом побудови та використання у навчальному процесі з фізичного виховання системи обробки даних фізичної підготовленості, розвитку та функціональних вимог студентів з подальшим аналізом даних, що дозволить виявити приховані закономірності та взаємозв'язки.

Для поточного дослідження було проведено тестування студентів ЧНУ імені Петра Могили з академічного веслування на тренажері Concept 2. У цьому тренажері для фіксації та зберігання результатів використовується монітор PM5, що дає можливість накопичення та збереження даних за певний період часу. Існує можливість обміну даними між монітором PM5 та персональним комп'ютером, відповідно до якого дані зберігаються в форматі CSV.

У результаті було отримано набір вхідних даних, одна частина ознак якого представлена даними з протоколу тестування студентів, а інша була отримана шляхом імпорту даних з моніторів pm5 веслувальних тренажерів Concept2. Після проведення очистки зайвих даних тренувань з тренажерів (тестування проводилось у певні дати протягом трьох днів, а монітор pm5 має можливість зберігати дані протягом одного року, тому потрібно було залишити тільки дані за певні дати) була проведена звірка імпортованих з монітору даних за часом початку тесту з даними протоколів, у результаті чого було знайдено кому саме зі студентів належать відповідні дані окремого тестування.

У табл. 1 представлено основні характеристики набору даних.

Таблиця 1 – Прибутковість від вкладення коштів за проектами

Назва	Опис	Джерело отримання
ID	ID студента	Протокол тестування
Name	ПІБ студента	Протокол тестування
Sex	Стать (варіанти значень: 'м', 'ж')	Протокол тестування
Nationality	Національність (варіанти значень: 'Ukrainian', 'Indian')	Протокол тестування
Age	Вік студента	Протокол тестування

Height	Зріст, вимірний перед проведенням тестування	Протокол тестування
Weight	Маса, виміряна перед проведенням тестування	Протокол тестування
HeartRateBefore	Пульс, вимірний перед виконанням тесту	Протокол тестування
HeartRateAfter	Пульс, вимірний після тесту	Протокол тестування
Meters	Загальна відстань у метрах, пройдена за весь час виконання тесту	Монітор тренажера
Avg SPM	Середня кількість помахів весла при виконанні тесту	Монітор тренажера
/500m	Час на проходження 500 м за загальною відстанню	Монітор тренажера
Cal/hr	Потужність помаху весла в калоріях у годину (0.001163 Вт) за загальною відстанню	Монітор тренажера
Watt	Потужність помаху весла у ваттах (Вт) за загальною відстанню	Монітор тренажера

Окрім п'яти показників, представлених у кінці таблиці, з монітора було отримано такі ж показники за певний проміжний час проходження тесту з академічного веслування: за перші 2 хвилини, за середні 2 хвилини (3–4 хвилини) та за останні 2 хвилини (5–6 хвилини).

З вихідного набору проведено первинний аналіз вихідних даних за допомогою мови Python з використанням бібліотеки Pandas. Дані з вхідного набору було експортовано у структуру DataFrame. В аналізі даних достатньо важливим є виявлення залежностей між ознаками. За допомогою бібліотеки Seaborn побудовано матрицю кореляції між кількісними ознаками віку, маси, зросту, пульсом перед та після виконання тесту та кількостями пройденої відстані у певний проміжок часу та загалом (рис. 1).

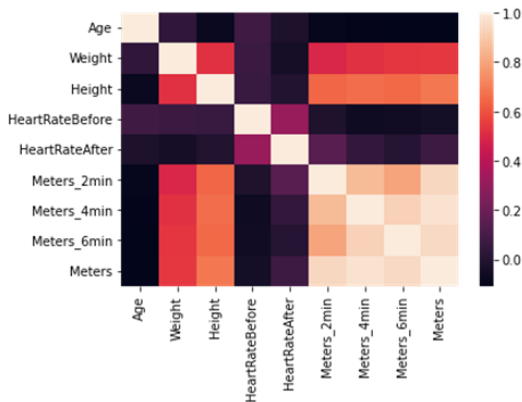


Рисунок 1 – Кореляційна матриця залежності кількісних ознак

Як видно з рисунку, вік майже не має залежності з іншими змінними, існує слабка залежність між ним та масою тіла, а також з величинами пульсів до та після виконання тесту. Втім, величини пройдених відстаней за певний період часу та загальної відстані між собою мають високу кореляцію.

У подальшому планується більш глибокий аналіз із використанням методів машинного навчання та штучного інтелекту, а також реалізація за допомогою фреймворку Django вебзастосунку для фіксації даних спортивних тренувань та тестів замість ведення паперових протоколів викладачів із фізичного виховання. У подальшому система на основі глибокого аналізу даних зможе зробити прогнози наступних результатів та підібрати необхідні режими тренування.

УДК 004.457

Гроза О. А.,
студент факультету комп'ютерних наук,
Кіяшко М. С.,
студент факультету комп'ютерних наук,
Боровльова С. Ю.,
старший викладач кафедри ІПЗ,
Давиденко Є. О.,
канд. техн. наук, доцент, завідувач кафедри ІПЗ,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ ЗНАНЬ

У зв'язку з останніми світовими подіями: пандемією COVID-19 та російським вторгненням на територію України – онлайн-навчання набуло значного поширення та безпрецедентної актуальності. Програмні засоби, що уможливають дистанційну освіту стали невід'ємною та критично важливою частиною навчального процесу.

Одним із напрямків забезпечення можливості віддаленого навчання є розробка автоматизованих систем проведення тестування. На даний момент існує багато аналогічних систем, проте залишається попит на створення нових, адже постійно з'являються нові технології, що дозволяють зробити процес тестування знань ще більш універсальним та зручним.

Тестування є ефективним засобом підсумкового контролю знань, що зводить до мінімуму вплив людського фактору та можливої упередженості на оцінку учня. Проведення поточного контролю у формі тестування дозволяє виявити прогалини в знаннях та надає викладачеві цінну зворотну інформацію, на основі якої можна скорегувати план занять для максимального засвоєння учнями матеріалу. Також тестування є чудовим засобом безпосередньо навчання: давно відомий психологам «ефект тестування» полягає на тому, що відтворення інформації з пам'яті при проходженні тесту допомагає значно краще запам'ятати новий матеріал, ніж такі розповсюджені способи, як перечитування чи створення конспекту. Отже, тестування є цінним інструментом, що може використовуватись протягом усього навчального процесу.

Традиційний спосіб проведення тестування є досить трудомістким: він передбачає друк матеріалів, ручну перевірку та обрахунок статистики. Водночас він не гарантує достовірності результатів, адже можуть трапитися помилки під час ручної обробки даних. Автоматизація тестування значено пришвидшує цей процес, а отже звільняє час вчителів. Також усувається фактор людської помилки при перевірці. Програмні системи, що роблять процес тестування простішим, сприяють більш широкому використанню цього корисного елемента навчального процесу у закладах освіти. Окрім того, програмні засоби можуть надавати додаткові можливості, такі як легка інтеграція в тест мультимедійних елементів, контроль за часом проходження, візуалізація статистики тощо.

Мета розробки полягає у підвищенні ефективності перевірки знань здобувачів освіти за рахунок зменшення кількості помилок при перевірці тестів, підвищення швидкості та точності підрахунку результатів. Застосунок призначений насамперед для використання у закладах освіти, однак система може бути використана і для створення тестів на довільну цікаву для користувачів тему.

Серед наявних на ринку систем автоматизації тестування був виявлений ряд недоліків, які були виправлені у розробленій системі. Позитивні аспекти аналогічних систем узяті за приклад під час розробки застосунку.

Перевагами системи порівняно з аналогічними є:

- можливість планування часу проходження учнем тесту;
- рейтинг найпопулярніших тестів та найуспішніших користувачів за кількістю пройдених тестів та оцінкою;
- відсутність жорсткого поділу на ролі «учень» та «вчитель»: кожен користувач може як створювати, так і проходити тести;
- сучасний та легкий у використанні інтерфейс;

– наявність Telegram-боту як додаткового засобу взаємодії із застосунком.

Програмне забезпечення умовно можна поділити на дві частини: функціонал для конструювання тестів та їх проходження. Форму створення тесту представлено на рис. 1.

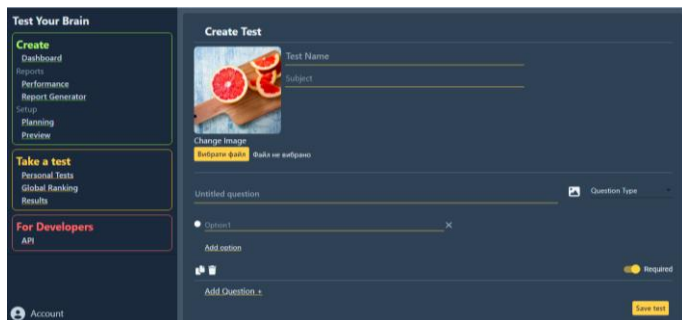


Рисунок 1 – Сторінка створення тесту

Після створення тесту можливий його попередній перегляд без збереження результатів. Це дозволяє розробнику тесту оцінити зручність проходження, виявити та виправити можливі помилки при внесенні даних. Зміст тесту можна оновити завдяки режиму редагування. Розробник тесту має можливість контролювати час, коли тест доступний для проходження, визначати його статус як приватний (доступний лише визначеним особам) або загальнодоступний. Користувач також отримує інформацію про кількість проходжень кожного створеного ним тесту та середній бал. На сторінці статистики можна переглянути динаміку зміни результатів за обраний період. Рейтинг найпопулярніших тестів заохочує до розробки цікавих та якісних тестів (рис. 2).

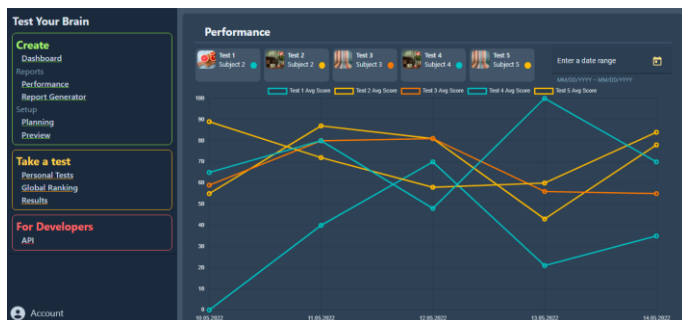


Рисунок 2 – Сторінка статистики результатів тестування

Кожен користувач має можливість шукати загальнодоступні тести за тематикою або назвою та проходити їх. На окремій сторінці висвітлюються тести, доступ до яких наданий користувачеві особисто. Користувач може в будь-який час переглянути свої результати. Рейтинг успішності користувачів заохочує до проходження тестів.

Оскільки Telegram є одним з найбільш популярних месенджерів в Україні, створено бот, за допомогою якого можна швидко переглянути статистику проходження та результати тестів.

Розроблений застосунок має клієнт-серверну архітектуру. Backend написано на мові C# із використанням паттерна проектування Loosely Coupled Monolith. Сутність паттерна полягає у розбитті системи на різні проекти, що забезпечує зручність при розробці, налагодженні, тестуванні системи в цілому. Frontend застосунку створено на базі фреймворку Angular. Важливий функціонал застосунка – динамічне створення та зміна форм з декількома рівнями вкладеності – реалізується за допомогою технології Reactive Forms. Стилізація виконується засобами фреймворку стилів Tailwind CSS. Завдяки гнучкому функціональному підходу до створення стилів дана технологія дозволяє легко досягти естетичного та нестандартного дизайну інтерфейсу.

Отже, створена система використовує сучасні технології для вдосконалення процесу проведення тестування знань.

УДК 004.652.4:004.424

Дворецький М. Л.,
канд. техн. наук, доцент б.в.з. кафедри ІПЗ,
Дворецька С. В.,
старший викладач кафедри ІПЗ,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

ШЛЯХИ РЕАЛІЗАЦІЇ НАСЛІДУВАННЯ ПРИ РОБОТІ ІЗ ОБ'ЄКТНОЮ МОДЕЛЛЮ ІЗ ВИКОРИСТАННЯМ РЕЛЯЦІЙНИХ БАЗ ДАНИХ

На початку необхідність використання баз даних була обумовлена потребою реалізації так званих «інформаційних» систем. Інформаційна система (ІС) – взаємозалежна сукупність засобів, методів і персоналу, використаних для зберігання, обробки й представлення інформації. Відповідно при розробці такого роду систем первинною виступала

база даних та її структура, під яку створювалось клієнтське програмне забезпечення.

Але поступово в результаті еволюції програмного забезпечення з'являються триланкові застосунки, що стають передумовою для створення та розвитку вебзастосунків, технологій односторінкових вебзастосунків (SPA) та прогресивних вебзастосунків (PWA). Також цьому сприяє перехід від персональних комп'ютерів до мобільних пристроїв, що у свою чергу ініціює зміщення популярності від десктопних застосунків до мобільних на вебзастосунки.

На даному етапі, метою використання бази даних у такого роду застосунках стає також збереження їх стану для подальшого коректного продовження роботи із ними. Крім того, враховуючи реалізацію більшої частини логіки на сервері застосунків (а не на серверні БД), первинним стає реалізація саме цієї частини програмного коду, а структура БД стає вторинною, тобто буде вже залежати від особливостей цієї реалізації.

Останні десятиліття серед основних парадигм розробки програмного коду найбільшої популярності набув об'єктно-орієнтований підхід. У рамках об'єктно-орієнтованого підходу система розбивається на набір об'єктів, відповідних об'єктам реального світу, які взаємодіють між собою шляхом посилки повідомлень. Іншою особливістю є об'єднання в об'єкті як атрибутивних даних (характеристики, властивості), так і поведінки (функції, методи). Поведінка об'єкту закладається на сервері застосунків та вимагає збереження стану, оскільки її особливості залежать лише від властивостей об'єкту або вхідних параметрів. Отже, щоб зберегти стан об'єкту, достатньо виконати збереження набору його властивостей.

Найбільш популярною моделлю збереження даних у БД була (і поки що залишається) реляційна модель. Була спроба створення об'єктної моделі, що завершилась не зовсім вдало. Також на даний момент паралельно із реляційною моделлю потужно розвивається напрямок постреляційних NoSQL баз даних, але він не може замінити реляційну модель, оскільки кожен із цих двох підходів є більш доцільним для своєї групи задач.

Саме тому, для адаптації роботи об'єктно-орієнтованого підходу під час реалізації логіки програмного забезпечення з боку сервера застосунків до реляційної моделі даних, що використовується для збереження стану набору об'єктів (що характеризує стан застосунку в цілому), було створено технологію object-relational mapping (ORM).

ORM (Об'єктно-реляційна проєкція) – технологія програмування, яка зв'язує реляційні бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи «віртуальну об'єктну базу даних». Суть

проблеми полягає в перетворенні об'єктів у форму, в якій вони можуть бути збережені у базах даних, і які легко можуть бути витягнуті в подальшому, зі збереженням властивостей об'єктів і відношень між ними.

Застосунки, що використовують ORM, зазвичай працюватимуть повільніше і використовуватимуть більше пам'яті, ніж програми, написані «вручну». Але ORM позбавляє програміста від написання великої кількості коду, часто одноманітного і схильного до помилок, тим самим значно підвищуючи швидкість розробки.

Наслідування – концепція об'єктно-орієнтованого програмування, згідно з якою тип даних може наслідувати дані та функціональність певного існуючого типу, сприяючи повторному використанню компонентів програмного забезпечення. В контексті наслідування класів в ООП використовується наступна термінологія. Суперклас, батьківський клас, предок, батько або надклас клас, який успадковується в підкласах, тобто клас, від якого успадковуються інші класи. Підклас, похідний клас, дочірній клас, клас нащадок, клас спадкоємець або клас-реалізатор – клас, успадкований від суперкласу, тобто клас, визначений через успадкування від іншого класу. Базовий клас - це клас, що знаходиться на вершині ієрархії наслідування класів і в основі дерева підкласів, тобто не є підкласом і не успадковує від інших класів.

Наслідування є механізмом повторного використання коду та сприяє незалежному розширенню програмного забезпечення. Як уже зазначалось, у загальному випадку підклас може наслідувати від суперкласу дані та функціональність. Говорячи про представлення (збереження) даних об'єктів класу на рівні БД, нас передусім цікавитиме наслідування даних або властивостей об'єктів.

Далі розглянемо декілька підходів щодо реалізації наслідування на рівні представлення об'єктів у БД із використанням ORM Doctrine. Перший підхід має назву «Однотабличне наслідування» (Single Table Inheritance) і полягає у представленні об'єктів всіх підкласів, що наслідують від базового класу, в одній таблиці БД.

Розглянемо даний підхід на прикладі реалізації базового класу «Person» із властивістю «name», та два класи «Student» та «Lecturer», що його розширюватимуть. Клас «Student» матиме додатково властивості «university», «specialty» та «course», а клас «Lecturer» - властивості «department», «academicDegree» та «academicTitle».

У таблиці представлено як властивості базового класу «Person», так і всі властивості похідних класів «Student» та «Lecturer». Також маємо додаткове поле, в якому зазначається, до якого класу відноситься той чи інший екземпляр (рядок таблиці). При чому для екземплярів класів

«Person» та «Lecturer» властивості класу «Student» («university», «specialty» та «course») матимуть значення «null» (рис. 1). Так само, для екземплярів класів «Person» та «Student» властивості класу «Lecturer» («department», «academicDegree» та «academicTitle») матимуть значення «null» (рис. 2).

id	name	discr	university	specialty	course	department	academicDegree	academicTitle
11	Kolya	person	NULL	NULL	NULL	NULL	NULL	NULL
17	Maria	person	NULL	NULL	NULL	NULL	NULL	NULL
27	Petro	lecturer	NULL	NULL	NULL	software engineering	канд.техн.наук	доцент

Рисунок 1 – Властивості «university», «specialty» та «course» для екземплярів класів «Person» та «Lecturer»

id	name	discr	university	specialty	course	department	academicDegree	academicTitle
11	Kolya	person	NULL	NULL	NULL	NULL	NULL	NULL
17	Maria	person	NULL	NULL	NULL	NULL	NULL	NULL
26	Piter	student	ЧНУ ім.П.Могили	інженерія програмного забезпечення	3	NULL	NULL	NULL

Рисунок 2 – Властивості «department», «academicDegree» та «academicTitle» для екземплярів класів «Person» та «Student»

Для реалізації даного підходу при використанні ORM Doctrine анотація класу «Person» набуває наступного вигляду (рис. 3).

```

9  /**
10 * @ORM\Entity
11 * @ORM\Table(name="persons")
12 * @ORM\InheritanceType("SINGLE_TABLE")
13 * @ORM\DiscriminatorColumn(name="discr", type="string")
14 * @ORM\DiscriminatorMap({"person" = "Person", "student" = "Student",
15 * "lecturer" = "Lecturer"})
16 */
17 class Person {

```

Рисунок 3 – Властивості «department», «academicDegree» та «academicTitle» для екземплярів класів «Person» та «Student»

Використання підходу «Single Table Inheritance» має ряд недоліків. Серед основних слід згадати велику кількість берегів у таблиці, що містять значення «null» у випадку великої кількості похідних класів, або великої кількості специфічних властивостей похідних класів. Відповідно, це призводить до невиправданого збільшення обсягів БД. Крім того, зосереджуючи всі дані в одній таблиці, матимемо ризик виникнення блокувань при багатокористувацькому режимі роботи. Також маємо необхідність у виконанні додаткової фільтрації при

виборі об'єктів певного типу, що у свою чергу також негативно впливає на продуктивність.

Іншим підходом щодо реалізації наслідування класів на рівні БД є наслідування із використанням зв'язаних таблиць. У цьому випадку властивості базової сутності знаходяться в одній таблиці, а властивості похідного класу – в іншій. У цьому випадку при зверненні до об'єкту підкласу виконуватиметься запит до двох таблиць із використанням оператора «join».

У нашому випадку для класів «Person», «Student» та «Lecturer» на рівні БД буде створено три таблиці: «persons» - для представлення властивостей базового класу для об'єктів всіх трьох класів; «student» - для специфічних властивостей похідного класу «Student»; і «lecturer» - для специфічних властивостей класу «Lecturer». У такий спосіб зникає надлишкова витрата дискового простору для представлення полів із значеннями «null», як це було у випадку використання однієї таблиці. Структура таблиць матиме наступного вигляду (рис. 4).

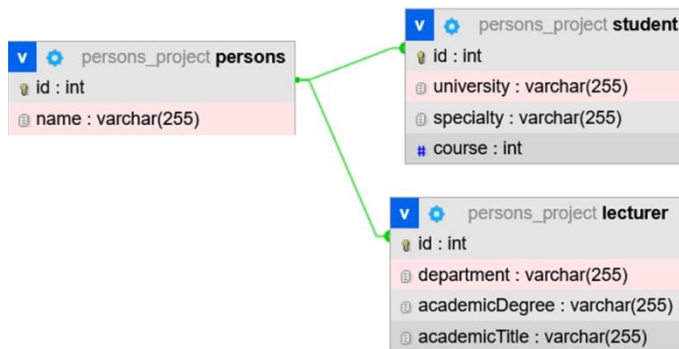


Рисунок 4 – Структура таблиць «persons», «student» та «lecturer»

В ORM Doctrine даний підхід має назву «Class Table Inheritance». Для його реалізації необхідно виконати зміни у анотаціях класу «Person», а саме змінити тип наслідування на «JOINED» (рис. 5).

```
9  /**
10 * @ORM\Entity
11 * @ORM\Table(name="persons")
12 * @ORM\InheritanceType("JOINED")
13 * @ORM\DiscriminatorColumn(name="discr", type="string")
14 * @ORM\DiscriminatorMap({"person" = "Person", "student" = "Student",
15 * "lecturer" = "Lecturer"})
16 */
17 class Person {
```

Рисунок 5 – Зміни в анотаціях класу «Person»

Перевагами розглянутого підходу реалізації наслідування є позбавлення від берегів із значенням «null», як це було у випадку із однією таблицею. Однак загальна частина даних (властивості базового класу) все ще залишаються в одній таблиці, що у свою чергу веде до підвищеної ймовірності виникнення блокувань на рівні БД. Крім того, виникає необхідність у використанні операції «JOIN» при вибірці даних об'єктів як базового так і похідних класів, що у свою чергу також негативно впливає на продуктивність роботи вибірки при великих обсягах таблиць.

Існує ще один підхід для реалізації наслідування, що полягає у представленні похідних класів у вигляді окремих, незалежних таблиць. У нашому випадку список полів таблиці «persons» має бути продубльовано у таблицях «student» та «lecturer».

Даний підхід мінімізує вірогідність виникнення блокувань, оскільки дані об'єктів різних класів представлені у незалежних таблицях. Крім того, для отримання даних об'єкту достатньо звернення до однієї таблиці, замість двох (або більше) із використанням оператора «JOIN». Головним недоліком даного підходу на перший погляд вбачається дублювання структури властивостей базового класу у декількох таблицях.

Але з іншого боку, якщо говорити про дані, то дублювання не буде, оскільки кожна частина даних об'єктів, що є інстанціями певного класу, зберігаються лише в одній таблиці. А часткове дублювання структури таблиць БД не буде мати значення у випадку генерації та модифікації (синхронізації) структури БД виключно з боку ORM Doctrine. Тобто у цьому випадку всі зміни структури виконуються у класах застосунку із подальшою синхронізацією із структурою БД. В іншому випадку (при прямому внесенні змін у структуру БД) виникає ймовірність невідповідності структури різних таблиць, що перебувають в ієрархії одного базового класу в об'єктній моделі застосунку.

В ORM Doctrine даний підхід може бути реалізовано за допомогою наслідування Mapped Superclasses. Mapped Superclass – це абстрактний або конкретний клас, який забезпечує представлення спільної частини властивостей для своїх підкласів, але сам по собі не є сутністю (не представлений на рівні таблиці БД). Мета такого суперкласу полягає у визначенні стану об'єкту відносно переліку властивостей, яка є спільною для кількох класів-сутностей.

На жаль, клас, що використовує анотацію Mapped Superclass, не може бути сутністю, і відповідно немає можливості виконувати запити безпосередньо до Mapped Superclass. Іншим суттєвим недоліком є те, що всі зв'язки із іншими класами (посилальні береги), визначені

відображеним суперкласом, повинні бути односпрямованими (лише зі стороною власника). Це означає, що асоціації «один до багатьох» взагалі неможливі для відображеного суперкласу. Крім того, зв'язки «багато-до-багатьох» можливі лише в тому випадку, якщо відображений суперклас на даний момент використовується лише в одній сутності.

Ще одна можливість реалізації наслідування, що присутня у мові РНР, полягає у використанні трейтів. Даний підхід дозволяє у нашому випадку реалізувати три окремих класи сутності без наслідування, як такого, а спільні властивості винести до окремого трейту.

УДК 004.6

Кірей К. О.,

канд. пед. наук, доцент, доцент кафедри ІПЗ,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

МОДИФІКАЦІЯ ПРОДУКТІВ БІЗНЕС-АНАЛІТИКИ ДЛЯ ОБРОБКИ ВЕЛИКИХ ДАНИХ

Традиційні продукти бізнес-аналітики спочатку не були призначені для обробки великих даних. Вони орієнтувалися на роботу з добре структурованими і зрозумілими даними, що зберігалися в сховищі реляційних даних і відображалися на екрані персонального комп'ютера. Великі дані є терміном, який з'явився у 2008 році та зараз використовується для ідентифікації наборів даних, з якими не можна впоратися з використанням традиційних методологій та програмних засобів через їхній великий розмір і складність. Традиційна ж бізнес-аналітика, зазвичай, застосовується до окремих вибірок даних, а не до всього доступного обсягу даних.

У доповіді розглянуто що має змінитися, коли методи бізнес-аналітики використовуються для аналізу великих даних. Модифікація методів бізнес-аналітики, по-перше, має стосуватися трьох складових: самих даних, аналітичних алгоритмів їхньої обробки та ІТ-інфраструктури. У доповіді розглянуто більш детально напрями трансформації кожної складової.

Бурхливий розвиток інформаційних технологій, зокрема, мережевих технологій накопичення та зберігання даних, поява нових електронних сервісів, сприяє значному зростанню потоків та обсягів даних. Це призвело до зміни парадигми накопичення даних. Самі дані стали цінним ресурсом, що можна застосовувати в різних галузях. Відбувся

перехід від парадигми накопичення даних – «ми знаємо, які дані нам потрібні і збираємо саме їх», до – «ми не знаємо, які дані нам потрібні, збираємо всі дані, що можливо, а потім з'ясуємо їхню цінність». Великі дані складаються зі структурованих, слабоструктурованих і неструктурованих даних. Обсяг цих даних може бути величезним, а самі дані – достатньо складними. Для того щоби великі дані були цінними вони мають задовольняти певним властивостям. У рамках кожної системи перелік провідних властивостей може варіюватися. Однак тут можна зазначити основні чинники, які слід мати на увазі:

- дані можуть надходити з джерел, що не заслуговують довіри. Аналіз великих даних часто включає агрегацію даних із різних джерел, як зовнішніх, так і внутрішніх. Проблема зазвичай полягає в неможливості перевірити надійність зовнішніх джерел інформації. Наприклад, достовірність інформації з повідомлень Twitter не може бути гарантованою, вона може бути заснована на неперевіреніх даних. Крім того, тут також необхідно враховувати цілісність даних;

- дані можуть бути некоректними. Некоректні дані – це неточні, неповні або помилкові дані: від тексту з орфографічними помилками до показань погано відкаліброваного датчика, а також дубльовані дані: досі немає єдиного підходу до того, як треба «очищати» дані: у межах певного інтервалу, або все в режимі реального часу. Є також думка, що некоректні дані взагалі не треба очищати, оскільки вони можуть містити цікаві аномалії. Стратегія очищення визначається джерелом, типом даних і метою аналізу. Наприклад, при розробленні фільтра проти спаму важливо визначити підозрілі елементи даних, тому «чистити» їх не треба;

- співвідношення «сигнал-шум» може бути низьким. Іншими словами, корисна інформація може становити лише малу частку наявних даних, а все інше буде «шумом». Перевага аналізу великих даних якраз у тому, що він здатний витягти з «гучних» даних дрібний сигнал – але цей сигнал може виявитися дійсно незначним;

- дані можуть надходити в режимі реального часу. У багатьох випадках доводиться аналізувати потоки даних у режимі реального часу.

Управління великими даними – важлива частина процесу аналітики. Для підвищення ефективності бізнес-аналітики потрібні більш досконалі рішення з управління даними, які забезпечать достовірність даних, взятих із нових джерел. Це особливо важливо, якщо такі дані поєднуються з наявними достовірними даними, які розміщені в сховищі. Також тут знадобиться удосконалити забезпечення безпеки та конфіденційності даних. Треба звернути увагу на те, що управління інформацією – це, по-перше, здатність створити інформаційний ресурс,

якому могли б довіряти співробітники, партнери та клієнти компанії, а також державні установи. Організація має бути обізнана про правила захисту інформації та управління даними, що застосовуються до різних джерел великих даних. Такими даними, наприклад, є дані про стан здоров'я, захищені законом про спадкоємність та підзвітність медичного страхування (Health Insurance Accountability and Portability Act, HIPAA), або персональні дані, такі як імена та адреси. Так в Україні діють чотири закони, щодо захисту інформації, це Закон України «Про захист інформації в інформаційно-комунікаційних системах», Закон України «Про захист персональних даних», Закон України «Про інформацію» та Закон України «Про державну таємницю». Окрім цього є низка постанов КМУ, нормативні документи в галузі технічного захисту інформації (НД ТЗІ) та державні стандарти України (ДСТУ) щодо створення і функціонування КСЗІ та галузеві стандарти щодо захисту інформації.

Якщо система управління даними не забезпечує їхній належний захист, то компанія порушить закон, працюючи з такими даними. Також треба звернути увагу не тільки на використання даних, а і їхнє видалення. Деяка частина зібраних даних може виявитися непотрібною або стане непотрібною через певний проміжок часу, і компанія має подбати про належне їхнє видалення. Решті даних необхідно забезпечити захист і управління. Отже, яку б інформацію ми не використовували, необхідна чітко визначена стратегія безпеки й погоджена з нею стратегія управління даними. Безпека й управління даними в сукупності мають гарантувати відповідальність усіх сторін, що беруть участь у впровадженні системи управління інформацією. Відповідальність за інформаційну безпеку так чи інакше несе вся організація загалом. Навіть впровадження всіх новітніх засобів безпеки не усуне наявні ризики, якщо кінцеві користувачі не будуть розуміти своєї ролі в захисті даних, з якими працюють.

Аналітичні алгоритми також потребують певних модифікацій. Коли аналітика виходить за межі одного комп'ютера – під час роботи з великими даними алгоритми можуть вимагати доопрацювання, однак такі внутрішні зміни не мають завдати шкоди зовнішнім функціональним можливостям. Нині вже є багато рішень пов'язаних із великими даними, серед яких можна звернути увагу на розробки таких компаній, як Google (Google AI (ai.google/research)), Amazon (Сервіси хмарних технологій (aws.amazon.com/ru/?nc2=h_lg)), IBM (www.ibm.com/analytics/products_cognitiveclass.ai/learn/big-data/), Oracle (Big Data, Big Data Appliance, Big Data Cloud Service, Big Data Connectors, Big Data Discovery, Big Data Platform, Big Data Preparation Cloud Service, Big Data SQL, Big Data Spatial and Graph

(www.oracle.com/ru/products/oracle-a-z.html#b), Microsoft (www.microsoft.com/en-us/sql-server/big-data) тощо. Перевага інфраструктури великих даних полягає в тому, що вона дає змогу за лічені хвилини обробити модель, обробка якої раніше займала години й навіть дні. З'являється можливість циклічно застосовувати модель сотні й сотні разів. Однак у разі регресії мільярдів рядків даних у розподіленому середовищі необхідно пам'ятати про вимоги щодо ресурсів, пов'язаних з обсягом даних і їхнім розташуванням у кластері. Алгоритми мають враховувати різні особливості оброблюваних даних.

Нині з'являються нові аналітичні можливості, які здатні обробляти великі дані безпосередньо в джерелах, не вимагаючи попереднього їхнього збереження. Такий підхід дає змогу зберігати тільки значущі дані, а також аналізувати інформацію на більш ранніх етапах, виявляючи ключові події завчасно, що дуже важливо для прийняття рішень у режимі реального часу.

Звичайно, аналітика буде розвиватися й надалі. Наприклад, для відображення постійно мінливих даних можуть знадобитися можливості візуалізації в режимі реального часу. Але як на практиці побудувати графік, що складається з мільярдів точок? Або, приміром, як без втрат у швидкості і якості аналізу застосовувати алгоритми прогнозування до постійно зростаючого набору складних даних? Це лише деякі перспективні напрями розробок у галузі аналітики.

Інфраструктура, необхідна для підтримки великих даних, докладно описана в багатьох дослідженнях. Зазначимо лише ключові вимоги щодо платформи для роботи з великими даними:

- Інтеграція технологій. Інфраструктура має інтегрувати нові технології великих даних із традиційними технологіями. Це дасть змогу обробляти всі види великих даних і робити їх придатними для традиційних методів аналізу.

- Зберігання великих обсягів розрізнених даних. Для цього можна застосовувати систему Hadoop. Hadoop – проєкт фонду Apache Software Foundation, вільно поширюваний набір утиліт, бібліотек і фреймворків для розробки та виконання розподілених програм, які працюють на кластерах із сотень і тисяч вузлів. Використовується для реалізації пошукових і тематичних механізмів багатьох високонавантажених вебсайтів. Розроблений на Java в рамках обчислювальної парадигми MapReduce, згідно з якою застосунок поділяється на велику кількість однакових елементарних завдань, виконуваних на вузлах кластера і природним чином зведених у кінцевий результат. Ця система здатна обробляти, зберігати й контролювати великі обсяги структурованих, слабоструктурованих і неструктурованих даних.

– Обробка переміщуваних даних. Датчики, інтелектуальні пристрої, аудіо- та відеотехніка, журнали систем постійно генерують великі потоки таких даних, які можуть бути використані в процесах забезпечення прийняття рішень у режимі реального часу.

– Дані в сховищі. Рішення, оптимізоване для операційних і пікових робочих навантажень, дасть змогу аналізувати і зберігати зростаючі обсяги достовірних даних, а також управляти цими даними.

І звичайно необхідна можливість інтеграції вже наявних даних із результатами аналізу великих даних.

Отже, рішення для бізнес-аналітики мають бути модифікованими з урахуванням великих обсягів даних і забезпечувати інтеграцію традиційної аналітики з аналітикою великих даних. Модифікація має враховувати особливості пов'язані із самими даними – це одержання даних, що відповідають певним властивостям. Процеси аналізу, збереження, захисту та видалення даних мають враховувати не тільки необхідність захисту корпоративної таємниці, а й чинне законодавство під яке підпадають певні дані. IT-інфраструктура має забезпечувати виконання всіх процесів пов'язаних із бізнес-аналітикою для обробки великих даних із дотриманням певних вимог.

УДК 004.42

Новожилів Я. В.,

студент факультету комп'ютерних наук,

Бойко А. П.,

канд. техн. наук, доцент кафедри КІ,

ЧНУ ім. Петра Могили, м. Миколаїв, Україна

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТРАНСПОРТНОЇ ЛОГІСТИКИ

У сучасному світі майже всі сфери суспільства в тій чи іншій мірі пов'язані з вантажоперевезеннями. Це імпорт і експорт, обмін вантажами між підприємствами, доставка вантажів в торгові точки, приватні перевезення та інше. При цьому транспортування різних видів вантажів супроводжується різноманітними логістичними рішеннями: вибір оптимального маршруту, виду транспорту, підготовка документів, розрахунок вартості, завантаження, вивантаження та інше.

В останні роки стрімко розвиваються логістичні компанії, які просувають свої послуги за допомогою цифрових ресурсів, у тому числі вебсайтів. Проаналізувавши ринок сучасних вебсервісів, можна

відмітити, майже відсутнє універсальне рішення для створення ланцюгу логістики в Україні. Існує лише декілька спеціалізованих сервісів, які мають обмежений функціонал та важкий для розуміння користувачів інтерфейс. Таким чином, розробка сучасного програмного забезпечення для організації вантажоперевезень представляється актуальною задачею, так як сфера логістики є одною з найперспективніших сфер в економіці. Вона охоплює великий спектр економічної діяльності за своїм функціональним призначенням.

Метою даної роботи є систематизація проведення логістичних операцій за допомогою створення спеціалізованого вебсайту.

Корпоративний спеціалізований вебсайт – це канал зв'язку між компанією і клієнтами, партнерами і відвідувачами. Цей канал працює цілодобово і доступний практично з будь-якої точки світу.

Власний сайт компанії дозволить:

1. Зменшити витрати на зовнішню та медійну рекламу.
2. Збільшити дохід шляхом залучення клієнтів з інтернету.
3. Представити актуальну інформацію клієнтам і партнерам.

В першу чергу власний сайт – це іміджевий інструмент для будь-якої компанії, тому він повинен мати привабливий дизайн і чітку архітектуру. З точки зору дизайну – обмежень в оформленні майже не існує, головна вимога – воно має відповідати фірмовому стилю компанії.

Корпоративний сайт ще називають віртуальним офісом: це означає, що відвідування такого інтернет-майданчика за своєю результативністю рівнозначно візиту до центрального офісу компанії.

По суті, корпоративний сайт являє собою розширену версію сайту-візитки. Але якщо сайт-візитка більше підходить представникам малого бізнесу, то корпоративний портал створюється для великих компаній, корпорацій, холдингів. Втім, і невеликі фірми можуть істотно зміцнити свої позиції завдяки якісному і добре налагодженому корпоративному сайту.

Будь-які з сайтів повинні мати зважені складові, для того щоб бути успішними і мати можливість конкурувати у своїй галузі. Для створення дійсно конкурентоздатного вебсервісу, незалежно від його спрямованості, необхідно звернути увагу на декілька чинників:

1. Дизайн.
2. Контент.
3. Унікальний трафік.

Процес розробки комерційного сайту неможливий без попередньої постановки цілей і завдань. Аналіз сайтів конкурентів, аналіз цільової аудиторії, виявлення «сильних» і «слабких» сторін проєкту – це основні завдання, з яких починається розробка комерційного сайту.

У даній роботі за основу розробки програмного забезпечення було взято спеціалізовані українські сервіси з логістики, такі як «Logist.fm», «Logistic Plus Ukraine» та «РіоТранс». Дані сервіси пропонують послуги з перевезення та/або налагодження вже існуючих логістичних сполучень. Більшість сервісів мають структуру типу «Landing Page». Функціонал даних вебсервісів не розкриває всіх можливості для транспортної сфери діяльності.

Вебсервіс, що створюється, має наступні характеристики:

- Логіка сайту пишеться за допомогою мови програмування Python.
- Frontend вебсервісу розробляється за допомогою PHP. Дана мова програмування дозволяє швидко, зручно та гарно оформляти візуальну частину сервісу.
- Наповнення вебсервісу корисною ексклюзивною та необхідною інформацією.
- Налаштована система зв'язку із користувачем.
- Швидкість загрузки сайту.
- Кросс-платформеність або багатоплатформеність.

На рис. 1 зображено інтерфейс головної сторінки вебсайту, що розроблюється. На ній користувач може дізнатись основну інформацію про вантажоперевезення, замовити доставку, розрахувати приблизну вартість перевезення та прочитати відгуки про компанію.

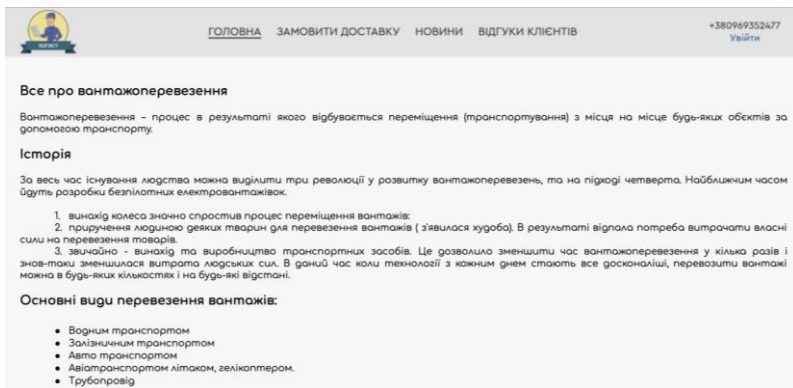


Рисунок 1 – Вигляд головного меню сайту

Таким чином, використання спеціалізованого програмного забезпечення допоможе транспортним компаніям з організацією логістичних операцій та просуванню послуг на ринку товарів.

Стовманенко В. О.,
аспірант кафедри ІПЗ,
Давиденко Є. О.,
канд. техн. наук, доцент, завідувач кафедри ІПЗ,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

РОЗПОДІЛЕНИЙ ІНТЕРПРЕТАТОР ДЛЯ ОБМІНУ ДАНИМИ В ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМАХ

У хмарних середовищах часто застосовується підхід із розбиттям системи на компоненти-сервіси. Кожен з них виконує якусь частину роботи, забезпечуючи загальну роботу системи. З одного боку таким чином можна гнучко працювати з навантаженням (додавати більше ресурсів саме на ту галузь, яка цього потребує), тримати код ізольованим (щоб кожен компонент працював тільки з певним функціоналом і не впливав на роботу інших компонентів) та не мати єдиної точки відмови. Проте подібна побудова системи викликає і певні складнощі. Компоненти системи, що можуть перебувати на різних фізичних пристроях мають використовувати певний канал для обміну даними. Це можуть бути синхронні протоколи обміну на кшталт HTTP, RPC побудований на обміні двійковими даними через сокет або асинхронний обмін повідомленнями. Окрім того існують проблеми синхронізації даних між різними компонентами. Переваги переважають недоліки. І на поточний час є різні підходи для усунення цих недоліків. Дана робота концентрується на механізмах взаємодії між компонентами.

З кінця 90-их років ХХ-го століття велися розробки систем, які б спрощували побудову розподілених систем. Розподілений інтерпретатор мови Erlang став відкритим у 1998-му році. Особливістю мови є реалізація ООП. Кожен об'єкт класу отримує свій «зелений» потік, який виконується інтерпретатором. Взаємодія між класами здійснюється шляхом обміну повідомленнями. Всю необхідну інфраструктуру для цього забезпечує інтерпретатор. Окрім того інтерпретатори, що виконуються в межах однієї мережі можуть об'єднуватися в групи, які «знають» про існування інших і можуть автоматично пов'язуватися для виконання спільного завдання.

В 1993-му році вийшла перша версія операційної системи Plan 9. Вона дозволяє використовувати ресурси (процесори, сховище, принтери, графіка) різних пристроїв у мережі. Застосунки з локального пристрою можна виконувати на віддалених машинах і отримувати виведення на локальному комп'ютері. Деякі системні компоненти

операційної системи також можуть працювати на іншій. Система не забезпечує автоматичного обміну даних між компонентами та процесами, але реалізує протокол 9p, який дозволяє представляти ресурси процесу в якості елементів файлової системи. До того ж кожне вхідження може не знаходитися на локальному комп'ютері.

Проте розвиток протоколів обміну даних загального призначення (Protobuf для синхронного обміну даних або AMQP для асинхронного) запропонували рішення, які дозволяли достатньо просто обмінюватися даними і не витрачати значних обсягів часу для реалізації подібного обміну. Тому подальший розвиток розподілених автоматичних засобів не був перспективним так як не пропонував достатньої кількості покращень, які можна було б отримати з розвитком складних алгоритмів розподілу навантаження між вузлами мережі. Проте з поточним рівнем використання хмарних середовищ та засобів поділених на сервіси, запит на подібні системи може бути збільшено. Окрім того подібні підходи до розподілу можна застосовувати і у вбудованих системах, які мають достатньо ресурсів для виконання певної «полегшеної» версії інтерпретатора, що дозволить об'єднувати подібні системи з іншими системами, що виконуються, наприклад, у хмарному середовищі.

Відмінністю, що відрізняє проєкт інтерпретатора від прикладів наведених є покладання на наявні протоколи обміну даними (AMQP) використання імперативної мови програмування (Basic, Perl, Python тощо) із реалізацією ООП у більш звичному для більшості програмістів вигляді (на відміну від функціонального підходу Erlang з синтаксисом взятим з Prolog). Притому з наявних прикладів можна взяти алгоритми для розподілення навантаження та виконання коду на різних фізичних пристроях. Зокрема для більш вдалого розподілу виконання задач і встановлення пріоритетів від користувача для виконання можна використати алгоритми нечіткої логіки. В той же час виконання СППР з використанням розподілених інтерпретаторів дозволить спростити масштабування задач у середовищах з багатьма фізичними пристроями. Інтерпретатор має забезпечити автоматичну передачу даних та завдань на інші вузли мережі, тому автор коду не буде займатися синхронізацією. Окрім того варто розглянути декілька підходів до розподілу навантаження:

– *Автоматичний*. Інтерпретатор має сам визначати ситуації, які дозволять перенести виконання тієї чи іншої ділянки коду на інший вузол. Це збільшує складність інтерпретатора і вимагає складного аналізу коду, що виконується. Окрім того користувач втрачає змогу оптимізувати чутливі до швидкодії ділянки коду. Маються на увазі ситуації, коли обсяг роботи є меншим ніж певне критичне значення і виконання на одному комп'ютері дасть кращий результат ніж розподіл.

– *Напівавтоматичний*. Вимагає введення окремої мовної конструкції, що переносила б виконання на інший вузол. Таким чином код написаний раніше працюватиме без змін і, в залежності від реалізації, без значних змін у швидкодії. Алгоритм розподілення навантаження може бути значно простіший, так як очікується команда від користувача коли варто це робити. Таким чином інтерпретатор тільки забезпечує перенесення потрібних даних обраним каналом зв'язку. Це дозволяє мати гнучкість над рівнем розподіленості та швидкодією. Розробник завжди знає, коли відбувається швидкий виклик локального методу, а коли відносно дорогий в плані використаних ресурсів, мережевий виклик.

УДК 004.4'22

Фісун М. Т.,
д-р техн. наук, професор, кафедри ІПЗ,
Кандиба І. О.,
старший викладач кафедри ІПЗ,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

ПОДАЛЬШИЙ РОЗВИТОК МЕТОДОЛОГІЇ IDEF3 ЗА РАХУНОК ДОДАВАННЯ МЕТОДІВ МЕРЕЖНОГО ПЛАНУВАННЯ

Однією із технологій моделювання дій є методологія IDEF3, основою представлення якої є відповідні діаграми. Інструментарій або CASE (Computer Aided System Engineering) для її використання реалізовано в пакеті програм BPWin. Об'єктами діаграм IDEF3 є;

– *Одиниця роботи* (Unit of Work – UOW) – важливий компонент моделі. Діаграми IDEF3 відображають дію у вигляді прямокутника і можуть мати такі синоніми, як модуль, функціонал, підсистема, процес і т.п.

– *Зв'язки* виділяють істотні взаємини між діями. Всі зв'язки в IDEF3 є одно напрямленими. В діаграмах IDEF3 три типи зв'язків; часовий, об'єктний і невизначений.

– *З'єднання розгалужують* зв'язків (jump) або перехрестя. На цих перехрестях використовується алгебра логіки – логічні операції **OR** (або), **XOR** (ексклюзивне або), **&** (і).

Крім когнітивного представлення моделі поведінки об'єкта дослідження у вигляді діаграм IDEF3, цю модель можна представити у вигляді квадратної квазіматриці, у якій роль «змінних» виконують

UOW, а в комірках представлені зв'язки (перехрестя) між ними та характеристики цих зв'язків. Зазначені об'єкти, з'єднані за певними правилами, ще називають сценарієм розвитку подій.

Подальший розвиток цієї методології можна здійснити шляхом як це можливе, формального опису UOW. Пропонується там, де UOW можна представити мережевими графіками (МГ) виконання складових UOW, які позначимо w_i , тобто $W=\{w_i\}$, а сам мережевий графік являє собою орграф $G(w_i, v_j, F_{ij})$, де, крім w_i , $V=\{v_j\}$ – множина вершин; F_{ij} – функція відображення V на W . Класичний мережевий графік є жорсткою структурою, тобто не припускає альтернатив. Суттєвим кроком для отримання можливості моделювання різних варіантів реалізації проектів стало визначення типових альтернативних фрагментів (ТАФ), які розглядаються як «будівельні» модулі МГ. При його побудові ТАФ обирають в залежності від реальних обставин та прогнозу надходження тих чи інших ресурсів. В узагальненому МГ типові альтернативні фрагменти з'єднуються у вершинах, як і в діаграмах IDEF3, символами логічних операцій **OR** і **&**. Обробка таких МГ складніше, ніж простих, тому такі об'єкти. Тому вони не реалізовані у відомих пакетах програм з управління проектами. В той же час, в діаграмах IDEF3 передбачено як зазначені операції, так ще й операція **XOR**.

УДК 519.816

Швед А. В.,

д-р техн. наук, доцент, доцент кафедри ІПЗ,
ЧНУ ім. Петра Могили, Миколаїв, Україна

ОСОБЛИВОСТІ РОЗРОБКИ СИСТЕМ СИТУАЦІЙНОГО УПРАВЛІННЯ РЕАЛЬНОГО ЧАСУ

Управління сучасними підприємствами в умовах зростаючої конкуренції, підвищення складності задач, що вирішуються, апіорної невизначеності і високої динаміки змін попиту і пропозиції, вимагають щоденного вирішення взаємопов'язаних і взаємоузгоджених задач організації робіт, планування, моніторингу та контролю ресурсів, що використовуються. Такий стан справ поступово призводить до відмови від класичних принципів менеджменту, представленого «програмно-цільовим управлінням» і переходу на більш гнучке і ефективне «ситуаційне управління». Основою такого управління є командна робота, побудована на взаєморозумінні і згоді (консенсусі) осіб, які приймають рішення в режимі реального часу.

Ситуаційне управління при цьому заперечує формальне механічне слідування створеними раніше «повторюваними» правилами без будь-якого аналізу суті ситуації і, навпаки, пропонує детальний розбір всіх особливостей ситуації і колективну генерацію рішень в форматі діалогу всіх зацікавлених осіб.

Таке управління реалізується через створення ситуаційних центрів (СЦ), розробка і впровадження яких активно проводиться в багатьох розвинених країнах світу. СЦ стають основою для управління великими системами типу держава, галузь, оборона, екологічні об'єкти, об'єкти з підвищеним ступенем небезпеки та ін. В останні роки СЦ почали розроблятися і впроваджуватися в Україні.

Участь людей в ситуаційному управлінні говорить про те, що СЦ – перш за все «організаційна система, яка використовує підходи ситуаційного управління складними об'єктами (системами) на базі нових інформаційних технологій, що дозволяють об'єднати людино-машинні можливості в режимі колективної підготовки і прийняття рішень». Оскільки СЦ являє собою організаційну систему, розглянемо більш докладно ключові ролі фахівців, які беруть участь в процесі колективної підготовки і прийнятті рішень для ситуаційного управління об'єктом. Спрощений цикл прийняття рішень в СЦ наведено на рис. 1.

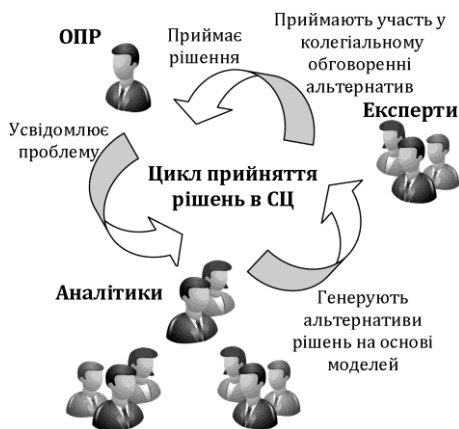


Рисунок 1 – Спрощений цикл прийняття рішень в СЦ

Можна виділити наступні ролі фахівців, які беруть участь в процесі прийняття рішень:

ОПР – особа, що приймає рішення – це спеціаліст в предметній області, що обговорюється в СЦ, який усвідомив необхідність

колегіального вирішення деякої управлінської проблеми, та приймає участь в колективному прийнятті рішень та несе відповідальність за прийняте рішення.

Аналітик – це фахівець в предметній області, що обговорюється в СЦ, який має досвід і навички для моделювання можливих альтернативних рішень управлінської проблеми і не несе відповідальність за прийняте рішення.

Експерт – це досвідчений фахівець в предметній області, що обговорюється в СЦ, який бере участь в колегіальному обговоренні можливих альтернативних рішень управлінської проблеми і не несе відповідальності за прийняте рішення.

Прийняття рішень в ситуаційному управлінні повинно проводитись в реальному часі у міру виникнення проблемних ситуацій. Зважаючи на це, системи ситуаційного управління реалізуються у вигляді інтелектуальних систем підтримки прийняття рішень реального часу (ІСППР РЧ.) На системи, що функціонують в реальному часі накладаються певні обмеження на часові (динамічні) характеристики.

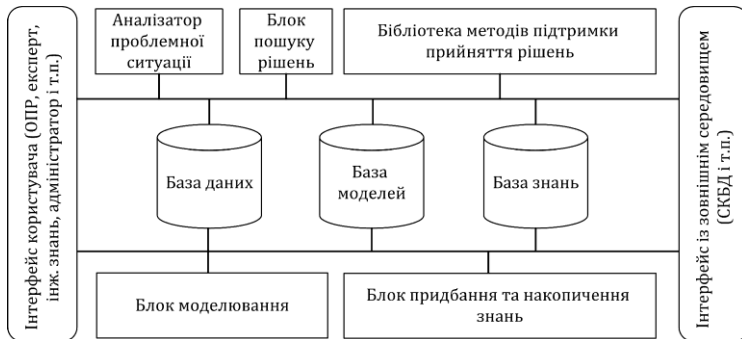


Рисунок 2 – Базова архітектура ІСППР РЧ

При побудові таких систем необхідно враховувати їх специфіку:

- необхідність отримання рішення в умовах часових обмежень, визначених реальним керованим процесом;
- необхідність врахування часового фактора при описі проблемної ситуації в процесі пошуку рішень;
- неможливість отримання всієї об'єктивної інформації, необхідної для рішення, і в зв'язку з цим використання суб'єктивної експертної інформації;
- багатоваріантність пошуку рішень;
- необхідність і введення додаткової інформації в базу знань системи при пошуку рішення.

Стисло розглянемо основні призначення блоків і елементів базової архітектури ІСППР РЧ. Узагальнена структура ІСППР РЧ наведена на рис. 2.

Аналізатор проблемної ситуації є одним із основних блоків системи підтримки прийняття рішень (СППР), він допомагає досліджувати і структурувати задачу, що розв'язується. У межах такої структуризації формується перелік факторів і альтернатив, визначаються і описуються їх властивості (ознаки, характеристики, атрибути), виявляються обмеження, задаються критерії оцінки. Цей блок тісно взаємодіє з базою моделей, в якій міститься набір можливих моделей, до яких може привести структуризація задач для вже розглянутих проблемних ситуацій. Серед моделей можуть бути як традиційні об'єктивні моделі, що описуються методами традиційної математики, так і суб'єктивні моделі, побудовані на основі суб'єктивних оцінок експертів.

Блок пошуку рішень є ключовим елементом СППР і включає в себе засоби, які допомагають користувачам знаходити найкращі шляхи вирішення попередньо структурованих проблем. На вхід блоку надходить, з одного боку, формалізоване уявлення проблеми, а з іншого – вимоги до форми остаточного рішення, яке може бути представлено трьома типовими задачами: впорядкування множини об'єктів; виділення з множини об'єктів одного або декількох кращих або допустимих; розбиття множини об'єктів на групи (класи, кластери).

Формальне представлення проблемної ситуації і форми остаточного рішення висувають певні вимоги до типу необхідних моделей і даних, формують потребу в суб'єктивних експертних знаннях, накладають обмеження на використовувані методи прийняття рішень. Для виконання вищезазначених умов до складу СППР включена бібліотека, яка містить різні методи розв'язку багатокритеріальних задач на основі об'єктивних і суб'єктивних моделей.

База даних являє собою традиційну компоненту для всіх комп'ютерних систем. У відповідності до задач прийняття рішень база даних повинна містити фактографічні відомості, експертні оцінки, текстові документи та іншу інформацію. База знань повинна містити об'єктивні знання про предметну область користувача, суб'єктивні правила і критерії, що відображають досвід ОПР та експертів. Крім цього в базі знань повинен знаходитися так званий «архів історичних аналогій», в якому можна було б накопичувати інформацію про результати застосування СППР в минулому при вирішенні конкретних задач. Для моделювання та подання знань використовується цілий ряд методів, частина з яких розглянута в даному навчальному посібнику

(когнітивні карти, метод міркувань за прецедентами, теорія грубих множин тощо).

Інтерфейс «користувач-система», що забезпечує зв'язок користувача з компонентами СППР, включає засоби управління базами даних, моделей і знань, управління та генерації діалогу допомоги користувачу. Засоби управління базами даних, моделей і знань служать для створення, вилучення та модифікації вмісту відповідних баз. При вирішенні задач користувач взаємодіє з кожною з баз через блок управління і генерації діалогу. В скрутних випадках інтерфейс повинен надавати користувачу допомогу у вигляді інструкцій, підказок, повідомлень і т. п. [5].

Висновки. На сьогодні досить актуальною проблемою в області штучного інтелекту є розробка інтелектуальних систем реального часу, типовими представниками яких є ІСППР РЧ, зорієнтовані на відкриті і динамічні предметні області. В основі таких систем лежать здатні до адаптації, модифікації та навчання моделі подання та оперування знаннями, зорієнтовані на специфіку проблемної (предметної) області. В роботі проаналізовані особливості та загальні аспекти побудови систем ситуаційного управління, що реалізуються у вигляді ІСППР РЧ; розглянуті основні ролі фахівців, які беруть участь в процесі прийняття рішень для ситуаційного управління аналізованого об'єкта.

УДК 004.67

Швед А. В.,

д-р. техн. наук, доцент, доцент кафедри ІПЗ,

Фаленкова М. В.,

викладачка кафедри ІПЗ,

Давиденко Є. О.,

канд. техн. наук, доцент, завідувач кафедри ІПЗ,

ЧНУ ім. Петра Могили, м. Миколаїв, Україна

ІНФОРМАЦІЙНО-АНАЛІТИЧНА СИСТЕМА ОЦІНЮВАННЯ СТАЛОГО РОЗВИТКУ СУСПІЛЬСТВА

Розвиток і зростання масштабів господарської діяльності, в ході яких людина використовує дедалі більшу кількість природних ресурсів, що обумовлюють тотальне посилення антропо-техногенного тиску на навколишнє середовище та порушення рівноваги в навколишньому середовищі. А це, в свою чергу, призводить до загострення соціально-економічних проблем. Одночасно з вичерпанням запасів невідновлю-

ваних сировинних і енергетичних ресурсів посилюється забруднення навколишнього середовища, особливо водних ресурсів та атмосферного повітря, зменшуються площі лісів і родючих земель, зникають окремі види рослин, тварин тощо. Все це в кінцевому підсумку підриває природно-ресурсний потенціал суспільного виробництва і негативно позначається на здоров'ї людини.

Сталий розвиток (СР) – це такий розвиток суспільства, коли задоволення потреб сьогodнішніх поколінь не повинно ставити під загрозу можливості майбутніх поколінь задовольняти свої потреби.

Концепція сталого розвитку покликана узгодити три основні компоненти: економічну, екологічну та соціальну таким чином, щоб підтримувати постійний рівень якості та безпеки життя людей від покоління до покоління. Дану концепцію можна представити більш формально, як:

$$CP=(E, C, П),$$

де СР – сталий розвиток, Е – економічна складова, С – соціальна складова, П – природоохоронна (екологічна) складова.

Цю тріадну концепцію можна представити в графічному вигляді, як показано на рис. 1.



Рисунок 1 – Тріадна концепція сталого розвитку

Системне узгодження і баланс цих трьох складових – завдання величезної складності. Зокрема, взаємозв'язок соціальної і екологічної складових приводить до необхідності збереження однакових прав сьогodнішніх і майбутніх поколінь на використання природних ресурсів. Взаємодія соціальної та економічної складових вимагає досягнення справедливості при розподілі матеріальних благ між людьми й надання цілеспрямованої допомоги бідним верствам суспільства. І, нарешті, взаємозв'язок природоохоронної та економічної складових потребує вартісної оцінки техногенних впливів на навколишнє середовище. Вирішення цих завдань – головний виклик сучасності для національних

урядів, авторитетних міжнародних організацій та всіх прогресивних людей світу.

Для вимірювання процесів сталого розвитку, в просторі вказаних трьох складових в роботі була запропонована метрика (МІУР), що відповідає стандартам і положенням, розробленим Комісією ООН і максимально наближена до предметної області, яка досліджується.

Згідно МІУР сталий розвиток оцінюється за допомогою відповідного індексу (I_{cp}) в просторі трьох вимірів: економічного ($I_{ек}$), екологічного (I_e) і соціально-інституційного (I_c) (рис. 2). Цей індекс є вектором, норма якого визначає рівень сталого розвитку, а його просторове положення в системі координат ($I_{ек}$, I_e , I_c) характеризує ступінь «гармонійності» цього розвитку (ступінь гармонізації сталого розвитку – G). Рівновіддаленість вектора I_{cp} від кожної з координат ($I_{ек}$, I_e , I_c) буде відповідати найбільшій гармонійності сталого розвитку.

Наближення цього вектора до однієї з координат буде вказувати на пріоритетний розвиток за відповідним виміру і нехтування двома іншими. Індекс (I_{cp}) і ступінь гармонізації сталого розвитку (G) обчислюються за складовими ($I_{ек}$, I_e , I_c). Цю метрику можна представити у вигляді

$$МІУР(I_{cp}, G),$$

де МІУР – метрика вимірювання сталого розвитку; I_{cp} – індекс сталого розвитку; G – рівень гармонізації сталого розвитку.

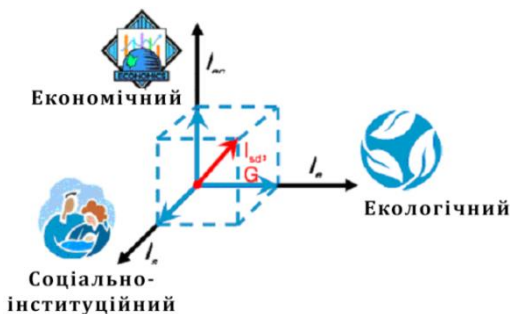


Рисунок 2 – Виміри сталого розвитку

МІУР складається з таких основних компонентів, як індекс сталого розвитку, ступінь гармонізації СР, індекси вимірювань, індикаторів, параметрів.

Індекс сталого розвитку – інтегрована оцінка, яка враховує спільно всі три виміри сталого розвитку, і, тим самим, відображає

взаємозв'язок між трьома нероздільними сферами розвитку суспільства: економічної, екологічної та соціальної.

Ступінь гармонізації сталого розвитку (СГСР) – це показник, який відображає баланс між економічним, екологічним і соціально-інституціональним вимірами СР. Під СГСР (рис. 3) будемо розуміти кут між вектором I_{cp} з нормою

$$\|I_{yp}\| = \sqrt{(I_{ек}^2 + I_e^2 + I_c^2)} \quad (1)$$

і «ідеальним» вектором, який є рівновіддаленим від кожної з координат $I_{ек}$, I_e , I_c з нормою

$$\|1\| = \sqrt{(1^2 + 1^2 + 1^2)} \quad (2)$$

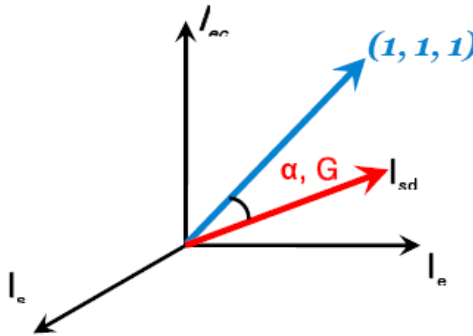


Рисунок 3 – Ступінь гармонізації сталого розвитку

Цей кут визначається зі співвідношення:

$$\alpha = \arccos \frac{I_{ек}^2 + I_e^2 + I_c^2}{\sqrt{3} \cdot \sqrt{(I_{ек}^2 + I_e^2 + I_c^2)}} \quad (3)$$

I змінюється в межах:

$$0 \leq \alpha \leq \alpha_{max}; \alpha \arccos \frac{1}{\sqrt{3}}.$$

Для зручності порівняння регіонів за ступенем гармонізації сталого розвитку приведемо цей показник до наступного нормованому вигляду:

$$G = \frac{G' - G_{min}}{G_{max} - G_{min}}, \quad (4)$$

де $G' = 1 - \frac{\alpha}{\alpha_{max}}$; $G \frac{\alpha_1}{\alpha_{max,max}}$; $G \frac{\alpha_2}{\alpha_{max,min}}$; $\alpha_1 = 0$; $\alpha_2 = \frac{45}{\pi} \arccos \frac{1}{\sqrt{3}}$.

У результаті застосування цієї нормалізації ступінь гармонізації G буде змінюватися в діапазоні $[0-1]$. Вона буде рости в міру наближення G до 1 і зменшуватися при наближенні G до 0. Таким чином, модель МІУР дозволяє обчислювати індекс сталого розвитку

(I_{cp}) і ступінь гармонізації цього розвитку (G) для кожного регіону країни, для якого існують дані індекси.

Аналізуючи поведінку ступеня гармонізації сталого розвитку G для різних міст України, можна зробити висновок, що цей показник знижується в міру зменшення індексу сталого розвитку. Це пояснюється тим, що в міру ослаблення політики сталого розвитку для різних міст, гармонізація їх розвитку в економічному, екологічному та соціально-інституційному вимірах погіршується.

**Підсекція: АВТОМАТИЗАЦІЯ ТА
КОМП'ЮТЕРНО-ІНЖЕНЕРНІ ТЕХНОЛОГІЇ**

УДК 504.05:574.59:581.63

Прищепов О. Ф.,

канд. техн. наук, доцент, доцент кафедри екології

Щесюк О. В.,

канд. техн. наук, доцент,

доцент кафедри АКІТ,

Розганяєв Д. О.,

студент 4 курсу спеціальності АКІТ,

Яковішин А. Я.,

студент 4 курсу спеціальності АКІТ

ЧНУ ім. Петра Могили, м. Миколаїв, Україна

**МЕТОДИКА ОЦІНКИ РІВНЯ ЕКОЛОГІЧНОЇ
ДОСКОНАЛОСТІ ДВЗ ТРАНСПОРТНОГО ЗАСОБУ**

Постановка проблеми. Оцінка рівня екологічної безпеки двигуна транспортного засобу за стандартною методикою є досить складним завданням, яке пов'язане з визначенням індексу якості середовища F , для розрахунку якого необхідно виявляти концентрацію шкідливих домішок у навколишньому середовищі та порівнювати її із гранично допустимими концентраціями (ГДК). Мобільність транспортного засобу, природно-кліматичні явища, розташування автомобільних, залізничних та інших шляхів у не населених пунктах, а також інші фактори не дозволяють на цьому етапі об'єктивно оцінювати рівень екологічної безпеки транспортних засобів. Цю стандартну методику можна застосовувати у досить обмежених умовах, наприклад, окремі райони або вулиці населених пунктів, закриті приміщення тощо.

В даний час у всьому світі йде розробка та впровадження системи нормування екологічних вимог до об'єктів транспорту та транспортних технологій у вигляді гранично допустимих норм викиду токсичних речовин з відпрацьованими газами, рівнів шуму, вібрацій, електромагнітних полів, питомих обсягів споживання окремих видів природних ресурсів, рівня комфорту та ін. Ці норми безпосередньо пов'язані з ГДК окремих домішок в атмосферному повітрі, воді, ґрунті, на конкретних площах території та встановлюються як компроміс у задоволенні різноспрямованих вимог (суспільна потреба, технічна можливість реалізації, вартість). Підходи до вирішення цієї комплексної

проблеми висвітлені в ряді роботи, де здійснено взаємопов'язання екологічних нормативів об'єктів транспорту з санітарно-гігієнічними нормами.

Нормування токсичності відпрацьованих газів транспортних засобів здійснюється з метою отримання порівняних оцінок екологічної досконалості різних конструкцій, матеріалів, технологічних процесів та управління рівнем впливу на навколишнє середовище. Норми викидів забруднюючих речовин нових транспортних засобів розробляються та приймаються багатьма країнами, а також спільнотою країн. У таблиці наведено значення норм викидів нових автомобілів у європейських країнах за типом випробувань у їздових циклах.

Таблиця 1 – Динаміка значень норм викидів легковими автомобілями масою до 1250 кг, г/км

Ступінь	Рік введення	Частинки	NO_x	C_xH_y	CO
EURO 1	1993	- / 0.14	0.97 / 0.97	-	2.72 / 2.72
EURO 2	1996	- / 0.08	0.5 / 0.67	-	2.2 / 1.0
EURO 3	2000	- / 0.05	0.14 / 0.5	0.17 / 0.06	1.5 / 0.6
EURO 4	2005	- / 0.025	0.07 / 0.25	0.08 / 0.05	0.7 / 0.47

Примітка. Чисельник / знаменник - бензинові / дизелі

Визначення рівня екологічної досконалості ДВЗ. Для оцінки рівня екологічної безпеки ДВЗ авторами пропонується введення принципово нового екологічного параметра – рівня екологічної досконалості двигуна транспортного засобу, який визначатиме величину відхилення викиду шкідливих домішок в атмосферу з вихлопними газами від норми їх викиду, встановленої в конкретній країні та в даний час, наприклад,

$$E_s^{EURO4} = \frac{N_s^{EURO4}}{m_{sp} K_{ST.C}}$$

де E_s^{EURO4} – рівень екологічної досконалості транспортного засобу за нормами європейських країн для s-ої забруднюючої речовини;

m_{sp} – пробіговий викид s-ої забруднюючої речовини;

$K_{ST.C}$ – коефіцієнт, що враховує вплив технічного стану автомобіля на масовий викид s-ої забруднюючої речовини;

N_s^{EURO4} – норма викиду s-ої забруднюючої речовини для нового транспортного засобу, яка встановлена у європейських країнах, г/км.

Значення параметра E_s може бути рівним одиниці, бути меншим одиниці, а в деяких випадках і більше одиниці, якщо двигун

транспортного засобу в даний час отримав більш високий ступінь досконалості та його шкідливі викиди менші за встановлені норми. Прикладом можуть бути імпортовані автомобілі, що експлуатуються в Україні.

У міру зносу двигуна параметр E_s знижуватиметься (крім E_{NO}) і прийме мінімальне значення до кінця його ресурсу.

Поточне значення $E_s(L)$ при $L > L_{ycm}$ у процесі експлуатації двигуна можна визначати з виразу

$$E_s(L) = \frac{N_s}{m_{s_{np}} \left[(K_{ST.C_{min}} + a(L - L_{ycm}) + b(L - L_{ycm})^2) \right]}$$

А поточне значення $E'_s(L)$ двигуна з підвищеним ресурсом при $L > L_{ycm}$ буде рівним

$$E'_s(L) = \frac{N_s}{m_{s_{np}} \left[(K_{ST.C_{min}} + a'(L - L_{ycm}) + b(L - L_{ycm})^2) \right]}$$

При $L < L_{ycm}$

$$E_s(L) = \frac{N_s}{m_{s_{np}} \cdot K_{ST.C_{min}}};$$

та $E'_s(L) = \frac{N_s}{m_{s_{np}} \cdot K_{ST.C_{min}}}$ при $L < L'_{ycm}$

Зміна параметра $\Delta E_s(L)$ на ділянці $L'_{ycm} - L_{ycm}$, пов'язаного з підвищенням ресурсу двигуна, може бути обчислено за формулою

$$E_s(L) = \frac{N_s}{m_{s_{np}} \left[\left(\frac{1}{K_{ST.C_{min}}} - \frac{1}{K_{ST.C_{min}} + a(L - L_{ycm}) + b(L - L_{ycm})^2} \right) \right]}$$

На ділянці $P' - L_{ycm}$

$$\Delta E_s(L) = \frac{N_s}{m_{s_{np}} \left[\left(\frac{1}{K_{ST.C_{min}} + a'(L - L'_{ycm}) + b'(L - L'_{ycm})^2} - \frac{1}{K_{ST.C_{min}} + a(L - L_{ycm}) + b(L - L_{ycm})^2} \right) \right]}$$

Висновок. Запропоновано новий екологічний параметр - рівень екологічної досконалості двигуна транспортного засобу, за допомогою якого можна визначати величину відхилення викиду шкідливих домішок в атмосферу з вихлопними газами від норми їхнього викиду, встановленої в конкретній країні та в даний час.

Кубов В. І.,
канд. фіз.-мат. наук, доцент, доцент кафедри АКТ,
Димитров Ю. Ю.,
викладач кафедри АКТ,
Руденко Д. Г.,
магістрант кафедри АКТ,
Курков М. Д.,
магістрант кафедри АКТ,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

ДОСВІД ЗАСТОСУВАННЯ ХМАРНИХ ТЕХНОЛОГІЙ У СИСТЕМАХ АВТОМАТИЗАЦІЇ НА БАЗІ GSM-МОДЕМУ ТА OWENCLOUD

Вступ. Концепція індустрій IIOT та Industry 4.0 передбачає оптимізацію виробничих витрат шляхом максимальної автоматизації технологічного процесу. Використання хмарних технологій у системах автоматизації є невід'ємною частиною цих концепцій, що дозволяє скоротити OPEX-витрати.

Мета роботи – розглянути можливість і ефективність застосування засобів GSM-комунікації ТОВ «ВО «ОВЕН» у зв'язці з GSM-мережами в Україні.

Використання хмарних технологій у системах автоматизації є невід'ємною частиною концепцій IIOT та Industry 4.0, що дозволяє скоротити OPEX-витрати виробничих та енергетичних підприємств. Для завдань диспетчеризації та керування віддалених об'єктів, на яких відсутня мережа Інтернет, проте є покриття мобільних операторів, можливо використовувати спеціалізовані промислові пристрої радіозв'язку. Використання застарілих концепцій передачі за радіозв'язком передбачає наявність пристрою прийому і передачі як у польових ділянках технологічних систем, так і у диспетчерському пункті, або за необхідності декількох диспетчерських пунктах. Ця концепція підвищує витрати на організацію системи диспетчеризації. В сучасних концепціях Internet of Things передача даних відбувається шляхом завантаження технічних параметрів у так зване «хмарне» середовище. Для вивантаження даних із хмарного середовища у диспетчерських пунктах необхідний лише доступ до мережі Інтернет.

Для тестування організації моніторингу технологічних параметрів за GSM-каналами зв'язку в Україні нами було розроблено і виготовлено випробувальний стенд віддаленого GSM-моніторингу. У

випробувальному стенді були використані наступні прилади систем автоматизації: пристрій керуючий ОВЕН ПР-200, мережевий GSM шлюз ОВЕН ПМ-210, джерело живлення Meanwell (=24В, 60 Вт.), з'єднувальні клеми, блок вимірювання температури ТЕМІКС БИТ-12Д-24 з інтерфейсом RS-485 (протокол Modbus).



Рис. 1. ОВЕН ПР-200 (зліва), ОВЕН ПМ-210 (в центрі) та джерело живлення 24В (справа) на випробувальному стенді

Під час роботи були виконані тестові передачі даних від Modbus-пристроїв у хмарне середовище сервісу OWENCloud. Після зміни вхідних параметрів Modbus-контролерів відслідковувався час оновлення даних у хмарному середовищі.

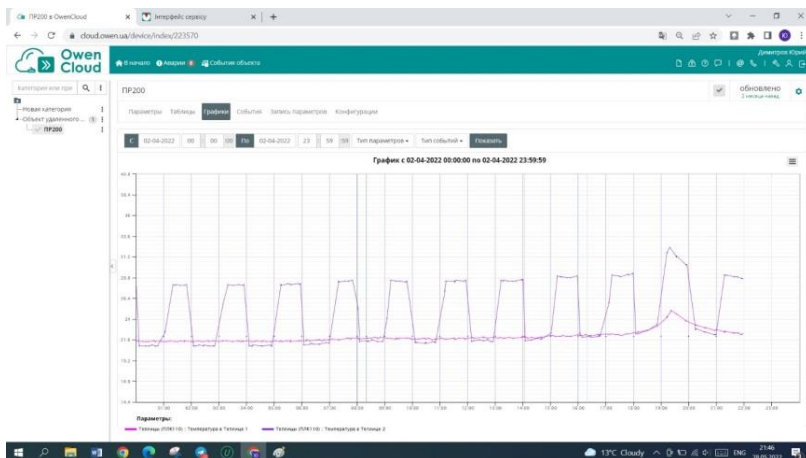


Рисунок 2 – WEB-інтерфейс сервісу OWENCloud

Налаштування через хмарний сервіс досить просте, хоча під час налаштування виникають непорозуміння, пов'язані з актуалізацією інструкцій з налаштування. В той же час ТОВ «ВО ОВЕН» це українська компанія, у якій є відділ технічної підтримки, а отже отримати консультації та технічну підтримку можливо українською мовою та досить швидко.

До мінусів можна віднести те, що у комплекті з пристроєм виробником надається антена для іншого радіочастотного діапазону, що в свою чергу призводить до нестабільної роботи процесу передачі даних від апаратної частини до хмарного сервісу, згідно спостережень від 1 хвилини до 5. При заміні антени на відповідну до частот GSM діапазону (900 МГц) ситуація зі зв'язком покращилась, проте все одно оновлені дані потрапляли до хмарної системи у межах 1 хвилини. Крім того GSM-модуль у пристрої лише один, а отже немає змоги виконати резервування передачі показань у випадку проблем зі зв'язком в певного оператора GSM-зв'язку.

Висновки. Дана система віддаленого моніторингу і керування не повинна використовуватись у технологічних процесах в яких важлива висока швидкість обробки інформації. Проте ця система є фінансово доступною, і може бути використана в системах, де швидкість збору інформації та керування не є критичними і можуть використовувати параметри з часом оновлення близько 10 хвилин.

УДК 62.52

Бєліков О. Є.,
канд. техн. наук, доцент кафедри АКІТ,
Нерега М. В.,
студент 4 курсу спеціальності АКІТ
Шенкевич В. М.,
старший викладач кафедри АКІТ,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

АВТОМАТИЧНИЙ ДЕЗИНФЕКТОР ДЛЯ РУК

Разом із поширенням інфекційних захворювань виникає потреба у регулярній дезінфекції рук, так як одним із найбільш поширених способів передачі інфекції є потрапляння клітин вірусу до слизових оболонок людини з поверхні рук при торканні обличчя. Повсюдне носіння рукавичок не може вирішити цю проблему, так як неможлива

достатньо часто їх зміна. Портативні дезінфектори для рук не завжди є в наявності при потребі дезінфекції.

Стационарні дезінфектори для рук та пульверизатори для дезінфікуючих розчинів потребують безпосереднього контакту із органами управління ними, що при їх використанні великою кількістю людей може мати зворотний ефект по відношенню до запобігання поширенню інфекцій. Тому найбільш оптимальним рішенням завдання дезінфекції рук або інших поверхонь і предметів є безконтактні засоби.

На рис.1 показано спрощену функціональну схему автоматичного дезінфектора для рук.



Рисунок 1 – Функціональна схема автоматичного дезінфектора для рук

У мінімальній конфігурації система складається з мікроконтролера, датчика присутності рук, помпи та одного розпилювача. У розширеній інтерактивній конфігурації наявні: однопоточний комп'ютер, датчик температури, датчик присутності рук, датчик присутності людини, а також підсилювач з динаміком для відтворення голосових команд, дисплеї, веб-камера, помпа, 3 форсунки для розпилювання, та модуль передачі даних для контролю та моніторингу.

Розпилювач працює за принципом мікророзпилення, що дозволяє використовувати мінімальну кількість антисептика та заощадити дезінфікуючий засіб. Розпилення рідкого антисептика на руки здійснюється, не торкаючись пристрою, що забезпечує чистоту рук. Апарат не є спеціальною медичною технікою, але може знайти широке застосування у лікувально-профілактичних установах України. Завдяки

здатності пристрою відтворювати голосові команди, ми залуцаємо більше людей. Також наявність безконтактного датчика температури та модуля передачі даних дає можливість створити певні пропускні пункти. При підвищеній температурі пристрій буде сигналізувати це звуковою командою та за необхідності відправляти дані на певний сервер для моніторингу та контролю людей з високою температурою під час епідемії.

УДК 681.625.9:531.1(043.3)

Гекова Т. В.,
студентка VI курсу кафедри АКІТ
Беліков О. Є.,
канд. техн. наук, завідувач кафедри АКІТ,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ КІНЕМАТИЧНИХ ПАРАМЕТРІВ РОБОТИЗОВАНИХ СИСТЕМ РУХУ В ПРОСТОРІ

Більшість інженерів світу ставить за мету підвищити якість продукції, яку продукує будь-яка роботизована або автоматизована система, до якої відносять 3D-принтер. Крім підвищення якості, при створенні об'ємного об'єкту, інженери прагнуть створити компактну (малогабаритну) роботизовану систему, яка б в найкоротші терміни справлялася з поставленими завданнями.

Узагальнити всі намагання інженерів можна констатацією того, що одночасно інженерська спільнота своєю системною роботою ставить за мету створити автоматизовану (роботизовану) систему, яка б сукупно могла створювати:

- високоякісні 3D моделі з пластику (висока точність, монолітність моделі, друк без потреби пост обробки);
- була малогабаритною та водночас автономною системою;
- своєчасно (максимально швидко) проводила пошаровий друк об'ємної деталі з пластику;

Беручи до уваги три ключових критерія, що вказані вище, було визначено, що роботизована система руху в просторі, простіше 3D-принтер, повинна базуватися на принципах кінематики – CoreXY, в якій передбачено пересування робочого столу вздовж саме Z осі. Принтер має обмежену (малу) область друку для об'ємних моделей, як

і декларувалось в технічному завданні проєкту. Така конфігурація сприяє (дозволяє) компенсації всіх вібрацій, які породжує сам принтер, під час роботи пристрою, а крім того прискорює процес друку об'ємних моделей (швидкість друку це не єдиний, але критичний показник роботизованої системи друку моделей з пластику).

Зважаючи на всі недоліки, які проявляються при пошаровому створенні деталей з пластику, було створено для демонстрації принципу роботи функціональну схему роботизованої системи руху в просторі, що призначена проводити пошаровий друк з пластику, рисунок 1.

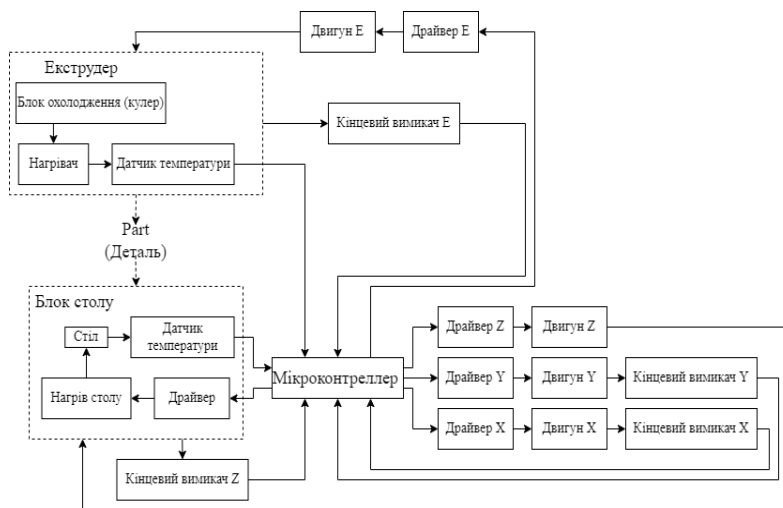


Рисунок 1 – Функціональна схема роботизованої системи руху в просторі задля пошарового нарощування моделі

Таким чином, зважаючи на недоліки споріднених роботизованих (автоматизованих) систем, у магістерській роботі поставлено за мету провести комплексний аналіз відомих роботизованих систем для пошарового друку об'ємних моделей, з подальшою розробкою заявленої роботизованої системи, що здійснює рух у просторі задля пошарового нарощування моделі з пластику, переваги якої наведені нижче.

Переваги роботизованої системи руху в просторі задля пошарового нарощування моделі з пластику:

- висока точність друку об'ємних моделей;
- роботизована система є безшумною впродовж своєї роботи;

- роботизована система не засмічує робочу область впродовж друку моделі;
- роботизована система має значно менші габаритні розміри, а також, закритий корпус;
- роботизована система є високопродуктивною та довготривалою у роботі;
- роботизована система не потребує постійного нагляду під час друку моделі, яку пошарово створює;
- роботизована система друкує затребуваними пластиками, включаючи нейлон;
- роботизована система має низьку собівартість.

УДК 621.38

Паламарчук А. О.,
студент 4 курсу спеціальності АКІТ,
Кубов В. І.,
канд. фіз.-мат. наук, доцент, доцент кафедри АКІТ,
Димитров Ю. Ю.,
викладач кафедри АКІТ,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

ВИКОРИСТАННЯ GSM-МОДЕМУ ДЛЯ КОНТРОЛЮ ТА УПРАВЛІННЯ СИСТЕМАМИ АВТОМАТИЧНОГО КЕРУВАННЯ

Монтаж, обслуговування, ремонт, керування та моніторинг стану електротехнічного обладнання систем автоматичного керування здійснюється кваліфікованими кадрами, витрачаючи при цьому значну кількість часу та коштів.

Для полегшення виконання таких процедур як керування та моніторинг потрібно звернутися до засобів дистанційного контролю та керування.

Задачу дистанційного контролю можна без значних витрат реалізувати на базі комплексу мікроконтролерної плати та GSM-модема. Мікроконтролерна плата типу Arduino на базі процесора ATmega328 дає можливість під'єднати до 10 цифрових та 6 аналогових сенсорів, або під'єднати 16 цифрових виконавчих пристроїв – актуаторів, що цілком достатньо для більшості задач малої автоматизації. Найбільш популярний GSM-модем на базі Sim900 має

достатньо простий послідовний інтерфейс з використанням текстових АТ-команд.

Програмно-апаратні ресурси SIM900 досить добре розвинені, завдяки чому звичний функціонал GSM-модуля поповнений такими програмними можливостями:

- GSM Location – визначення координат розташування за інформацією про базові станції.
- SSL – протокол захищених сокетів для забезпечення безпечної передачі даних через мережу.
- MMS – Мультимедійні повідомлення.
- Jamming Detection – детектування факту «глушіння» GSM-каналу.
- DTMF decoder – декодування DTMF-тонів.
- EMAIL – поштовий протокол передачі електронних листів.
- SMS Autorun – керування роботою модуля за допомогою SMS.
- PING – функція запиту «відлуння» через Інтернет для перевірки доступності адресата.
- AMR Play – функція відтворення аудіофайлів в динамік або у бік віддаленого абонента під час активного голосового з'єднання.
- 5MB User Memo – вбудована в модуль пам'ять для даних користувача.
- Easy Scan – функція отримання інформації про навколишні базові станції без підключення SIM-картки.

За допомогою даних програмних можливостей здійснювати контроль та керування можна за допомогою лише мобільного телефону та мікроконтролерного блока керування обладнаного GSM-модемом, котрий передає дані у вигляді SMS-повідомлень, одразу на мобільний телефон.

Тобто реалізація дистанційного контролю та управління електротехнічним обладнанням систем автоматичного керування за допомогою GSM-модему на базі Sim900 є досить простою, доступною та може бути використана для задач контролю та керування широкого спектру електротехнічного обладнання у будь-яких системах автоматизації.

Калевич Д. К.,
студент 4 курсу спеціальності АКІТ,
Шенкевич В. М.,
старший викладач кафедри АКІТ
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

ВИБІР СИСТЕМИ ВЕНТИЛЯЦІЇ В ОФІСНОМУ ПРИМІЩЕННІ

В Україні за останнє десятиріччя особливо в містах стрімко збільшилася кількість офісних установ. Офісні робітники сьогодні становлять найбільш масову професійну групу, чисельність якої перевищує 60–70 % працюючого населення, а якщо взяти місто Київ – то всі 90 %.

Для забезпечення оптимальних параметрів мікроклімату для офісних приміщень (температура та відносна вологість повітря, вміст CO₂, інтенсивність світла, шумове та пилове забруднення) та повітрообміну потрібно використовувати спеціальні технічні засоби.

Результати проведених еколого-гігієнічних досліджень офісних приміщень свідчать про те, що параметри мікроклімату часто не відповідають нормам чинного санітарного законодавства. Компанії часто орендують приміщення в давно збудованих непристосованих будівлях, або ж під офіси переобладнують підвальні, технічні та складські приміщення, які спочатку не призначалися для роботи людей впродовж дня. Перепланування будівлі, непрофесійний ремонт, встановлення сучасних герметичних вікон зменшує ефективність припливної витяжної системи вентиляції, якщо вона є.

Оптимізація якості повітря в приміщенні (Air quality index, AQI) необхідна для забезпечення самопочуття та продуктивності працівників. Персонал працює в середньому 40 годин на тиждень, і тепер, коли COVID-19 впровадив культуру віддаленої роботи у велику кількість організацій, дуже важливо, щоб ці комунальні будівлі забезпечували здорове та продуктивне робоче середовище.

Мікроклімат усередині будівель має особливе значення у сформованій епідеміологічній обстановці – за результатами останніх досліджень, контроль та регуляція показників температури та вологості повітря здатні значно скоротити масштаби поширення COVID-19.

Вентиляція передбачає процес видалення з приміщення забрудненого повітря на зміну свіжому. Саме завдяки цьому вдається організувати сприятливий мікроклімат у кімнатах будь-якого

призначення, повітря насичується киснем. Насамперед приміщення з гарною аерацією сприяє доброму самопочуттю та здоровому функціонуванню людського організму. Також підтримання оптимального мікроклімату – гарантія тривалої безпеки різних предметів, матеріалів, продуктів, а також уповільнення руйнування будівельних конструкцій. Вентиляція дозволяє впоратися із: зайвим теплом; надмірною вологою; пилом, брудом; шкідливими речовинами, парами, газами.

При неефективній вентиляції існує ризик зниження рівня кисню та збільшення CO₂, підвищення вологості та конденсація вологи на охолоджених поверхнях (стіни, скло, віконні косяки), яка забезпечує сприятливе середовище для розвитку цвілі, а також гнильних грибків. Використання неякісної теплоізоляції або неправильне її розміщення призводить до погіршення параметрів мікроклімату і хімічного складу повітря.

При плануванні системи вентиляції для приміщення, в якому постійно будуть люди, потрібно правильно вибрати спосіб роботи системи.

Побутові кондиціонери для офісів використовуються лише при можливості їх встановлення, а цьому часто заважають короткі траси побутових настінних кондиціонерів, відсутність місця встановлення внутрішнього блоку та зовнішнього. Невелика площа покриття побутовим кондиціонером змушує робити покупку та встановлення відразу кількох кондиціонерів у велике приміщення офісу.

Канальні внутрішні блоки відмінно підходять для офісів з високими стелями, каналний блок ховається в фальш стелі і повітропроводами виконується розведення подачі та забору повітря. Плюсом є те, що можна реалізувати подачу таким чином, щоб прохолодне повітря, що подається, не заважало співробітникам офісу.

Можна виділити три види каналних вентиляційних систем:

- витяжна
- припливна
- припливно-витяжна

Витяжний та припливний види вентиляції окремо не забезпечують нормальних умов для внутрішнього клімату приміщення. Отже єдиним оптимальним варіантом вентиляційної системи для офісу є припливно-витяжна вентиляція офісних приміщень. Цей тип вентиляції є поєднанням перших двох. Припливно-витяжна система забезпечує як приплив, так і видалення повітря, налагоджуючи хороший повітрообмін. Цей вид можна назвати найпридатнішим для офісних приміщень (рис.1).

Подача повітряних мас до кімнат здійснюється за допомогою мережі повітроводів, настінних решіток, стельових дифузорів. А саме каналне обладнання зазвичай знаходиться за підшивною стелею техприміщень, коридорів, санвузлів.

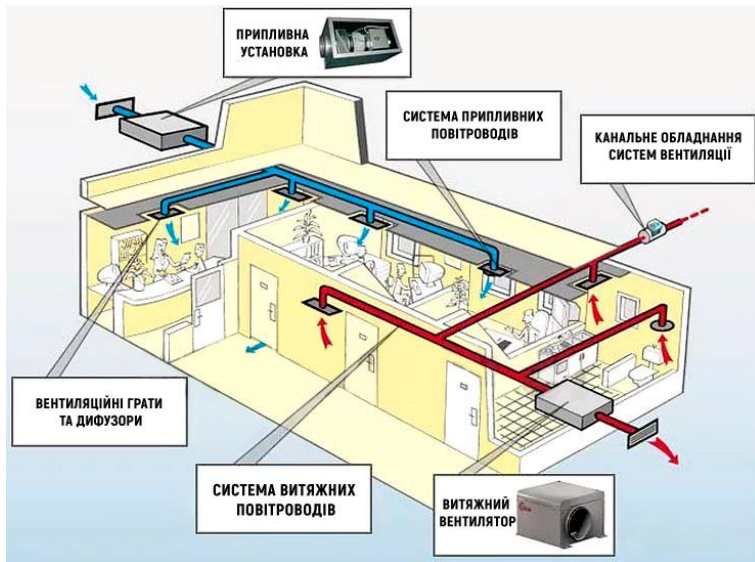


Рисунок 1 – Структурне представлення ВОВР

Але, ми можемо зіткнутися з питанням про те, що припливно-витяжна вентиляція в офісі є дорогою в процесі експлуатації за рахунок того, що припливне повітря в зимовий період року необхідно підігрівати до кімнатної температури.

Ця проблема особливо актуальна для таких об'єктів як офіси, оскільки:

- необхідна продуктивність системи вентиляції безпосередньо пов'язана з кількістю людей, що знаходяться в приміщенні, а в офісах їх, як правило, багато і вони знаходяться там постійно протягом робочого дня;
- часто в офісних приміщеннях немає альтернативних джерел тепла, крім електрики, яка зрозуміло найдорожчий зі способів.

Одним з найсучасніших та найефективніших рішень для розробки системи припливно-витяжної вентиляції є застосування системи з рекуперацією тепла (Рис.2). У сучасних установках вона дозволяє заощаджувати до 70 % тепла за рахунок підігріву припливного повітря,

що видаляється. Таким чином, можна обходитися без додаткового підігріву припливного повітря при великих морозах, а коли потужності рекуперації не вистачає, система підключає автоматичний нагрівач, але споживання електроенергії в такому разі багаторазово знижується.

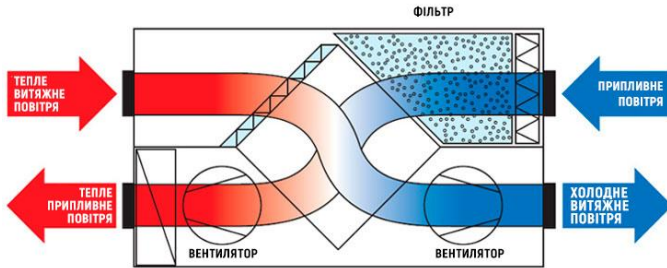


Рисунок 2 – Принцип роботи системи рекуператора

Можна зробити такі висновки:

– побутові кондиціонери мало підходять для використання у навіть невеликих офісах через велику кількість обмежень і невелику площу покриття;

– єдиним оптимальним варіантом каналної вентиляційної системи для офісу є припливно-витяжна вентиляція офісних приміщень. Припливно-витяжна система забезпечує як приплив, так і видалення повітря, налагоджуючи хороший повітрообмін. Цей вид можна назвати найпридатнішим для офісних приміщень. А завдяки застосуванню системи з рекуперацією тепла, можна обходитися без додаткового підігріву припливного повітря при великих морозах.

УДК 62-529

Трунов О. М.,
доктор. техн. наук, професор, професор кафедри АКІТ,
Скоройд М. Ю.,
аспірант кафедри АКІТ,
ЧНУ ім. Петра Могили м. Миколаїв Україна

ІНТЕГРАЦІЯ ПЛАТ ARDUINO ДЛЯ КЕРУВАННЯ ЕЛЕКТРОПРИВОДАМИ ЗА ДОПОМОГОЮ SIMULINK

Високомоментність та точність кутового повороту надає перевагу застосуванню у якості приводу кроковим двигунам з постійними

магнітами, особливо самарій-кобальтовими та фені-боровими. Другою рисою сучасного етапу розвитку галузі без екіпажного робото-технічного заміщення у сферах виробничого, побутового, військового та іншої сфери діяльності людини є поширеність плат Arduino. У зв'язку з цим актуальною стає проблема формування знань мов та навичок програмування цих плат і вибору способу втілення змодельованих систем у фізичну модель. Одним з ефективних способів формування таких знань і навичок є програмування та модельне налаштування у середовищі Matlab. Для більшості із останніх плат, що містять Wi-Fi канали зв'язку є не розв'язаною задачею. Інтеграція плат сімейства Arduino у програмний продукт Matlab за означеними аспектами з допомогою пакету Simulink Support Package for Arduino є метою поставленої роботи. Реалізація такої мети спрощує вимоги до знання мови програмування, а також дозволяє наочно спостерігати вхідні та вихідні осцилограми безпосередньо від плат Arduino, задаючи вхідні сигнали на виходи плат Arduino і тим самим набувати знання і навички з маніпуляцій руху крокового двигуна.

Об'єктом дослідження є інтеграція управління можливостей плат Arduino до програмного продукту Matlab.

Предметом дослідження є розробка моделі і методу інтегрування та налаштувань пакету Simulink Support Package for Arduino для комп'ютеризованої системи керування кроковими двигунами.

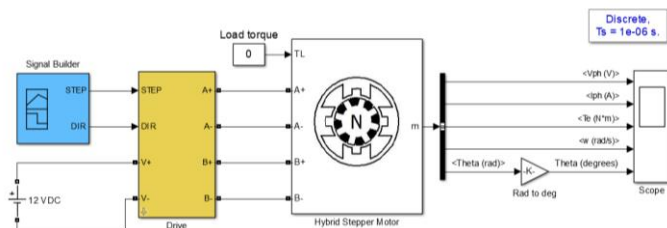


Рисунок 1 – Модель керування кроковим двигуном

Методи дослідження: проектування макету керування; проектування електричної схеми, друкованої плати розширення для Arduino Uno, підключення плат Arduino до програми Matlab, дослідження впливів для різних параметрів керування. Для розв'язання поставлених задач було обрано, як базову, модель керування кроковим двигуном, що запропоновано на рис. 1 представлено модель керування кроковим двигуном в Simulink. Її особливістю, що визначає

функціональні можливості є здатність для сформованої сукупності значень сигналів Step та Dig обернути ротор двигуна. Перший з двох типів сигналу визначає кут повороту, а другий напрям обертання. Також на схемі видно, що керування двигуном забезпечено за допомогою драйверу для керування кроковим двигуном, який живиться від напруги 12 В.

Для реалізації цієї схеми було створено монтажну плату, а для замість блоку Signal Builder було запропоновано використовувати мікроконтролер ArduinoUnoRev2 та блок драйверу A4988 для керування кроковим двигуном Nema17. Електрична монтажна плата була створена для формування знань і вмій з програмування та навичок з модельного налаштування у середовищі Matlab за допомогою програми Sprint-Layout. Блок схемне відображення процесу керування платою ArduinoUnoRev2, або інших з цього сімейства, за допомогою Simulink зображено на рисунку 2. Там відображено, що виходи сигналів з блоку Signal Builder подаються на внутрішні цифрові виходи 2,3 плати ArduinoUnoRev2. Внаслідок перебору значень із упорядкованої множини сигналів або із випадкової шляхом подачі цифрового сигналу була продемонстрована придатність системи до відгуку. Приклад складного керування двигуном протягом 25 секунд (відкладено по вісі абсис) для вхідного керуючого сигналу (відкладено по вісі ординат у одиницях вольт) представлено на рис. 3 верхнє поле сигнал 1. Існує три варіанти формування вхідного сигналу: завдання параметрів сигналу при заданому законі; завдання сукупності значень та третій вивід у поле сигналу відображень точок, які графічно переміщуються курсором на полі графіку, що забезпечує формування сигналу і форми графічно. Тут же на нижньому полі показано відповідну часову залежність сигнал 2. У третьому полі інтерфейсу представлено вербальне та цифрове відображення даних про вхідний та вихідний сигнали.

Таким чином, внаслідок моделювання у роботі продемонстровано, що існує можливість інтеграції плат Arduino, використовуючи пакет Simulink Support Package for Arduino для програми Matlab, що у свою чергу забезпечує формування знань і вмій з програмування та навичок з модельного налаштування у середовищі Matlab за допомогою програми Sprint-Layout. Завдяки отриманим результатам з моделювання було реалізовано макет для керування.

Використання такого методу керування двигуном через проміжний контролер може забезпечити задавання траєкторії руху, швидкості, задавати сторону руху, що знадобиться при складних задачах на точному рівні управління кутівими положеннями ротора двигуна.

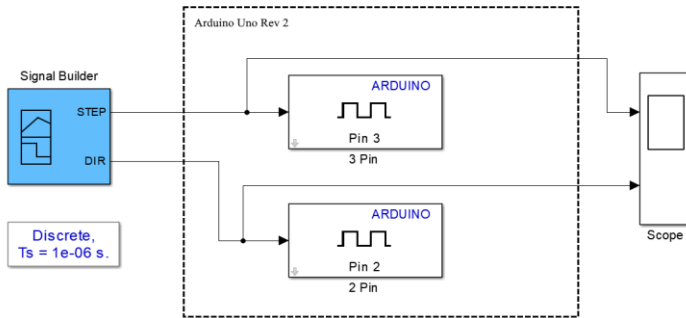


Рисунок 2 – Подача сигналів з цифрових виходів Arduino uno за допомогою програми Simulink

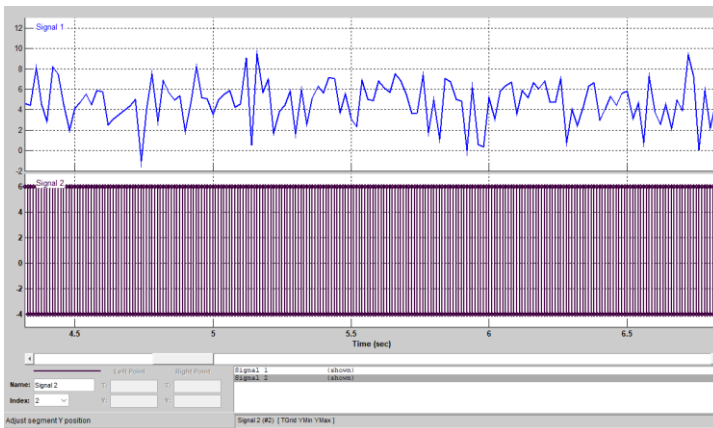


Рисунок 3 – Приклад задаючого вихідного сигналів для керування кроковим двигуном

Для реалізації цієї схеми було створено електричну монтажну плату, у якій замість блоку Signal Builder буде використовуватись мікроконтролер ArduinoUnoRev2 або інші такого сімейства, для блоку драйверу використовуємо драйвер для керування A4988, а кроковий двигун пета 17.

Електрична монтажна плата була створена за допомогою програми Sprint-Layout, вигляд цієї плати зображений на рис. 4. Керування платою ArduinoUnoRev2 за допомогою Simulink зображено на рисунку 3. Там показано, що виходи сигналів з блоку Signal Builder подаються на цифрові виходи 2,3 Arduino uno.

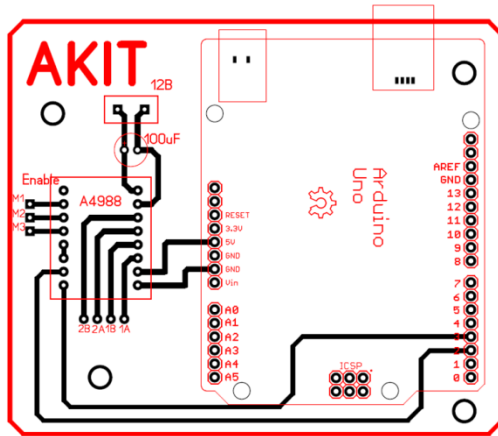


Рисунок 4 – Електрична монтажна плата керування кроковим двигуном

Таким чином, запропоноване подальше удосконалення моделі шляхом інтеграції плат сімейства Arduino у програмний продукт Matlab за допомогою пакету Simulink Support Package for Arduino спрощує вимоги до оператора із рівня знання мови програмування, а також дозволяє наочно спостерігати вхідні та вихідні осцилограми. Участь у безпосередній корекції вихідних сигналів плати Arduino, утворює середовище для ефективного формування знань і навичок з маніпуляцій рухом крокового двигуна. Така інтеграція пакету Simulink Support Package for Arduino у процес керування дозволяє оператору вивчати можливості управління і опанувувати розширенні функції плати Arduino разом із розширенням області застосування програмного продукту Matlab.

Висновки

1. Розроблена модель зовнішнього впливу, що здійснюється через мікроконтролер широкого розповсюдження у середовищі програмування з відкритим програмним кодом. Широкий перелік існуючих відкритих програм забезпечуватиме також успіх методу інтегрування та налаштувань пакету Simulink Support Package for Arduino для комп'ютеризованої системи керування кроковими двигунами у середовищі Matlab.

2. Розв'язана задача моделювання шляхом програмного аналітичного або точково-множинного або точково-графічного завдання вхідного сигналу, була апробована на основі, базової, моделі керування кроковим двигуном. Реалізація схеми під'єднання та моделювання

дозволить створювати електричну монтажну плату, яка спростить заміну блоку Signal Builder забезпечить надійне з'єднання і використання мікроконтролера ArduinoUnoRev2. Така модель керування кроковим двигуном в Simulink визначає розширення функціональних можливостей керування і дозволить опанувати розширенні функції плати Arduino разом із розширенням області застосування Matlab.

УДК 65.011.56

Сідєлєв М. І.,
канд. техн. наук, доцент, доцент кафедри АКІТ,
Кудінов Д. В.,
студент 4 курсу спеціальності АКІТ,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

ВИБІР SCADA-СИСТЕМИ В АВТОМАТИЗОВАНІЙ СИСТЕМІ КЕРУВАННЯ НАСОСНОЇ СТАНЦІЇ ТЕПЛОВОЇ МЕРЕЖІ

Насосні станції призначені для збільшення напору та зміни тиску у магістральних трубопроводах теплової мережі. Основна частина насосного агрегату – це електродвигун. Близько 70% витрат електроенергії припадає на частку електричних двигунів. Оскільки ціни на ресурси підвищуються з кожним роком, перед промисловими підприємствами та підприємствами житлово-комунального господарства постало завдання енергозбереження. Енергоефективність системи керування насосною станцією теплової мережі забезпечується, насамперед, використанням SCADA-системи.

SCADA-система повинна виконувати наступні функції:

1) Комунікаційні функції: між підсистемами та елементами SCADA; із системою верхнього рівня по основному та резервному каналам зв'язку.

2) Інформаційні функції: функції збору, первинної обробки та відображення інформації про перебіг технологічного процесу; функції формування аварійної та попереджувальної сигналізації; функції формування показників роботи технологічних об'єктів; функції реєстрації подій, аварійних повідомлень та значень параметрів.

3) Функції дистанційного керування: функції дистанційного керування за командою оператора; функції автоматизованого керування з алгоритмів SCADA; функції автоматичного захисту технологічного

обладнання – агрегатні, станційні, алгоритми захисту для всього трубопроводу.

4) Функції формування звітно-облікової документації: з урахування технічного стану технологічного обладнання; з урахування технічного стану програмно-технічних засобів; з урахування споживання паливно-енергетичних ресурсів; з урахування балансу та якості товарної продукції.

5) Функції налаштування компонентів SCADA включають функції базового програмного забезпечення SCADA та контролерів з налаштування та конфігурування засобів автоматизації.

Завдання, які вирішуються SCADA-системами, містять наступне: завдання збору та обробки інформації; завдання контролю та управління технологічним процесом та обладнанням насосної станції (НС) та/або павільйону; завдання відображення інформації; відображення технологічних схем; формування табличних форм відображення інформації; формування аварійних та попереджувальних сигналів; формування трендів за вимірюваними параметрами; завдання формування архівної інформації; завдання формування журналу подій та системного журналу; завдання контролю доступу до системи; тимчасові мітки з ПЛК та синхронізація за часом; онлайн перемикач мов з англійської на українську та назад.

Для прийняття рішення про використану SCADA-систему процес вибору поділимо на 2 етапи, які розглядаються експертами - спеціалістами з контрольно-вимірювальних приладів та автоматизації.

Етап 1 – вибір споживчих критеріїв. Практика показує, що здебільшого інженери називають такі властивості SCADA:

- надійність – основний критерій вибору основи системи автоматизації. Найважливіша надійність виконавчих модулів, тобто їхня здатність працювати тривалий час без збоїв;

- драйвери виконавчих об'єктів та контролерів – для узагальненої оцінки бажана підтримка якомога більшої кількості типів пристроїв, а з погляду користувача важлива підтримка конкретного обладнання, яке передбачається використовувати;

- Softlogic – система програмування контролерів, інтегрована зі SCADA – великий плюс системі. Контролер програмується тими самими засобами, як і АРМ, тобто у тих самих редакторах і тих самих стандартних мовами програмування, тому відпадає необхідність залучення додаткових програмістів;

- підтримка OPC – останнім часом набуває все більшого значення, хоча поки що далеко не всі виробники постачають OPC-сервери зі своїм обладнанням. Підтримка всіх типів взаємодії з пристроями (драйвери,

Softlogic та OPC) гарантує користувачеві вирішення проблем сумісності з обладнанням, у тому числі з погляду перспектив нарощування системи;

- налагоджувальні засоби – полегшують налагодження алгоритмів обробки даних у системі;

- продуктивність у реальному часі – параметр, критичний для автоматизації швидких процесів, управління агрегатами, систем з великим потоком даних для архівування, особливо для масштабних розподілених систем;

- гаряче резервування – необхідна функція для автоматизації життєво важливих ділянок виробництва та просто великих систем;

- графічні можливості – сучасна SCADA повинна мати редактора для створення мнемосхем з використанням тривимірної графіки та розвиненими можливостями анімації, підтримкою імпорту із зовнішніх графічних файлів та можливістю роботи з бібліотеками елементів;

- вбудовані бібліотеки – здатні прискорити створення графічних екранів із мнемосхемами за рахунок використання готових елементів;

- сучасні технології (WEB, GSM) – дедалі частіше стають основою реальних проєктів розподілених систем;

- побудова розподілених систем – клієнт-серверна архітектура, підтримка стандартних протоколів зв'язку та різних типів підключення, простота та прозорість налаштування обміну інформацією між окремими вузлами системи автоматизації – без усього цього немислима сучасна АСУТП;

- генерація звітів – сучасна SCADA надає гнучкі засоби аналізу архівних даних та підготовки звітів. Часто цей критерій виявляється найважливішим після надійності виконавчих модулів;

- документація – має бути повною та добре структурованою. Бажана наявність і друкованої та електронної версії;

- технічна підтримка – допоможе швидко знайти вихід зі скрутної ситуації. Витрати часу на пошук власного шляху вирішення проблеми, що виникла, навіть при високій кваліфікації користувача можуть бути незрівнянно більшими, ніж на дзвінок або e-mail у службу техпідтримки від виробника SCADA;

- простота освоєння – цей критерій дуже суб'єктивний і багато в чому залежить від рівня та характеру базових знань початківця користувача системи.

Етап 2 – оцінка значущості критеріїв, тобто оцінка важливості кожного з вищезгаданих критеріїв для конкретного проєкту, т.зв. «зважування» критеріїв.

Проведена оцінка за вищеперахованими критеріями наступних SCADA-систем: Genesis, IFIX, InTouch, Citect. Після врахування оцінки значущості критеріїв спеціалістами визначений пріоритет серед них: Genesis (компанія-розробник – Iconics). Таким чином представлений підхід оцінки SCADA-систем спрощує їх вибір в системах автоматизації на етапі проєктування.

УДК 796: 617.572-053.8-085

Трунов О. М.,
д-р техн. наук, професор, професор кафедри АКІТ,
Мальцев С. І.,
аспірант кафедри АКІТ,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

АНАЛІЗ ОСОБЛИВОСТЕЙ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ ПРОТЕЗІВ ВЕРХНІХ КІНЦІВОК

Потреба у антропоморфних пристроях, що частково функціонально відтворюють рухи верхніх кінцівок в останнє десятиріччя у світі катастрофічно зростає. Причиною тому є військові дії та їх наслідки: розриви не розмінованих мін, гранат та іншої не знешкодженої зброї. Консолідований розвиток робототехніки, очутливлених захватів, електромеханічних та пневматичних приводів разом із досягненнями електроніки та одноплатних мікроконтролерів і штучного інтелекту потребує перегляду концепції сучасного протезування. Однак, разом з таким переглядом для якісних змін функцій і можливостей протезів їх удосконаленню повинні передувати порівняльний аналіз характеристик оптимальних технічних рішень, що не можливо без побудови моделей та імітаційного моделювання.

В роботі на підставі аналізу досягнень проєктів, що містять приводи різної природи здійснено перегляд кінематичних схем п'яти палого протезу із чотирьох фаланговими пальцями.

Як базовий стандартний елемент розглядається зчленування двох фаланг, яке схематично подано на рис. 1.

Динаміку відносного руху такого зчленування розглянуто як динаміку відносного руху тіла під впливом розподілених сил, що модельовані у часі детермінованим та випадковим чином. Пропонується забезпечувати таке зчленування приводом, що реалізує відносний згин та фіксування положення фаланг. Крім того, передбачається, що зчленування забезпечується датчиком кута повороту, датчиком дотику

та датчиком зміщення уздовж поверхні фаланги. Запропоновано на підставі рис такої конструкції двох фаланг представляти конструкцію протезу як багатошарову модель нейромережевої структури. Кількість шарів у моделі пальця визначається кількістю фаланг пальця, а вихідні сигнали вихідного вузла збігаються до вузлів пружньо-паралельно зв'язаних фаланг. Такий фрагмент мережі представлено на рис. 2.

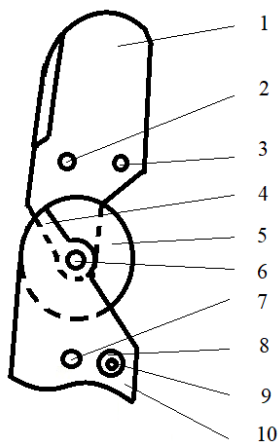


Рисунок 1 – Схематичне зображення кінематичної схеми зчленування фаланг пальця протезу.

1 – корпус фаланги; 2 – штифт тяги зворотнього руху фаланги;
 3 – штифт тяги відносного згину фаланги; 4 – корпус другої зчленованої фаланги; 5 – допоміжний диск; 6 – вісь шарніру; 7 – штифт тяги зворотнього руху фаланги; 8 – корпус підшипника ролика; 9 – вісь ролика

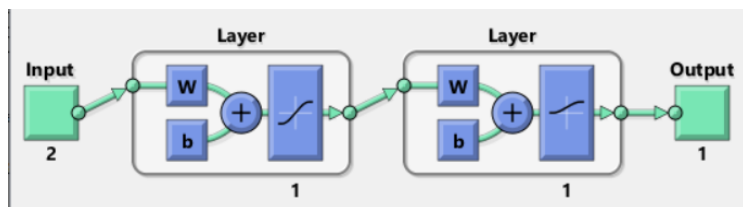


Рисунок 2 – Схематичне представлення фрагменту багатошарової мережі

На підставі модельних експериментів сформовані вимоги до додаткових сил які здатні гальмувати та фіксувати відносно положення фаланг зчленування та мінімальної структури датчиків зворотного зв'язку однієї фаланги і її взаємодії.

Висновки. Формування базового стандартного елементу зчленування, двох фаланг, дозволить здійснювати перегляд кінематичних схем п'ятипалоного протезу із чотирьох фаланговими пальцями, а застосування нейромережевої технології моделювати їх можливості.

УДК 004.67:004.032

Сухоставський Р. А.,
студент 4 курсу спеціальності АКІТ,
Щесюк О. В.,
канд. техн. наук, доцент, доцент кафедри АКІТ,
ЧНУ ім. Петра Могили, м. Миколаїв, Україна

СИСТЕМА АВТОМАТИЗОВАНОГО ДИСТАНЦІЙНОГО КОНТРОЛЮ МІКРОКЛІМАТУ ПРИМІЩЕННЯ ЗА ДОПОМОГОЮ «ХМАРНОГО СЕРВІСУ»

Актуальність

Мікроклімат – це один з найважливіших моментів, який дуже сильно впливає на можливість існування життя та бажаного рівня умов, що буде відповідати чи підвищувати стандарти для людей чи ресурсів, які працюють чи зберігаються у тому чи іншому приміщенні.

Станом на початок 2022 року було підраховано, що кількість офісів з працівниками приблизно 21570, а кількість складів з вмістом, які потребують спеціальних чи підвищених умов зберігання внаслідок початку війни зросла на 27 %.

Через відсутність достатнього рівня якості мікроклімату приблизно 3–5 % вмісту складів треба списувати щомісяця, що в перспективі виливається як в достатньо значимі втрати прибутку для компаній, а ефективність працівників через високий рівень CO₂ в приміщеннях та інші, також менш помітні параметри для людського ока, падає в середньому на 9–14 %. Саме тому дуже багато сфер для яких зараз досить гостро стоїть це питання, з яким достатньо добре може справлятися автоматизована система керування мікроклімату приміщення.

Встановлення таких систем у перспективі має велику актуальність через те, що умови праці та зберігання є одними з найважливіших в економічно-прогресивному нинішньому суспільстві.

Кінцевою метою матеріалу цієї доповіді є практичне застосування АСКМ для покращення показників рівня ефективності у роботі працівників та зберігання різного роду ресурсів в закритих приміщеннях, отримання можливості дистанційно аналізувати та

керувати процесами мікроклімату, які впливають на протікання вищезгаданих процесів.

Мікроклімат – це комплекс метеорологічних умов у приміщенні: температура, відносна вологість, повітрообмін, швидкість руху повітря, вміст у повітрі твердих частинок (пилу), наявність приємних запахів (ароматерапія) та ін.

Є декілька найвпливовіших параметрів, які ми розглянемо:

- Вуглекислий газ
- Вологість
- Температура повітря
- Рівень освітлення

Вуглекислий газ

Вуглекислий газ дуже негативно впливає на організм людини і є канцерогеном. Знаходження в приміщенні з високою концентрацією CO₂ може викликати слабкість, сонливість, головний біль, проблеми з концентрацією уваги або навіть негативні зміни в крові.

Дослідження показали, що при рівні CO₂ вище 1000 ppm, концентрація уваги знижувалася на 30 %. За рівня 1500 ppm – 79 % респондентів відчували втому, за 2000 ppm – 67 % опитаних зазначили, що не можуть зосередитися.

Вологість

Висока вологість збільшує віддачу тепла від тіла. Самопочуття погіршується, утворюється слабкість. Надлишок вологи може спричинити загострення серцево-судинних захворювань. У приміщенні може з'являтися грибок, цвіль, псується меблі.

Дуже сухе повітря прямо та опосередковано негативно впливає на наше здоров'я. Низька вологість при тривалому впливі на організм людини може призвести до пересихання очей, захворювань мозку, нирок, кровообігу, шкіри, очей, знизити стійкість організму до хронічних захворювань та алергій.

Температура повітря

Мікроклімат надає значний вплив на організм людини. Всі життєві процеси в організмі забезпечують енергією рухову активність, менша частина якої витрачається на виконання корисної роботи, а більшість перетворюється на теплову енергію. Це безперервне виділення тепла в довкілля, кількість якого змінюється від 85 (у стані спокою) до 500 Вт (при тяжкій фізичній роботі), забезпечує нормальний перебіг фізіологічних процесів.

Обов'язковою умовою життєдіяльності є повне відведення теплоти (теплопродукції), що виділяється організмом людини, у навколишнє середовище або захист організму людини від надмірної віддачі тепла у зовнішнє середовище. Порушення теплового балансу веде до перегріву

або переохолодження і, надалі - до порушення функціонального стану працівника, зниження та втрати працездатності, виникнення нещасних випадків, травм.

Зрештою, при перегріві можливі непритомність і смерть, при переохолодженні – замерзання. Менш виражені відхилення комбінацій параметрів мікроклімату, які забезпечують комфортний стан людини, сприяють продовженню тимчасової непрацездатності, виникнення професійної патології.

У важких випадках виникають виражені порушення з боку центральної нервової системи: затемнена свідомість, різке збудження, судоми, мимовільні рухи, галюцинації, марення. Дослідниками встановлено, що за температури повітря понад 30 °С працездатність людини починає падати.

Рішенням щодо покращення умов в процесі праці є АСКМ, які можуть забезпечити підтримку максимально комфортних для праці параметрів. Нижче приведено приклад функціональної схеми такого пристрою.

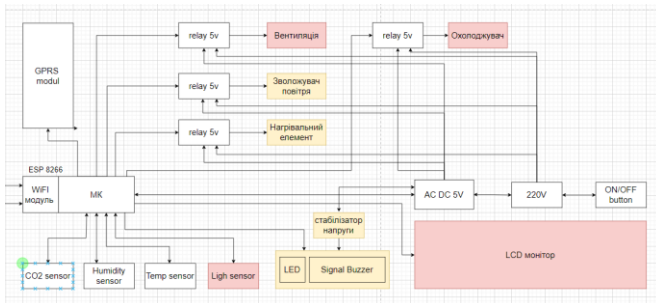


Рисунок 1 – Функціональна схема АСКМ

Рівень CO₂ (ppm) Вплив на людину:

- 300–400 ppm Оптиміальне значення.
- 400–600 ppm Норма для роботи в офісі без зниження продуктивності праці. На організм ніяк не впливає.
- До 600 ppm Норма для спальні, дитячого садка та школи.
- 600–1000 ppm Падає продуктивність праці, складніше концентрувати увагу та сприймати інформацію. З’являється головний біль.
- 1000–1500 ppm Людина стає млявою, падає концентрація уваги, складніше приймати рішення, збільшується кількість помилок у роботі. Виникає ризик розвитку хронічної втоми.
- Від 1500 р. Сильний головний біль. Зникає бажання працювати через втому.

Отже, підбиваючи підсумки, можна констатувати наступне: така річ як мікроклімат є дуже впливовою складовою на ефективність працівників офісу, виробництва тощо, і має великий економічний підтекст щодо кінцевого результату. Також не менш важливим це є у питанні зберігання продуктів на складах та сховищах.

Саме тому питання впливу мікроклімату є досить важливим, а рішення щодо керування мікрокліматом за допомогою АСКМ доволі актуальним, особливо в час критичного перепаду температур і швидких його змін.

УДК 004.45 (076.5)

Тогоєв О. Р.,
аспірант,

ЧНУ ім. Петра Могили, м. Миколаїв, Україна

СНІФФІНГ ІСТОРІЇ БРАУЗЕРА ЗА ДОПОМОГОЮ FAVICON

Сніфтери – це інструменти, що дозволяють перехоплювати, аналізувати та модернізувати всі запити, які через них проходять.

Запропонований метод використання сніфтера буде корисним для власників посадочних сторінок або інтернет-магазинів, щоб показати користувачеві релевантну рекламу, ґрунтуючись на знанні про те, які сайти конкурентів чи партнерів він вже відвідував.

Метод ґрунтується на підставі того, що `favicon.ico` відвідуваних користувачем сайтів, швидше за все, будуть закешовані для користувача і, відповідно будуть завантажуватися швидше, ніж для тих сайтів, які користувач ніколи не відвідував. Браузери завжди кешують `favicon.ico`, що обґрунтовує дієвість такого методу.

Нижче наведено код реалізації цього способу. За допомогою якого можна продемонструвати, що ви відвідували сайт `google.com`, але ніколи не були на сайті `chmnu.edu.ua`.

Цей код використовує константу `timeCachedLoadIco`, яка підбирається емпірично, тому точність результату не завжди стовідсоткова. Змінна `timeCachedLoadIco` - кількість мілісекунд, які було витрачено на завантаження картинки, яку слід вважати попаданням у кеш браузера. Результат виконання скрипта та результати завантаження фавіконок наведено на рис 1.

Ми можемо побачити, що швидкість завантаження `favicon` з `google.com` 150 мілісекунд – користувач відвідував `googl.com`, але не відвідував `chmnu.edu.ua` і має 610 мілісекунд відповідно.

```

var timeCachedLoadIco = 200;
var body = document.querySelector('body');

var testResults = [];
var testSites = [
  'chmnu.edu.ua', 'google.com'
];
testSites.forEach(test);
function test (each_site) {
  var start = new Date();
  var ico = new Image();
  img.src = 'https://' + each_site + '/favicon.ico';
  img.onload = function () {
    saveResult(each_site, start, new Date());
  }
  body.appendChild(ico);
}
function saveResult (each_site, start, end) {
  var timeLoadIco = end - start;
  testResults.push({
    host: each_site,
    start: start,
    end: end,
    diff: timeLoadIco,
    visited: timeLoadIco <= timeCachedLoadIco
  });
}

```

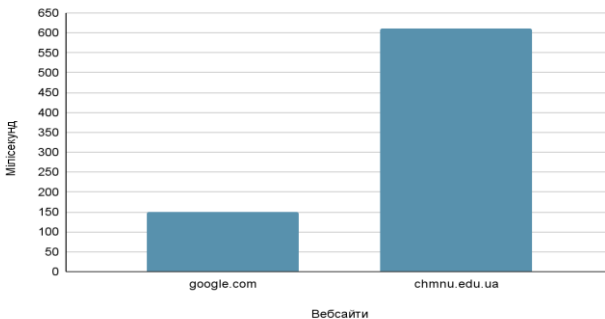


Рисунок 1 – Результати заміру швидкості завантаження favicon

Точніший метод повинен ґрунтуватися на підрахунку середнього значення між мінімальним і максимальним часом завантаження іконки, при цьому, одне з посилань має вести на іконку сайту, що гарантовано не існує в кеші. Тоді все, що менше цього середнього значення, можна вважати попаданням у кеш і, отже, свідчити, що користувач був на вказаному сайті.

Наведений метод дозволяє визначити вебсайти, які відвідував користувач, що досягається шляхом виміру часу завантаження іконки.

ЗМІСТ

Секція КОМП'ЮТЕРНІ НАУКИ

Підсекція: КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

<i>Баєв В. О., Пузирьов С. В.</i> Розробка моніторингової мережі на основі модулів ESP8266 Wemos	1
<i>Керекеслер М. О., Бурлаченко І. С.</i> Аналіз характеристик комплекуючих для безпілотних наземних транспортних засобів	4
<i>Костін О. А., Бурлаченко І. С.</i> Аналіз технологій машинного навчання для відстеження поз тіла людини	8
<i>Матюшок М.М., Бурлаченко І.С.</i> Системи навігації автономних безпілотних наземних транспортних засобів	12
<i>Мельник О. Д.</i> Переробка ПЕТ-пляшок у філамент та повторне використання відпрацьованого пластику для 3d-принтерів	16
<i>Пузирьов С. В., Кім В. М.</i> Система оповіщення на базі ESP-32 CAM, датчику руху та Telegram Bot Api	19
<i>Савінов В. Ю.</i> Захист листування за технологією PGP	21
<i>Степанов К. М., Крайник Я. М.</i> Система моделювання роботи декодування LDPC-кодів з використанням бібліотек JavaScript	25
<i>Солобуто Л. В., Полівода Д. В.</i> Система збору інформації для водія	26
<i>Сизоненко О. М., Христо О. І.</i> Система моніторингу процесу спікання дисперсних матеріалів електричним струмом	28
<i>Шевченко О. В., Крайник Я. М.</i> Система дослідження та моделювання роботи бібліотек стиснення	35

Секція ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ У РОЗВИТКУ СУСПІЛЬСТВА

Підсекція: ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ СИСТЕМИ

<i>Асєєв В. Д.</i> Соціальні мережі через метрики та огляд пз візуалізації графів	37
<i>Брагінець О. В., Воробйова А. І.</i> Застосування математичного пакета Maple до групового аналізу ДРЧП	40
<i>Донченко М. В.</i> Врахування надійності при оцінці безпеки систем	43
<i>Калініна І. О.</i> Порівняльний аналіз прогностичних моделей на основі дерев рішень	47
<i>Кулаковська І. В.</i> Аналіз розвитку слабкоструктурованих систем за допомогою методології когнітивних карт	52
<i>Сіденко Є. В., Кондратенко Г. В., Трухов А. С.</i> Машинне навчання для маршрутизації дрону	55
<i>Скакодуб О. С.</i> Автоматизація процесів керування безпілотними літальними апаратами	59
<i>Таранов М. О., Кулаковська І. В.</i> Навчальні заняття для студентів під час воєнного стану за допомогою відео з YouTube, конференцій Google Meet та системи електронного навчання Moodle	61
<i>Хомченко А. Н.</i> Ізоспектральні класи серендипових моделей бікубічних елементів	64

Секція ТЕХНІЧНІ НАУКИ

Підсекція: МОДЕЛІ, МЕТОДИ ТА ЗАСОБИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

<i>Антіпова К. О.</i> Огляд найбільш загальноживаних архітектур графових нейронних мереж	67
--	----

Боровльова С. Ю., Жлуктарьов А. А. Принципи використання Clean Architecture.....	72
Горбань Г. В., Використання засобів мови Python для аналізу даних результатів тренувань з академічного веслування.....	74
Гроза О. А., Кіяшко М. С., Боровльова С. Ю., Давиденко Є. О. Програмне забезпечення автоматизації тестування знань	77
Дворецький М. Л., Дворецька С. В. Шляхи реалізації наслідування при роботі із об'єктною моделлю із використанням реляційних баз даних	80
Кірей К. О. Модифікація продуктів бізнес-аналітики для обробки великих даних	86
Новожилів Я. В., Бойко А. П. Програмне забезпечення транспортної логістики	90
Стовманенко В. О., Давиденко Є. О. Розподілений інтерпретатор для обміну даними в інформаційно-аналітичних системах.....	93
Фісун М. Т., Кандиба І. О. Подальший розвиток методології іdef3 за рахунок додавання методів мережного планування	95
Швед А. В. Особливості розробки систем ситуаційного управління реального часу.....	96
Швед А. В., Фаленкова М. В., Давиденко Є. О. Інформаційно-аналітична система оцінювання сталого розвитку суспільства	100

**Підсекція: АВТОМАТИЗАЦІЯ ТА
КОМП'ЮТЕРНО-ІНЖЕНЕРНІ ТЕХНОЛОГІЇ**

Прищепов О. Ф., Щесюк О. В., Розганяєв Д. О., Яковішин А. Я. Методика оцінки рівня екологічної досконалості ДВЗ транспортного засобу.....	105
Кубов В. І., Димитров Ю. Ю., Руденко Д. Г., Курков М. Д. Досвід застосування хмарних технологій у системах автоматизації на базі GSM-модему та OWENCloud.....	108

Бєліков О. Є., Нєрега М. В., Шєнкевич В. М. Автоматичний дезінфектор для рук.....	110
Гєкова Т. В., Бєліков О. Є. Експериментальне дослідження кінематичних параметрів роботизованих систем руху в просторі	112
Паламарчук А. О., Кубов В. І., Димитров Ю. Ю. Використання gsm-модему для контролю та управління системами автоматичного керування	114
Калєвич Д. К., Шєнкевич В. М. Вибір системи вентиляції в офісному приміщенні	116
Трунов О. М., Скоройд М. Ю. Інтеграція плат Arduino для керування електроприводами за допомогою Simulink	119
Сідєлєв М. І., Кудінов Д.В. Вибір Scada-системи в автоматизованій системі керування насосної станції теплової мережі	124
Трунов О. М., Мальцев С. І. Аналіз особливостей нейромережєвих моделей протєзів верхніх кінцівок	127
Сухоставський Р. А., Щєсюк О. В. Система автоматизованого дистанційного контролю мікроклімату приміщення за допомогою «хмарного сервісу»	129
Тогоєв О. Р. Сніфінг історії браузеру за допомогою favicon.....	132

ДЛЯ НОТАТОК

ДЛЯ НОТАТОК

.

ДЛЯ НОТАТОК

.

Редактор *Н. Лебединцева*.
Технічний редактор, комп'ютерна верстка *Н. Кардаш*.

Підп. до друку 29.06.2022.
Формат 60x84¹/₁₆.
Гарнітура «Times New Roman».
Ум. друк. арк. 8,14. Обл.-вид. арк. 5,9.

Видавець і виготовлювач: ЧНУ ім. Петра Могили.
54003, м. Миколаїв, вул. 68 Десантників, 10.
Тел.: 8 (0512) 50-03-32, 8 (0512) 76-55-81, e-mail: rector@chmnu.edu.ua.
Свідоцтво суб'єкта видавничої справи ДК № 6124 від 05.04.2018.

АДРЕСА ОРГКОМІТЕТУ:

ОЛЬВІЙСЬКИЙ ФОРУМ – 2022:

СТРАТЕГІЇ КРАЇН ПРИЧОРНОМОРСЬКОГО РЕГІОНУ
В ГЕОПОЛІТИЧНОМУ ПРОСТОРИ

XVI Міжнародна наукова конференція

Чорноморський національний університет
імені Петра Могили,
вул. 68 Десантників, 10,
м. Миколаїв, 54003, Україна

Тел.: 8 (0512) 50–03–32,

8 (0512) 76–55–81,

8 (0512) 76–55–99,

факс: 50–00–69, 50–03–33,

E-mail: avi@chmnu.edu.ua, rector@chmnu.edu.ua

