

**РЕДАКЦІЙНА КОЛЕГІЯ:**

**Головний редактор:**

проф., д.т.н. Рудь В.Д. (м.Луцьк)

**Перший заступник головного редактора:**

доц., к.т.н. Герасимчук О.О. (м.Луцьк)

**Другий заступник головного редактора:**

доц., к.т.н. Лотиш В.В. (м.Луцьк)

доц., к.пед.н. Панасюк Н.Л. (м.Луцьк)

**Редактори-коректори:**

доц., к.т.н. Мельник К.В. (м.Луцьк)

Інженер-програміст Свиридчук К.А. (м.Луцьк)

**Члени редакційної колегії:**

проф., д.т.н. Божидарнік В.В. (м.Луцьк)

проф., д.т.н. Мазур М.П. (м.Луцьк)

проф., д.т.н. Пальчевський Б.О. (м.Луцьк)

проф., д.т.н. Мельник А.О. (м.Київ)

проф., д.пед.н. Горбатюк Р.М. (м.Тернопіль)

проф., д.пед.н. Поліщук Ю.Й. (м.Тернопіль)

проф., д.пед.н. Романишина Л.М. (м.Тернопіль)

проф., д.пед.н. Свистун В.І. (м.Київ)

проф., д.т.н. Сидорчук О.В. (м.Київ)

проф., д.т.н. Тарасенко В.П. (м.Київ)

доц., к.т.н. Гуменюк Л.О. (м.Луцьк)

доц., к.т.н. Кузнецов Р.М. (м.Луцьк)

доц., к.т.н. Пех П.А. (м.Луцьк)

доц., к.т.н. Повстяной О.Ю. (м.Луцьк)

доц., к.пед.н. Потапчук Л.М. (м.Луцьк)

доц., к.т.н. Решетило О.М. (м.Луцьк)

**Адреса редколегії:**

Луцький національний технічний університет,

кафедра комп'ютерної інженерії,

вул.Львівська 75, ауд.141

м.Луцьк, 43018

тел. (0332) 74-61-15

E-mail: cit@lntu.edu.ua,

ekaterinamelnik@gmail.com

**КОМП'ЮТЕРНО-ІНТЕГРОВАНІ**  
**ТЕХНОЛОГІЇ:**  
**ОСВІТА, НАУКА, ВИРОБНИЦТВО**

**№18 2015р.**

Журнал засновано у грудні 2010 р.  
Свідоцтво про реєстрацію КВ № 16705–5277 Р.  
Засновник: Луцький національний технічний університет  
Рекомендовано до друку Вченою радою  
Луцького національного технічного університету  
(протокол №8 засідання від 25.03.2014)  
Журнал рішенням президії ВАК України  
від 30 березня 2011  
№1-05/3 включено в перелік наукових фахових видань.  
Журнал має російський індекс наукового цитування  
(РІНЦ)

**ISSN:978-617-672-040-9**

## ЗМІСТ

### Інформатика та обчислювальна техніка

<b>Багнюк Н.В., Мельник В.М., Клеха О.В., Невідомський І.А.</b> Види DDoS-атак та алгоритм виявлення DDoS-атак типу flood-attack.	6
<b>Бортник К.Я., Прокопюк М.І.</b> Робота з com-портом у Windows та Android.	13
<b>Гринюк С.В., Міскевич О.І., Дацюк С.О.</b> Еволюція мікропроцесорної техніки на базі архітектури ARM та її вплив на розвиток обчислювальної техніки.	18
<b>Мельник К.В., Жигаревич О.К., Мороз С.М.</b> Дослідження та розробка групового органайзеру.	23
<b>Каганюк О.К., Панчук Р.Я.</b> Ринкова привабливість хмарного сервера транспортної логістики.	28
<b>Мельник В.М., Клімчук Н.О.</b> Основні концепції об'єктно-орієнтованих баз даних та їх проблематика.	33
<b>Мельник В.М.</b> Використання звичайних сокетів API для файлових систем типу Nadoop.	40
<b>Мельник В.М., Багнюк Н.В., Мельник К.В.</b> Алгоритми планування відомим обміном даних в операційних системах для мультисокетних багатоядерних серверів.	50
<b>Здолбіцька Н.В., Здолбіцький А.П., Найдюк Ю.Ю.</b> Програмне забезпечення для вибору режиму тренування на базі ОС Android.	61
<b>Пех П.А., Мельник В.М., Островський М.В.</b> C++Builder проект інтерполяції функцій многочленами Ньютона та Лагранжа.	66
<b>Пех П.А., Бондарчук В.О.</b> Дослідження ефективності синхронних та асинхронних технологій програмування у процесі розробки веб додатків.	73
<b>Пех П.А., Христинець Н.А., Прокопчук О.П.</b> C++Builder проект аналізу експериментальних даних, отриманих за симетричним композиційним ротатабельним уніформ планом другого порядку.	77
<b>Каганюк О.К., Бортник К.Я., Свиридюк В.В.</b> Аналіз аномальних станів трафіка комп'ютерної мережі на базі нейромереж.	83
<b>Здолбіцька Н.В., Здолбіцький А.П., Янчук Ю.В.</b> Дослідження пропускну здатності агрегованих інтернет-каналів.	87

### Автоматика та управління

<b>Андрущак І.Є., Шендрик Ю.В.</b> Реалізація методу прийняття рішення із застосуванням теорії втрат.	92
<b>Валяшек В.Б., Каплун А.В., Юзевич В.М.</b> Математичне та комп'ютерне моделювання фізичних характеристик матеріалу у вершині тріщини з урахуванням ефекту зміцнення.	97
<b>Гадай А.В.</b> Моделювання роботи синхронного двигуна для зменшення відхилень напруги в середовищі MATLAB/SIMULINK.	105
<b>Семенюк В.Я., Міскевич О.І., Горошко Д.О., Машевський М.В.</b> Оптимізація переміщень товарів при проектуванні виробничих систем.	110
<b>Смолянкін О.О., Маркіна Л.М., Захарчук О.Б.</b> Дослідження та регулювання теплових процесів в замкнутому об'єкті.	115
<b>Каганюк О.К.</b> Рівняння некерованого руху вуглевидобувних машин.	121
<b>Крадїнова Т.А.</b> Алгоритм розрахунку залишкового ресурсу залізобетонної балки.	127
<b>Літвінов К.А.</b> Математичні моделі ультразвукового рівнеміра з кільцевим рухом електричного сигналу.	132
<b>Проказа О.І.</b> Математичні моделі термодинамічних процесів в реакційній склянці апарату нейтралізації аміачної селітри.	142
<b>Шолом П.С., Жигаревич О.К., Рудник О.Ю.</b> Автоматизована інформаційна система управління відділеннями поштового зв'язку.	148
<b>Рябіченко А.В.</b> Дослідження термодинамічних процесів у людському організмі методом реологічних переходів.	155
<b>Сафонова С.О.</b> Метод визначення безпечного використання запобіжних клапанів.	160

<b>Фауре Е.В., Щерба А.І., Лавданський А.О.</b> Оцінка статистичних характеристик послідовності псевдовипадкових чисел, що породжена комбінаційним генератором.	165
<b>Здолбіцька Н.В., Здолбіцький А.П., Чміль О.М.</b> Розробка бібліотеки реального часу для анімації персонажа.	172
<b>Яремко С.А., Бевз С.В.</b> Побудова моделі рейтингового оцінювання результатів діяльності комерційних банків.	178

#### **Педагогічні науки**

<b>Лісковець С.М.</b> Методи дослідження окремих плоских трансцендентних кривих.	187
<b>Губаль Г. М.</b> Застосування кейс-методу у викладанні вищої математики та приклад науково-дослідного кейсу.	193

#### **Управління проектами**

<b>Сіваковська О.М.</b> Системні засади створення організаційно-технічних систем автоматизованого управління проектами та програмами	198
--	-----

УДК 371.13.001.76

Багнюк Н.В. к.т.н. доц., Мельник В.М.к.ф.-м.н.доц, Клеха О.В., Невідомський І.А.  
Луцький національний технічний університет

## ВИДИ DDoS-АТАК ТА АЛГОРИТМ ВИЯВЛЕННЯ DDoS-АТАК ТИПУ FLOOD-АТАК

**Багнюк Н.В., Мельник В.М., Клеха О.В., Невідомський І.А. Види DDoS-атак та алгоритм виявлення DDoS-атак типу flood-атак.** У статті розкрито існуючі види DDoS-атак та їх суть, а також деякі методи боротьби та протидії цим атакам з метою забезпечення надійного та постійного функціонування комп'ютерної мережі та її компонентів. Також запропоновано алгоритм протидії DDoS-атакам типу flood-атаки.

**Ключові слова:** DDoS-атака, види DDoS-атак, flood-атаки, функціонування комп'ютерної мережі.

**Багнюк Н.В., Мельник В.М., Клеха О.В., Невідомський І.А. Виды DDoS-атак и алгоритм обнаружения DDoS-атак типа flood-атак.** В статье раскрыты существующие виды DDoS-атак и их суть, а также некоторые методы борьбы и противодействия этим атакам с целью обеспечения надежного и постоянного функционирования компьютерной сети и ее компонентов. Также предложен алгоритм противодействия DDoS-атакам типа flood-атаки.

**Ключевые слова:** DDoS-атака, виды DDoS-атак, flood-атаки, функционирования компьютерной сети.

**Bagnyuk N.V., Melnyk V., Klekha A.V., Nevidomskyy I.A. Types of DDoS-attacks and detection algorithm of such DDoS-attacks as flood-attack.** The article considers the existing types of DDoS-attacks and their essence, and some methods of struggle and counter these attacks in order to ensure a reliable and continuous operation of a computer network and its components. Also, the counteractive algorithm DDoS-attacks flood-type attack.

**Keywords:** DDoS-attack, types of DDoS-attacks, flood-attack, computer network functioning.

### Постановка наукової проблеми.

Розподілена атака на відмову в обслуговуванні – це реальна і зростаюча загроза, з якою стикаються компанії в усьому світі. Ці атаки реалізуються великою кількістю програмних агентів, розміщених на хостах, які зловмисник скомпрометував раніше. Реалізація цих атак може призвести не тільки до виходу з ладу окремих хостів і служб, а й повністю або тимчасово зупинити роботу мережі. У зв'язку з критичністю і нетривіальністю даного класу атак, побудова ефективних засобів захисту від них являє собою складну науково-технічну проблему. На рівні маршрутизаторів захист від DDoS-атак вже досить успішно реалізували компанії Cisco Systems. Але в цілому проблема DDoS-атак на сьогоднішній день як і раніше дуже гостро стоїть для більшості компаній.

### Аналіз досліджень.

В обчислювальній техніці, атаки на відмову в обслуговуванні (DoS-атаки) або розподілені атаки на відмову в обслуговуванні (DDoS-атака) є спробою зробити машини або мережевий ресурс недоступним для можливих користувачів. Мотиви і цілі з DoS-атаки можуть відрізнятися, але в загальному випадку складаються із зусиль одного або декількох людей тимчасово або на невизначений термін перервати або призупинити надання послуг мережевих послуг.

Кожного дня по всьому світі відбуваються DDoS-атаки різні за масштабом (Рис.1) [7].



Рис.1. Цифрова карта DDoS-атак

Зазвичай використовуються такі терміни:

- Intruder: також називається нападник;
- Master: також називається оператором;
- Daemon: також званий агентом;
- Victim: завжди є жертвою.

Симптоми відмови в обслуговуванні згідно з US-CERT включають в себе:

- незвично низьку продуктивність мережі (повільне відкриття файлів або доступ до ресурсів);
- відсутність конкретного ресурсу;
- неможливість отримати доступ до будь-якого ресурсу;
- значне збільшення числа спам-листів (цей тип DoS-атаки включає в себе електронні листи з шкідливим вмістом);
- відключення бездротової мережі або доступу до мережі;
- «hit offline», тобто ціллю є позбавити вас підключення до мережі.

Відмови в обслуговуванні можуть також призвести до проблем у «гілці» мережі, в якій знаходиться фактична жертва нападників. Наприклад, пропускна здатність маршрутизатора між Інтернетом та локальною мережею може споживатися атакою, що призводить до збитків не тільки для потенційної жертви, але й для всієї мережі. Якщо атака здійснюється на досить великому масштабі – цілі географічні регіони, то через неправильно налаштоване обладнання мережевої інфраструктури, про що зловмисник міг і не знати, може бути порушена робота всієї мережі.

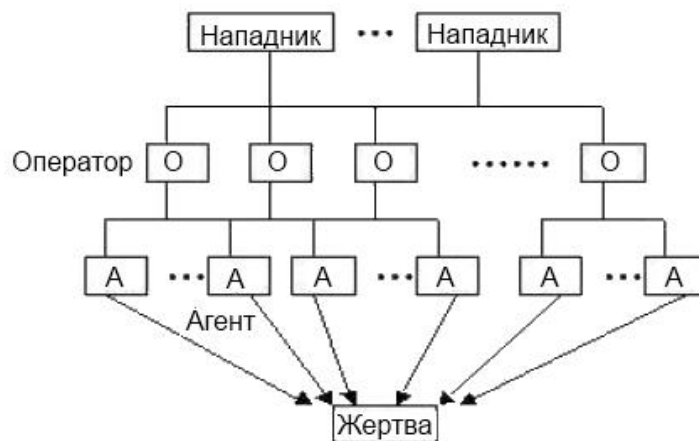


Рис. 2. Концептуальна схема DDoS-атаки

Одними з найпрогресивніших досліджень в області захисту від DDoS-атак є роботи професорів Peng T. Ong, Jaydip Sen та доктора Ashish Gupta. Структура атаки майже завжди є дуже складною, що не дозволяє в багатьох випадках відстежити нападника. Зв'язок між майстром і демонами може бути непомітним, так що стає важко знайти головний комп'ютер. Хоча деякі докази можуть існувати на одному або декількох комп'ютерах в мережі DDoS з місцем розташування майстра. Демони, як правило, автоматизовані так, що вони не є необхідними для постійного діалогу, котрий відбудеться між майстром і рештою мережі. Насправді, такі методи, які зазвичай використовуються, свідомо маскують особу і місцезнаходження господаря в мережі DDoS. Ці методи роблять процес аналізу атаки, блокування атакуючого трафіку і відстежування його до джерела надзвичайно важким.

У більшості випадків, системні адміністратори заражених систем навіть не знають, що демони були встановлені в системі. Навіть якщо вони знайшли і знищили програмне забезпечення DDoS, вони не можуть допомогти іншим користувачам визначити, чи є десь в системі ще розміщене подібне програмне забезпечення. Популярними системами для експлуатації є Web-сервери, електронна пошта, інші сервери, так як ці системи можуть мати велику кількість відкритих портів, великий обсяг трафіку, і навряд чи будуть швидко виведені з ладу, навіть при атаці на них.

DDoS-атаки завжди пов'язані з низкою систем. Типовий сценарій DDoS-атаки може відбуватись приблизно за наступними кроками:

- зловмисник знаходить одну або декілька систем в Інтернеті, які можна скомпрометувати і експлуатувати. Це зазвичай здійснюється за допомогою вкраденого облікового запису в системі з великим числом користувачів через неуважних адміністраторів чи користувачів, переважно із з'єднанням з високою пропускнуою здатністю до Інтернету;

- в зломану систему завантажуються будь-яка кількість таких інструментів, як сканери, детектори операційної системи, руткіти, а також програми DoS/DDoS. Ця система стає майстром DDoS. Майстер за допомогою програмного забезпечення дозволяє знайти ряд інших систем, які можна експлуатувати. Зловмисник сканує великі діапазони IP мережевих адресних блоків, щоб знайти системи, що мають вразливі місця в безпеці. Ця початкова фаза масового вторгнення використовує автоматизовані засоби, щоб віддалено зламати кілька сотень чи кілька тисяч хостів і встановити DDoS агенти в цих системах. Автоматизовані інструменти, що використовуються не є частиною інструментарію DDoS, але є способом обміну всередині груп злочинних хакерів. Ці зламані системи є початковими жертвами нападу DDoS. Згодом ці системи будуть експлуатуватись демонами DDoS, які здійснюватимуть фактичний напад;

- зловмисник має список систем, якими він може керувати, і в яких системах є демони DDoS. Фактичний напад відбувається, коли зловмисник запускає програму в головній системі, що спілкується з демонами DDoS, щоб почати атаку.

В таблиці 1 показані найпопулярніші та найвідоміші засоби для DDoS-атаки та їх основні специфікації.

**Таблиця 1. Засоби DDoS-атаки та їх специфікації**

Засоби DDoS-атаки	Зв'язок Intruder-to-master	Зв'язок Master-to-daemon	Зв'язок Daemon-to-master
Trinoo	27665/tcp	27444/udp	31335/udp
TFN	ICMP Echo/Echo Reply	ICMP Echo Reply	ICMP Echo/Echo Reply
Stacheldraht	16660/tcp	65000/tcp	ICMP EchoReply
Trinity	6667/tcp	6667/tcp (також 33270/tcp)	
Shaft	20432/tcp	18753/udp	20433/udp

Атака "відмова в обслуговуванні" характеризується явною спробою нападників відключити законних користувачів мережі або ресурсу від використання його доступних сервісів. Є дві основних форми DoS атак: ті, які є сервісами злому (crash services) і сервіси флуду (flood services).

Атака DoS може бути здійснена у ряді напрямків згідно з професором. П'ятьма основними типами атак є:

- споживання обчислювальних ресурсів, таких як пропускна здатність, дисковий простір або процесорний час;
- перешкоджання доступу до інформації про конфігурацію мережі, наприклад, інформацію про маршрутизацію;
- перешкоджання доступу до інформації про стан, наприклад, небажане скидання сеансів TCP;
- перешкоджання доступу до фізичних компонентів мережі;
- перешкоджання масової комунікації між передбачуваними користувачами і жертвами, так що вони вже не можуть спілкуватися.

Атака DoS може включати в себе використання шкідливих програм, призначених для:

- використання процесора на максимум, запобігаючи будь-якій новій роботі машини;
- помилки тригера в мікрокодах машини;
- помилки тригера в послідовності інструкцій, з тим, щоб змусити комп'ютер працювати в нестабільному стані або припиняти його роботу;

- використовувати помилки в операційній системі, тобто використовувати всі наявні засоби, щоб реальна робота не виконувалась або відбувся збій самої системи;
- збій самої операційної системи.

Рисунок 3 ілюструє тип широкосмугової атаки, котрий називається відбиваючою розподіленою відмовою в обслуговуванні (DRDOS-атакою). Метою DRDOS-атаки є приховати джерела трафіку атаки за допомогою третіх осіб (маршрутизаторів або веб-серверів) для передачі трафіку атаки до жертви. Ці безневинні треті особи називаються відбивачами. Будь-яка машина, яка відповідає на вхідний пакет може стати потенційним відбивачем. Напад DRDOS складається з трьох етапів. Перший етап являє собою типову DDoS-атаку, де нападники відправляють велику кількість пакетів до хоста жертви. Тим не менш, на другому етапі, після того як нападник отримав контроль над певною кількістю «зомбі», замість інструктажу «зомбі» для відправки трафіку атаки жертвам безпосередньо, «зомбі» наказано вислати третім особам фальшиві пакети з IP-адресом жертви в якості джерела IP-адресу. На третьому етапі, треті особи будуть надсилати відповідь до жертви, яка являє собою атаку DDoS. У порівнянні з традиційною DDoS-атакою, трафік від нападу DRDOS збільшений за допомогою третіх осіб. Це робить напад більш ширшим а, отже, і процес зупинення атаки буде більш важким. Крім того, джерела IP-адрес атаки є безневинними третіми особами. Це робить процес простеження джерела атаки вкрай складним. Нарешті, DRDOS атаки мають здатність посилювати трафік атаки, що робить атаку ще більш потужною.

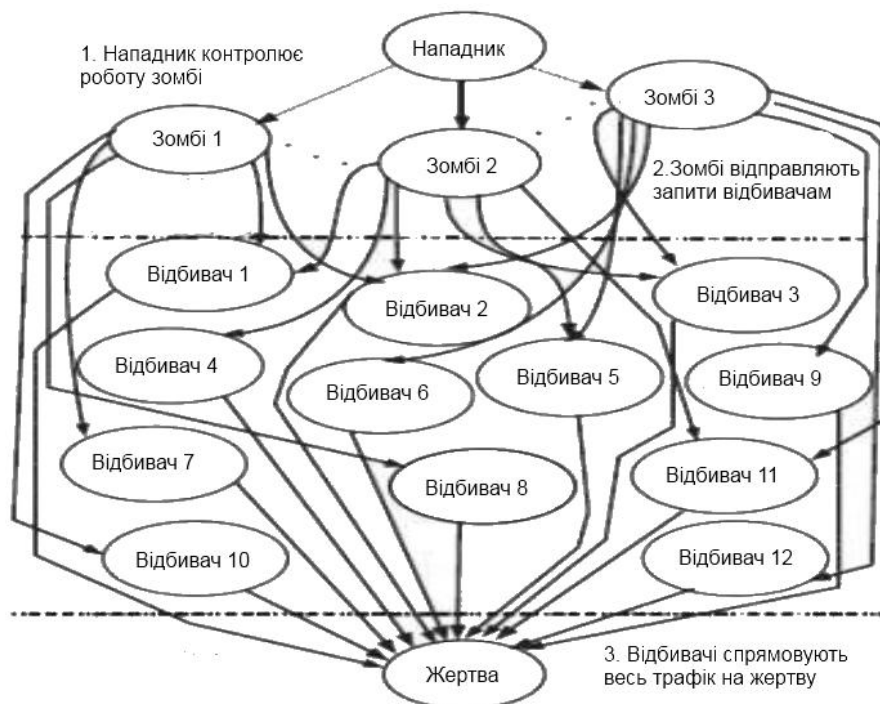


Рис. 3. DR-DoS-атака

HTTP flood відноситься до атаки, що бомбардує веб-сервери HTTP-запитами. HTTP flood спільна риса в більшості ботнетів програмного забезпечення. Для відправлення запиту HTTP потрібне підключення TCP, який вимагає справжнього IP-адресу. Зловмисники можуть підставити його за допомогою IP-адреси бота. Крім того, зловмисники можуть сформувати HTTP запити різними способами для того, щоб або збільшити силу атаки або уникнути виявлення. Наприклад, зловмисник може доручити ботнету відправляти HTTP-запити на скачування великих файлів від цілі. Так ціль повинна зчитати файл з жорсткого диска, зберегти його в пам'яті, завантажити його в пакети, а потім відправити пакети назад в ботнет. Таким чином, простий запит HTTP може спричинити значні витрати ресурсів процесору, пам'яті, пристроїв вводу/виводу і вихідних інтернет-з'єднань.

SYN flood-атака використовує уразливість TCP, а саме трьох ступінчатий запит, тому сервер повинен виділити велику структуру даних для всіх вхідних SYN пакетів, незалежно від його достовірності. Під час SYN flood-атаки, атакуючий посилає SYN пакети з вихідними IP-адресами, які не існують або не використовуються. Коли сервер заносить інформацію запиту в стек

пам'яті, він буде чекати підтвердження від клієнта, який відправив запит. У той час як запит очікує підтвердження, він буде залишатися в стеку пам'яті. Оскільки IP-адреса джерела, використовуваного в ході SYN flood-атаки може виявитися помилковою, сервер не отримує пакета з підтвердженням запитів. Кожне напіввідкрите з'єднання буде залишатися в стеку пам'яті до закінчення часу запиту. Це призводить стек пам'яті до переповнення. Таким чином, ніякі запити, в тому числі потрібні, не можуть бути оброблені і послуги системи будуть відключені. SYN flood-атака залишаються одним з найбільш потужних методів флуду.

Smurf-атака типу ICMP flood, де зловмисники використовують ICMP-пакети echo-запитів, спрямованих до широкомовних IP адресів з віддалених місць, щоб згенерувати відмову в обслуговуванні. Є три особи таких нападів: зловмисник, посередники та жертви. По-перше, зловмисник посилає один ICMP echo-запит пакет на широкомовний адрес мережі і запит направляється на всі вузли в межах посередницької мережі. По-друге, всі хости в межах посередницької мережі, відправляють ICMP echo-відповіді, що йдуть на адрес жертви. Рішення проти Smurf-атаки включає відключення IP спрямованих на послуги широкомовної передачі в посередницькій мережі. В даний час, Smurf-атаки досить рідкісні в Інтернеті, бо захиститися від таких атак не складно.

#### Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Загальна схема протидії DDoS-атакам представлена на рис.4.

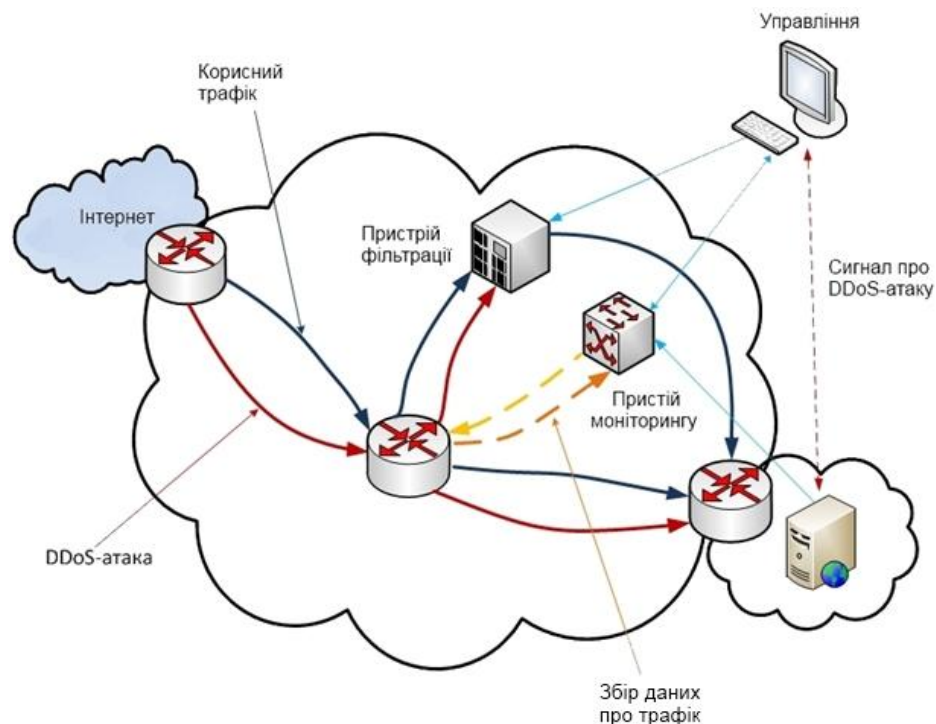


Рис. 4. Схема протидії DDoS-атаки

Технічна реалізація даного рішення передбачає наявність у мережі двох додаткових пристроїв, один з яких здійснює моніторинг вхідного трафіку і виявляє чи відбувається DDoS-атака, а другий фільтрує(очищає) зовнішній трафік.

У нормальному режимі роботи дані пристрої не повинні чинити жодного впливу на трафік що проходить. У разі ж атаки пристрій «очищення» затримує трафік, ідентифікований як DDoS-пакети, не допускаючи її потрапляння у відносно вузькосмугові клієнтські канали і на клієнтські ресурси, тим самим не перериваючи надання клієнту основної послуги.

Пристрій моніторингу на сервері виконує чотири дії з метою виявлення атаки DDoS та джерела атаки. Ці чотири дії виконуються послідовно в тому ж порядку, як вони вказані:

- 1) виявлення нападу;
- 2) ідентифікації джерел атак;
- 3) зупинення трафіку, що надходить від джерел атаки;
- 4) тестування, чи атаку успішно зірвано.



На рисунку 5 представлена блок схема виявлення та зупинення DDoS-атаки. З самого початку перевіряється чи йде збільшення числа з'єднань до сервера, ніж це відбувається зазвичай. Якщо ж все-таки є їх перевищення, то це є підставою для перевірки чи йде атака на наш сервер.

Перш за все перевіряється чи знаходяться IP адреси в списку виключень, тобто доступ до мережі вони не повинні мати.

Після цього підраховується кількість з'єднань з кожного IP адресу до нашого сервера. Коли їх більше, ніж  $n$ , то це є підставою вважати, що йде DDoS-атака. Щоб припинити атаку варто заблокувати дані IP адреси на 5 хвилин. За цей час буде зупинено атаку.

Дана схема є дієвою для всіх flood-атак. Єдина різниця що для кожного типу атаки буде відрізнятися число  $n$ .

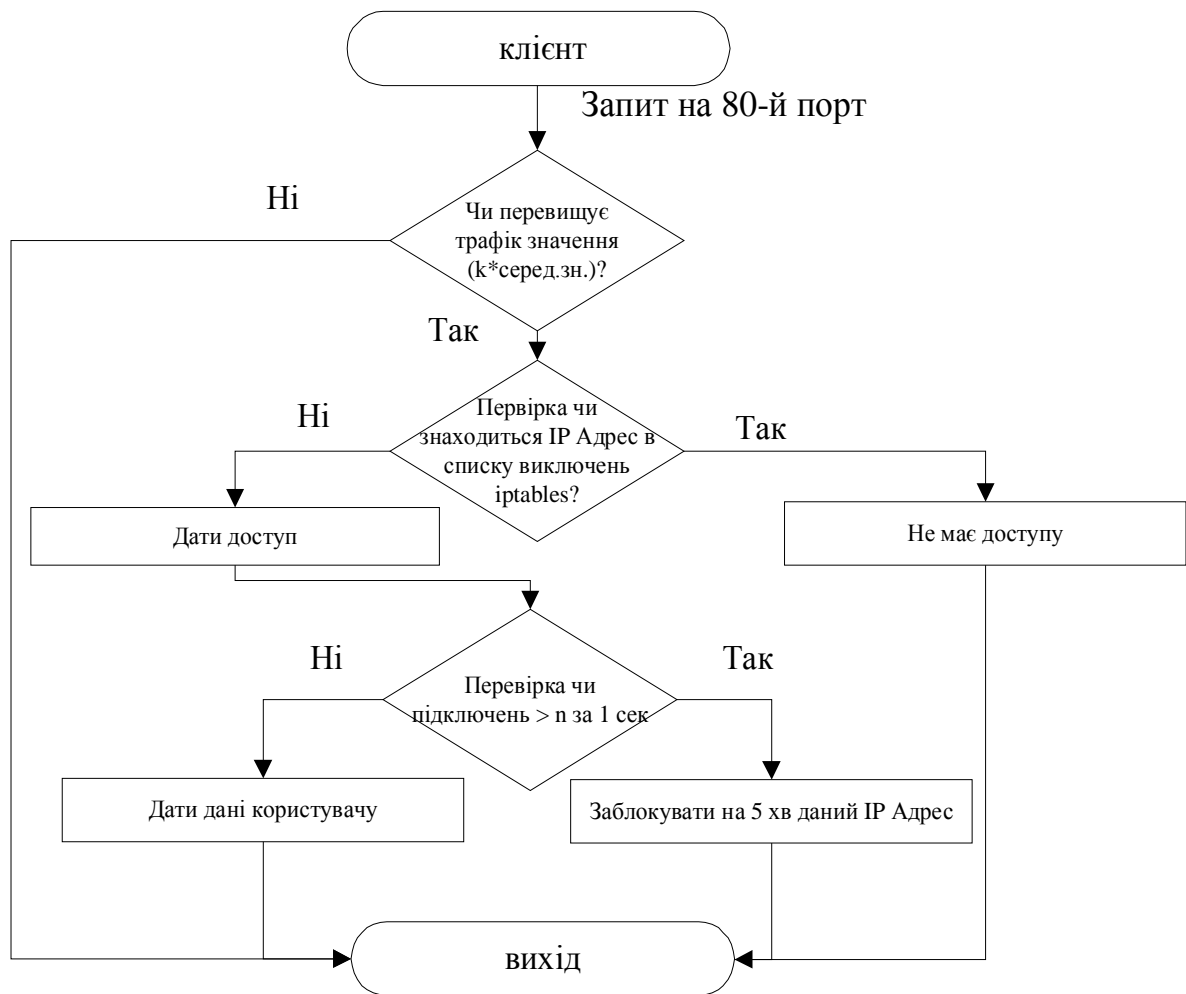


Рис. 5 – Блок схема виявлення та зупинення DDoS-атаки

Коли є підозра, що йде DDoS-атака всі дані про з'єднання записуємо в окремий текстовий файл. Ключовими полями в файлі, котрі нас цікавлять є час та вхідний IP-адрес з'єднання. Ці два поля за допомогою циклу в Python поміщаємо до словника (dictionary):

```

for i in range(len_line-1): //len_line – кількість рядків в файлі
    item = my_file.readline()
    item = string.split(item, " ")
    d[item[2]]=item[3] // d – dictionary, в котрий поміщаємо наші дані
    l+= [item[2]]
    
```

Тепер опрацюємо dictionary в відповідності чи є перевищення кількості з'єднань число  $n$  за 1 секунду за допомогою дано коду:

```

for i in range (t+1):
    l2=[]
    
```

```
for item in l:  
    if (float(item) > i) and (float(item) < (i+1)):  
        l2.append(d[item])  
l3+= [x for x in set(l2) if l2.count(x)>n]  
l3 = [e for i,e in enumerate(l3) if e not in l3[:i]]
```

В списку l3 містяться всі IP-адреси, з котрих йде DDoS-атака. Тому всі вони блокуються на 5 хвилин в iptables.

**Висновки та перспективи подальшого дослідження.** Таким чином, DDoS-атаки є реальною загрозою для функціонування будь-якої мережевої комп'ютерної системи. Хоча часто важко виявити і реагувати на розумно сплановані і підготовлені атаки. Після рекомендації, що містяться в цій статті буде легше захистити жертву, максимально знизивши при цьому вплив потенційно небезпечних атак.

1. Chen, E.Y. Detecting DoS Attacks on SIP Systems. In: Proceedings of the 1st IEEE Workshop on VoIP Management and Security, pp 53 – 58.
2. Forristal, J. Fireproofing against DoS Attacks. URL: <http://www.networkcomputing.com/1225/1225f3.html>, Network Computing.
3. Masure, K., Imai, H. Protection of Authenticated Key-Agreement Protocol against a Denial-of-Service Attack. In Proceedings of 1998 International Symposium on Information Theory and its Applications (ISITA'98), pp. 466 – 470, Oct. 1998.
4. Park, K., Lee, H. "On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internet. In Proceedings of ACM SIGCOMM'01, San Diego, CA, August 2001, pp. 15 – 26.
5. Park, K., Lee, H. On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack. In Proceedings of IEEE INFOCOM'01, Anchorage, Alaska, 2001, pp. 319 – 347.
6. Paxson, V. An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks. ACM Computing Communication Review, Vol. 31, No. 3, pp. 38 -47, 2001.
7. Paxson, V. An Analysis of Using Reflectors for Distributed Denial-of-Service Attacks. ACM Computing Communication Review, Vol. 31, No. 3, pp. 38 -47, 2001.
8. Peng, T., Leckie, C., Ramamohanarao, K. Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems. ACM Computing Surveys, Vol. 39, No. 1, April 2007.
9. Ramanathan, A.: WesDes: A Tool for Distributed Denial of Service Attack Detection. Thesis at Texas A&M University, August 2002.
10. Sen, J. A Novel Mechanism for Detection of Distributed Denial of Service Attacks. In Proceedings of the 1st International Conference on Computer Science and Information Technology (CCSIT 2011) – 2011, pp. 247 – 257
11. Top daily DDoS attacks worldwide. URL: <http://www.digitalattackmap.com>.

УДК 004.94

Бортник К.Я. к.т.н., Прокопюк М.І.

Луцький національний технічний університет

## РОБОТА З СОМ-ПОРТОМ У WINDOWS ТА ANDROID

**Бортник К.Я., Прокопюк М.І. Робота з сом-портом у Windows та android.** У даній статті висвітлено актуальне на даний час питання роботи з СОМ-портами у середовищах Windows та Android. Запропоновано найпоширеніші, прості та найефективніші способи для роботи з СОМ-портами.

**Ключові слова:** алгоритм, СОМ-порт, Android, файл.

**Бортник К.Я., Прокопюк М.І. Работа с сом-портом в Windows и android.** В данной статье рассматриваются актуальный в настоящее время вопрос работы с СОМ-портами в средах Windows и Android. Предложено распространенные, простые и эффективные способы для работы с СОМ-портами.

**Ключевые слова:** алгоритм, СОМ-порт, Android, файл

**Bortnyk K., Prokopyuk M. Working with com-port in Windows and android** In this article highlights the actual issue currently working with COM-ports in Windows and Android. A most common, simple and most effective ways to work with COM-ports.

**Keywords:** algorithm, COM-port, Android, file.

Розробники високорівневих мов програмування, очевидно, вважають прийом/передачу даних по протоколу rs232 через комунікаційний порт екзотичною процедурою: мовляв, пересічному користувачеві не потрібно, а непересічний - розбереться самостійно. Тому ні в turbo pascal, ні в delphi немає штатних засобів обміну даними таким способом. Проте останнім часом, особливо у зв'язку з поширенням мікропроцесорних пристроїв, таке завдання постає в аматорських програмах все частіше - в силу простоти і дешевизни реалізації послідовним портом обладнано багато наукових та інженерних приладів, різноманітні датчики і вимірники.

Простота апаратного виконання (для звичайного двостороннього зв'язку потрібно всього три дроти) асинхронного комунікаційного порта все таки веде до деякого ускладнення програмного забезпечення. Можна, звичайно, спробувати послати байт на пристрій com1 засобами dos, але успіх навряд буде досягнутий - як мінімум, треба спочатку налаштувати швидкість обміну. Тому для dos-програм це робиться засобами bios або прямим програмуванням порту "по залізу". В Windows, на щастя, є відповідні функції арі.

З послідовними і паралельними портами в Windows працюють як з файлами, тому для відкриття порта використовується функція CreateFile. Її прототип виглядає ось так:

```
HANDLE CreateFile(  
    LPCTSTR                lpFileName,  
    DWORD                  dwDesiredAccess,  
    DWORD                  dwShareMode,  
    LPSECURITY_ATTRIBUTES lpSecurityAttributes,  
    DWORD                  dwCreationDistribution,  
    DWORD                  dwFlagsAndAttributes,  
    HANDLE                 hTemplateFile  
);
```

Щоб відкрити СОМ-порт ви повинні виконати цю функцію в коді своєї програми, з заданими вхідними параметрами. Результатом роботи цієї функції буде 32-бітне число handle, по якому ви зможете звертатися до створеного функцією програмному об'єкту зв'язаного з вибраним СОМ-портом.

Параметры функции CreateFile:

- lpFileName - ім'я СОМ-порта. Може приймати значення: "COM1", "COM2", "COM3", "COM4", "COM5", "COM6", "COM7", "COM8", "COM9", якщо більше однієї цифри, то у форматі "\\.\COM47".

- `dwDesiredAccess` - режим доступу до файлу. Це чотирьохбайтове число, яке задає різні режими доступу до файлу. Нас цікавить тільки режим читання і запис, цей режим задається числом: `C0000000hex` в C можна замість числа записати константу з ім'ям `"GENERIC_READ | GENERIC_WRITE"`.
- `dwShareMode` - режим спільного доступу. COM-порти ПК не підтримують спільний доступ, тільки одна програма може відкрити порт. Тому цей параметр повинен бути рівний 0 (режим заборонений).
- `lpSecurityAttributes` - атрибути захисту файлу. Для COM-портів не використовується тому завжди рівні 0 (`"NULL"`).
- `dwCreationDistribution` - управління режимом автостворення файлу. Це чотирьохбайтове число, яке для COM портів завжди повинно бути `00000003hex` (`"OPEN_EXISTING"`).
- `dwFlagsAndAttributes` - задає атрибути створюваного файлу. Це чотирьохбайтове число, яке для COM портів завжди повинно бути 0 (`"NULL"`).
- `hTemplateFile` - описатель файлу "шаблону" за яким створювався файл. Для COM-портів не використовується тому завжди дорівнює 0 (`"NULL"`).

Приклад відкриття COM1:

```
Com_Handle = CreateFile("COM1", GENERIC_READ | GENERIC_WRITE, 0, NULL, OPEN_EXISTING, 0, NULL);
```

Після відкриття COM-порта можна передавати і принімати дані через цей COM-порт.

Для передавання даних використовується API функція `WriteFile` з бібліотеки `kernel32`. Для прийому даних використовується API функція `ReadFile` з бібліотеки `kernel32`.

Приклад опису функцій `WriteFile` і `ReadFile`:

```
BOOL ReadFile(  
    HANDLE hFile, // дескриптор COM порта  
    LPVOID lpBuffer, // Вказівник на буфер, який приймає прочитані дані з порта  
    DWORD nNumberOfBytesToRead, // Число байтів, які читаються з порта  
    LPDWORD lpNumberOfBytesRead, // Вказівник на змінну, яка отримує число прочитаних байтів  
    LPOVERLAPPED lpOverlapped // Вказівник на структуру OVERLAPPED.  
);  
BOOL WriteFile(  
    HANDLE hFile, // дескриптор COM порта  
    LPCVOID lpBuffer, // Вказівник на буфер, який містить дані, які будуть записані у файл.  
    DWORD nNumberOfBytesToWrite, // Число байтів, які будуть записані в файл.  
    LPDWORD lpNumberOfBytesWritten, // Вказівник на змінну, яка отримує число записаних байтів  
    LPOVERLAPPED lpOverlapped // Вказівник на структуру OVERLAPPED  
);
```

Основні параметри послідовного порта описуються структурою `DCB`. Тимчасові параметри – структурою `COMMTIMEOUTS`. Існує ще декілька інформаційних і керуючих структур, але вони використовуються рідше. Налаштування порту полягає в заповненні керуючих структур і наступному виклику функцій настройки:

```
typedef struct _DCB {  
    DWORD DCBlength; // довжина структури (DCB)  
    DWORD BaudRate; // швидкість в біт/сек  
    DWORD fBinary:1; // бинарний режим  
    DWORD fParity:1; // розширення контролю парності  
    DWORD fOutxCtsFlow:1; // слідкування за CTS
```

```
DWORD fOutxDsrFlow:1; // слідкування за DSR
DWORD fDtrControl:2; // режим роботи сигналу DTR
DWORD fDsrSensitivity:1; // чутливість до DSR
DWORD fTXContinueOnXoff:1; // продовження передавання при XOFF
DWORD fOutX:1; // програмне управління потоком при
передаванні (XON/XOFF)
DWORD fInX:1; // програмне управління потоком при
прийомі (XON/XOFF)
DWORD fErrorChar:1; // заміна помилкових символів
DWORD fNull:1; // дії при прийомі нульового символу
DWORD fRtsControl:2; // Задає режим управління потоком
для сигналу RTS
DWORD fAbortOnError:1; // ігнорування запису/читання при
помилці
DWORD fDummy2:17; // зарезервовано
WORD wReserved; // не використовується, рівне 0
WORD XonLim; // мін. кількість символів для
посилання XON
WORD XoffLim; // макс. кіл-ть символів для
посилання XOFF
BYTE ByteSize; // кількість біт в символі
BYTE Parity; // режим паритету 0-
4=no,odd,even,mark,space
BYTE StopBits; // довжина стопового біта 0,1,2 = 1,
1.5, 2
char XonChar; // символ для XON
char XoffChar; // символ для XOFF
char ErrorChar; // символ для заміни помилок
char EofChar; // символ кінця даних
char EvtChar; // символ події
WORD wReserved1; // резервний
} DCB;
```

Для роботи з DCB структурою використовують API функції з бібліотеки kernel32:

BuildCommDCB – заповнює зазначену структуру DCB значеннями, заданими в рядку управління пристроєм. Рядок управління пристроєм використовує синтаксис команди mode MS-DOS.

SetCommState – конфігурує комунікаційний пристрій згідно з даними вказаними в структурі DCB. Функція повторно ініціалізує всі апаратні і керуючі настройки, але не спорожняє черги виводу або введення даних.

GetCommState – читає DCB структуру.

Після закінчення роботи з портом, його потрібно закрити. Закриття порта здійснюється функцією CloseHandle з бібліотеки kernel32.

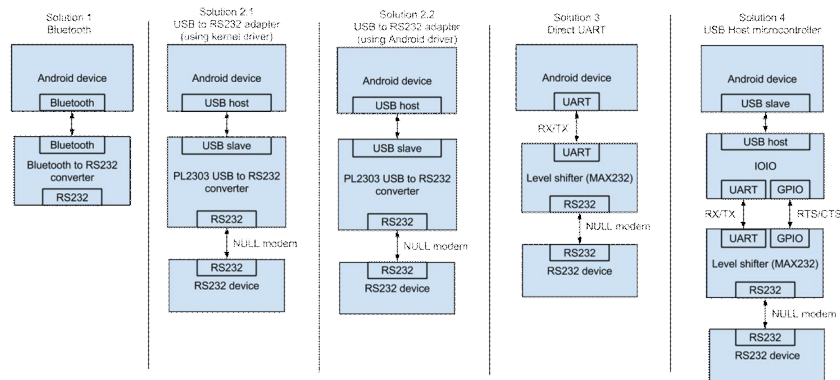
Приклад опису функції CloseHandle:

```
BOOL CloseHandle(HANDLE hObject);
```

Для роботи з COM-портом на Android, на даний час існують різноманітні способи. Одним із способів є використання Virtual Serial Port разом з TCP\_SerPort, яка створює віртуальний COM порт в системі і перенаправляє дані по мережі на Android. Також можна працювати через емулятор терміналу такий як Irssi ConnectBot з підтримкою USB Host. І варіант на якому я зупинився, це використання бібліотеки android-serialport-api.

Дане рішення представляє собою Java обгортку в якій через JNI здійснюються виклики до USB пристрою.

Розробники бібліотеки пропонують 4 варіанти підключення Android пристрою до COM-порта через USB.



Перевагами такого рішення є:

- USB-RS232 перетворювач можна легко знайти на будь-якому радіо ринку;
- Ніяких “збоень” при роботі з Android пристроями;
- Можлива висока швидкість передавання інформації.

Недоліками є:

- Знадобиться USB хост коннектор;
- Більш за все, пристрій прийдеться отримати рут-права.

Для початку нам знадобиться завантажити Android NDK – середовище для роботи з нативним кодом в Java. Також необхідно завантажити вихідний код проекту з svn для робіт из Serial port RS232.

Далі в Android проекті нам необхідно створити папку /jni і скопіювати в неї вміст папки /jni проекту, викачаного з svn (або просто скопіювати всю папку /jni) в проект Android. Потім необхідно додати наступні файли з викачаного проекту:

- SerialPort.java;
- Application.java;
- SerialPortActivity.java;
- SerialPortFinder.java.

Дані файли дозволяють працювати з COM портом, але Вам потрібно буде їх відредагувати під потреби свого проекту. Як відомо, всі пристрої в \*nix подібних системах знаходяться по шляху /dev. Для того щоб встановити шлях до пристрою і baud rate, необхідно у файлі Application.java встановити відповідні значення полів.

```
String path = -path to device-;
int baudrate = -baud rate-;
```

Клас SerialPortActivity.java є розширенням класу Activity і містить абстрактний метод protected abstract void onDataReceived (final byte [] buffer, final int size). Ви можете успадковувати від цього класу свої Activity, в яких буде робота з Serial Port, і, перевизначивши метод onDataReceived, обробити отримання даних з порту.

```
@Override
protected void onDataReceived(final byte[] buffer, final int size) {
    runOnUiThread(new Runnable() {
        public void run() {
            //TO DO your logic
        }
    });
}
```

Можна винести логіку роботи з портом в свої власні класи або класи і не використовувати успадкування від SerialPortActivity.java.

За допомогою класу SerialPortFinder.java з його методами getAllDevices() і getAllDevicesPath() ви можете отримати списки всіх пристроїв і їх шляхів відповідно.

Запис у порт здійснюється за допомогою простого запису в OutputStream, створеного за допомогою класу SerialPort.java, COM порту.

```
mOutputStream.write(new String("text").getBytes());  
mOutputStream.write('\n');
```

Завантаження нативного коду в Android додаток міститься в класі SerialPort.java за допомогою виклику System.loadLibrary("serial\_port"):

```
private native static FileDescriptor open(String path, int baudrate,  
int flags);  
public native void close();  
static {  
System.loadLibrary("serial_port");  
}
```

Параметр serial\_port - це модуль, отриманий внаслідок компіляції коду C за допомогою NDK. Він зазначений в make файлі /jni/Android.mk. Файл SerialPort.c в папці /jni містить виклик нативних функцій системи для роботи з Serial port. Такі параметри COM порту як Data bits, Parity, Stop bits та інші можна змінити в цьому файлі допомогою структури termios, наприклад, так:

```
cfg.c_cflag |= ~PARENB;  
cfg.c_cflag &= ~CSTOPB;  
cfg.c_cflag &= ~CSIZE;  
cfg.c_cflag |= CS8;  
(Data bits=8, Parity=none, Stop bits=1)
```

Після зміни файлу SerialPort.c необхідно скомпілювати необхідні бібліотеки наступним чином:

1. Відкрити командний рядок
2. Зайти в папку з NDK.
3. Встановити шлях до проекту - set NDK\_PROJECT\_PATH = <path to your android project>
4. Набрати - ndk-build.

Бібліотеки будуть скомпільовані і додані в папку /libs вашого проекту. Після цього для установки Android програми на пристрій можна виконати наступну команду:  
adb install <path to you .apk file>

Для коректної роботи serialPortFinder необхідно, щоб перехідник USB-RS232 був підключений, визначений в /dev.

Таким чином, у нас є всі інструменти для здійснення спілкування з великою кількістю пристроїв через через COM-порт у Windows та Android. Використання цих інструментів для відправки і отримання інформації допомагає управляти будь якими пристроями і реалізовувати най неочікуваніші бізнес ідеї.

1. А. А. Мячев, В. Н. Степанов, В. К. Щербо; Интерфейсы систем обработки данных. - М.: Радио и связь, 1989. - 416 с.
2. [https://ru.wikibooks.org/w/index.php?title=COM-порт\\_в\\_Windows\\_\(программирование\)](https://ru.wikibooks.org/w/index.php?title=COM-порт_в_Windows_(программирование))
3. [http://www.lammertbies.nl/comm/info/RS-232\\_specs.html](http://www.lammertbies.nl/comm/info/RS-232_specs.html)

УДК 004:31

Гринюк С.В., Міскевич О.І., Дацюк С.О.  
Луцький НТУ

## ЕВОЛЮЦІЯ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ НА БАЗІ АРХІТЕКТУРИ ARM ТА ЇЇ ВПЛИВ НА РОЗВИТОК ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

**Гринюк С.В., Міскевич О.І., Дацюк С.О.** Еволюція мікропроцесорної техніки на базі архітектури ARM та її вплив на розвиток обчислювальної техніки. В статті розглядається розвиток мікропроцесорної техніки на базі архітектури ARM. З розвитком людства виникла необхідність у використанні більш потужної обчислювальної техніки при низькій енерговитраті. Системи, які побудовані на базі мікропроцесорів з архітектурою ARM набирають все більших оборотів, а саме їх використовують в мобільних гаджетах.

**Ключові слова:** мікропроцесор, архітектура ARM, гаджет, обчислювальна техніка, архітектура x86.

**Гринюк С.В., Мискевич О.И., Дацюк С.А.** Эволюция микропроцессорной техники на базе архитектуры ARM и ее влияние на развитие вычислительной техники. В статье рассматривается развитие микропроцессорной техники на базе архитектуры ARM. С развитием человечества возникла необходимость в использовании более мощной вычислительной техники при низкой энергозатрат. Системы, построенные на базе процессоров с архитектурой ARM набирают все больших оборотов, а именно их используют в мобильных гаджетах.

**Ключевые слова:** микропроцессор, архитектура ARM, гаджет, вычислительная техника, архитектура x86.

**Grunjuk S., Miskevych O., Datsyuk S.** The evolution of microprocessor technology based on the ARM architecture and its influence on the development of computer technology. In the article the development of microprocessor technology-based architecture ARM. With the development of mankind became necessary to use more powerful computing with low power consumption. Systems that are based on the ARM architecture microprocessors are becoming more turns, ie they are used in mobile gadgets.

**Keywords:** microprocessor architecture ARM, gadget, computer science, architecture x86.

**Вступ.** На сьогоднішній день, коли людство вступило в пост-комп'ютерну еру, людство потребує все більш потужніших обчислювальних потужностей при низькій енергоємності. За останні 30 років обчислювальні системи, які допомагають нам у вирішенні поставлених задач стали у десятки разів потужніші. Більшу частину на ринку, звичайно, займають системи з великими потужностями на базі мікропроцесорів з архітектурою x86. Але в останні роки системи, які побудовані на базі мікропроцесорів з архітектурою ARM набирають все більших оборотів.

Перші чіпи ARM з'явилися ще три десятиліття тому завдяки старанням британської компанії Acorn Computers (нині ARM Limited), але довгий час перебували в тіні своїх більш іменитих побратимів - процесорів архітектури x86.

Тому в цій роботі буде освітлено розвиток архітектури ARM, її особливості та переваги над x86, її положення на ринку в даний час, перспективи розвитку та саме головне, як вплине і чи зможуть мікропроцесори на цій архітектурі замінити або частково витіснити системи на базі архітектури x86.

**Історія розвитку архітектури ARM.** Архітектура ARM розробляється фірмою ARM Holdings (раніше називалася Advanced RISC Machines). Перший процесор з архітектурою ARM був випущений в 1985 році. У процесі розвитку та удосконалення архітектури фірма ARM представила користувачам цілий ряд модифікацій ядра, що утворюють сімейства.

У наші дні 32-розрядна ARM - найбільш широко використовувана архітектура процесорів для мобільних пристроїв, і найбільш популярна архітектура 32-розрядних процесорів для вбудованих систем. З роки назад була представлена 64-розрядна ARM, але про неї ми поговоримо пізніше.

В даний час архітектуру ARM представляють два класи: «класичні» процесори, куди входять сімейства ARM7, ARM9 і ARM11, і сімейство Cortex, поділюване на три «профілю» Cortex-A, Cortex-R і Cortex-M. На рис. 1 представлена діаграма, що ілюструє розвиток архітектури ARM від покоління до покоління.



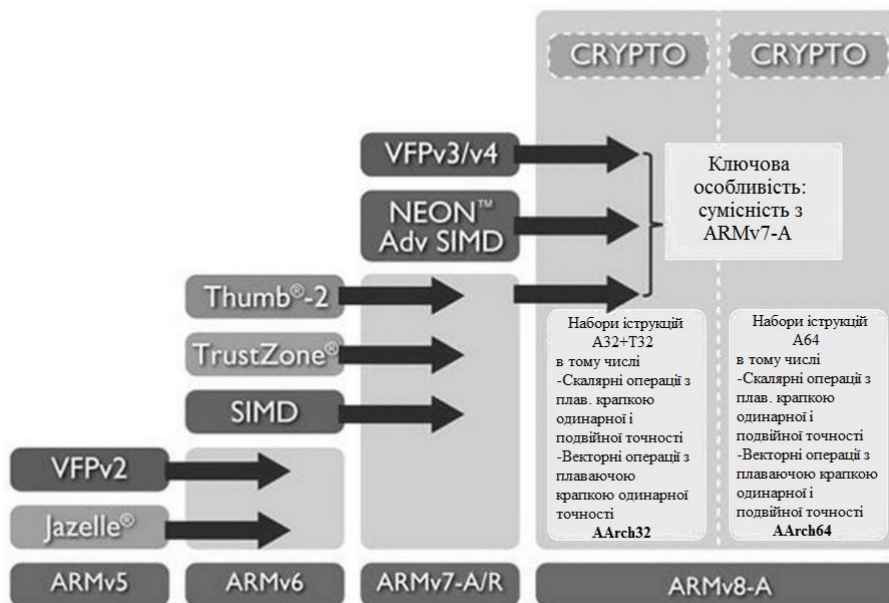


Рис. 1. Еволюція архітектури ARM

**Особливості та порівняння архітектури ARM.** Почати варто з того, що в процесорній архітектурі x86, яку зараз використовують компанії Intel і AMD, застосовується набір команд CISC (Complex Instruction Set Computer), хоч і не в чистому вигляді. Так, велику кількість складних за своєю структурою команд, що довгий час було відмінною рисою CISC, спочатку декодуються в прості, і тільки потім обробляються. Ясна річ, на весь цей ланцюжок дій йде чимало енергії.



Рис. 2. Чіп ARM1 - первісток компанії Acorn Computers, який проводився на фабриках VLSI

В якості енергоєфективної альтернативи виступають чіпи архітектури ARM з набором команд RISC (Reduced Instruction Set Computer). Його перевага в спочатку невеликому наборі простих команд, які обробляються з мінімальними витратами. Як результат, зараз на ринку споживчої електроніки мирно уживаються дві процесорні архітектури - x86 і ARM, кожна з яких має свої переваги і недоліки.

Архітектура x86 позиціонується як більш універсальна з точки зору посильних їй завдань, включаючи навіть настільки ресурсомісткі, як редагування фотографій, музики і відео, а також шифрування і стиснення даних. У свою чергу архітектура ARM «виїжджає» за рахунок вкрай низького енергоспоживання і в цілому достатньої продуктивності для найважливіших на сьогодні завдань: створення, перегляд веб-сторінок і відтворення медіаконтента.

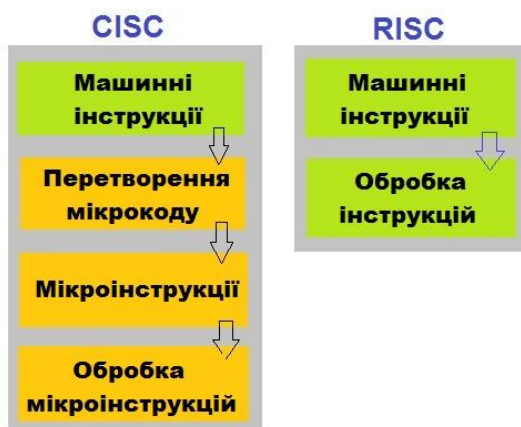


Рис. 3. Архітектурні відмінності процесорів x86 (набір команд CISC) і ARM (набір команд RISC)

З технічної точки зору називати чіпи архітектури ARM процесорами не зовсім вірно, адже крім одного або декількох обчислювальних ядер вони включають цілий ряд супутніх компонентів. Більш доречними в даному випадку є терміни однокристална система і система-на-чіпі (від англ. System on a chip).

Так, новітні однокристалні системи для смартфонів і планшетних комп'ютерів включають контролер оперативної пам'яті, графічний прискорювач, відеокодек, аудіокодек і опціонально модулі бездротового зв'язку. Вузькоспеціалізовані чіпи можуть включати додаткові контролери для взаємодії з периферійними пристроями, наприклад датчиками.

Окремі компоненти однокристалної системи можуть бути розроблені як безпосередньо ARM Limited, так і сторонніми компаніями. Яскравим тому прикладом є графічні прискорювачі, розробкою яких крім ARM Limited (графіка Mali) займаються Qualcomm (графіка Adreno) і NVIDIA (графіка GeForce ULP).

Не варто забувати і про компанію Imagination Technologies, яка нічим іншим, крім проектування графічних прискорювачів PowerVR, взагалі не займається. А адже саме їй належить мало не половина глобального ринку мобільного графіки.

ARM чіпи відрізняються від x86 лише двома аспектами. Перше – це енергоефективність. ARM чіпи потребують значно менше енергії для роботи, ніж процесори x86. Ще одна відміна сімейств ARM і x86 - мікроархітектури RISC і CISC. Однак і це вже не можна вважати принциповою відмінністю: починаючи з модифікації i486DX, x86-чіпи стали більше нагадувати RISC-процесори. Починаючи з цього покоління, мікросхеми, зберігаючи сумісність з усіма попередніми наборами команд, демонструють максимальну продуктивність тільки з обмеженим набором простих інструкцій, який підозріло схожий на набір RISC-команд. Тому сьогоднішні x86 можна сміливо вважати CISC-процесорами з RISC-ядрами: вбудований в мікросхему апаратний транслятор декодує складні CISC-інструкції в набір простих внутрішніх RISC-команд. Навіть незважаючи на те, що кожна CISC-інструкція може "розкладатися" на кілька RISC-команд, швидкість виконання останніх забезпечує значний приріст продуктивності. До того ж, не слід забувати про суперскалярної і суперконвейерная сучасних чіпів.

**Чіпи ARM сьогодні та перспективи розвитку.** Зараз компанія ARM Limited займається лише розробкою референсних процесорних архітектур і їх ліцензуванням. Створення ж конкретних моделей чіпів і їх подальше масове виробництво – це вже справа ліцензіатів ARM, яких налічується безліч. Є серед них як відомі лише у вузьких колах компанії на зразок STMicroelectronics, HiSilicon і Atmel, так і IT-гіганти, імена яких у всіх на слуху - Samsung,

NVIDIA і Qualcomm. З повним списком компаній-ліцензіатів можна ознайомитися на відповідній сторінці офіційного сайту ARM Limited.

Така велика кількість ліцензіатів викликано в першу чергу великою кількістю сфер застосування ARM-процесорів, причому мобільні гаджети - це лише вершина айсберга. Недорогі і енергоефективні чіпи використовуються у вбудованих системах, мережевому обладнанні і вимірювальних приладах. Платіжні термінали, зовнішні 3G-модеми та спортивні пульсометри - всі ці пристрої засновані на процесорній архітектурі ARM.

За підрахунками аналітиків, сама ARM Limited заробляє на кожному виробленому чіпі \$ 0,067 у вигляді роялті. Але це сильно усереднена сума, адже за собівартістю новітні багатоядерні процесори значно перевершують одноядерні чіпи застарілої архітектури.

Звичайно більшу частину чіпів, які випускаються на архітектурі ARM використовуються для мобільних пристроїв. Лідерами по випуску чіпів на сьогоднішній день Qualcomm, MediaTek, Apple, Samsung та Nvidia. За останні 5 років пройшло дуже значущих подій, які суттєво вплинули на ринок чіпів ARM і головним ініціатором стала компанія Apple.

Apple зробила чергову революцію в області однокристальних мобільних систем разом з A8X і це велика загроза для всіх конкурентів, включаючи таких виробників чіпів як Intel, Qualcomm, Samsung і Nvidia. Але вистачить порожніх слів, перейдемо до фактів.

Цікаві думки з приводу дій Apple на ринку планшетів і мобільних процесорів висловив головний редактор AI Даниель Еран Ділгер і я згоден з його спостереженнями, суть яких в тому, що Apple непомітно для простих споживачів просто рве конкурентів і їх бізнес в клепті. Intel вже втратила мільярди на протистоянні з компанією з Купертино. У Qualcomm, Samsung і Nvidia як у виробників мобільних процесорів справи теж йдуть не найкращим чином.

Адже в минулому ніщо не віщувало біди. Та ж Intel могла б заробити величезні гроші, якби надала Apple досить ефективний для iPad чіп чотири роки тому. Саме цю компанію Стів Джобс розглядав як постачальника мобільних процесорів для фірмових планшетів. От тільки коли дійшло до справи, чіп IntelAtom (тоді відомий під кодовою назвою Silverthorne) виявився, м'яко кажучи, ненажерливим, і від сили можна було розрахувати на 2-3 години автономності. Альтернатив з потрібними характеристиками на ринку не знайшлося теж. Ось одна з причин, що підштовхнули Apple на розробку власних процесорів, у результаті чого народився A4, що показав себе у всій красі в iPad і в iPhone 4.

Також величезними кроками чіпи ARM завоюють своє місце у серверних системах. Хороші результати в цьому показує компанія AMD. У 2014 почалися поставки новітніх ARM-сумісних чіпів, які реалізуються під кодовою назвою Seattle. Цей же рік стане початком продажів процесорів Berlin. Вони будуть поставлятися як у вигляді «чистого» ЦПУ, так і в якості інтегрованих пристроїв APU, що дозволить поєднати відразу і центральний процесор і графіку. Потім планується вихід чіпа Warsaw, який покликаний конкурувати з продукцією Intel в області високовиробничих рішень.

Продукт Seattle в першу чергу цікавий хоча б по тому, що являє собою принципово новий підхід AMD до виробництва чіпів, адже в них тепер поєднується процесори відразу двох архітектур. Ці дві підсистеми можна об'єднати за допомогою фірмової технології FreedomFabric, яка перейшла до компанії AMD після покупки SeaMicro.

На сьогоднішній день економія електроенергії вважається пріоритетним моментом, тому чіпи ARM застосовуються в багатьох смартфонах і подібні пристрої. До недавнього часу їх використання не було широкого поширено через їхню малу потужності, а також відсутності підтримки 64-бітних чіпів, що було вкрай необхідно в будь-яких серверах. Повідомляється, що в наступному році планується налагодити масове виробництво 64-бітових чіпів, які не тільки збережуть позитивні сторони даної архітектури, але ще й будуть володіти набагато більш високою потужністю. Її буде достатньо як для роботи в обладнанні, так і на мікросерверах.

Масові поставки цих пристроїв плануються на 2014 рік, причому призначатися вони будуть для хмарного хостингу, роботи з електронною поштою, а також для багатьох інших популярних функцій.

AMD повідомляє, що процесори Seattle стануть працювати на частоті вище, ніж 2 ГГц і виявиться, що вони фактично швидше нинішніх OpteronXKyoto. Кожен чіп цього нового продукту буде оснащений 8 ядрами і здатністю підтримки до 128 Гб ОЗУ, причому тут же заявлена апаратна підтримка 10-GbEthernet.

Крім усього іншого, тепер точно відомо, що процесори Berlin будуть підтримувати нове покоління AMD CPU Steamroller. Це дозволить зберегти їх попередні розміри, проте продуктивність істотно зросте. Також заявлено ще й про можливість підтримки архітектури AMD Heterogenous System Architecture. Нарешті Warsaw отримають від 12 до 16 процесорних ядер Piledriver, серверну специфікацію OpenCompute 3.0 і, як очікує сам виробник, підтримку серед операторів хмарних платформ.

За прогнозами Digitimes Research, протягом найближчих чотирьох років обсяг ринку серверів, який працюють на базі процесорів з архітектурою ARM, зросте більш ніж у 50 разів. У нинішньому році буде випущено близько 20 тисяч таких серверів, що приблизно складе 0,2% від обсягу світового ринку. При цьому в 2017 році поставки ARM-серверів збільшаться до 1,06 мільйона штук, в результаті чого їх частка складе 10-15 відсотків.

Якщо все дійсно так, то протягом наступних чотирьох років відвантаження серверів будуть збільшуватися щороку приблизно на 170%. На думку аналітиків, цей підйом можливий завдяки реалізації сумісності архітектури ARM з операційною системою Windows Server. При цьому залежно від рівня підтримки чіпів ARM виробниками, поставки таких серверів можуть збільшуватися від 150 до 180 відсотків щорічно.

Який ми можемо побачити подальший розвиток архітектури ARM? Останні 5 років стали великим проривом для цієї архітектури і розвиток буде все більше прискорюватись. Чіпи будуть ставати все потужніші і набирати нових рис, наприклад як когнітивні обчислення, які вже анонсовані для чіпа Snapdragon 820.

ARM представила нові мікропроцесори Cortex-A72, якими будуть оснащуватися смартфони та планшети наступного покоління.

За заявами розробників, за обчислювальної потужності, Cortex-A72 перевершує нині використовуються Cortex-A15 і Cortex-A57 в 3,5 рази і в 2 рази відповідно. Процесор підтримує 64-бітові команди, його максимальна тактова частота складає 2,5 ГГц, а в певних випадках можна добитися і 3 ГГц. Крім того, завдяки технології ARM big.LITTLE Cortex-A72 підтримує суміщення з ядрами серії Cortex-A53, що дозволяє майже на третину збільшити продуктивність і позитивно позначається на енергоспоживанні.

Новинка використовує на 75 відсотків менше енергії в порівнянні з актуальними процесорами при однакових навантаженнях. До речі, віце-президент ARM Ян Дрю заявив, що компанія планує покращити енергоефективність ще на 40-60 відсотків. За його словами, за час існування потужність смартфонів збільшилася майже в 50 разів, а показники акумуляторів вирости максимум удвічі. При цьому для більшості користувачів продуктивність достатня, а смартфони та планшети доводиться заряджати "щонаочі".

Разом з процесором ARM анонсувала графічний прискорювач Mali-T880, майже вдвічі підвищує продуктивність при зниженому на 40 відсотків енергоспоживанні в порівнянні з Mali-T760. Укупі з Cortex-A72, стверджують розробники, новий чіп забезпечить на гаджетах якість зображення, порівнянне з ігровими консолями. Ядро також орієнтоване на підтримку відео формату 4K.

**Висновок.** Мікроархітектура AMD має колосальний вплив на обчислювальні системи в різних сферах нашої діяльності. Чіпи, які десятки років використовувались, здебільшого, в побутовій техніці почали свій розвиток саме в даний час, коли людство ввійшло в пост-комп'ютерну еру. В наш час, коли світ озадачений проблемою збереження енергії, чіпи AMD знаходять своє місце у різноманітних сферах нашого життя і в майбутньому ці чіпи будуть витісняти обчислювальні системи на базі процесорів x86.

1. ARMv7-M Architecture Reference Manual // ARM Limited. [http://www.eecs.umich.edu/courses/eecs373/readings/ARMv7-M\\_ARM](http://www.eecs.umich.edu/courses/eecs373/readings/ARMv7-M_ARM).
2. ARM Processor Architecture // ARM Limited. <http://www.arm.com/products/processors/instruction-set-architectures/index.php>
3. ARM Architecture Reference Manual ARMv8, for ARMv8-A architecture profile // ARM Limited. [http://www.cs.utexas.edu/~peterson/arm/DDI0487A\\_a\\_armv8\\_arm\\_errata.pdf](http://www.cs.utexas.edu/~peterson/arm/DDI0487A_a_armv8_arm_errata.pdf)
4. Guy Larri. ARM810 – Dancing to the Beat of a Different Drum // ARM Limited. [http://www.dlhoffman.com/publiclibrary/software/hot\\_chips\\_papers/hc96/hc8\\_pdf/4.1.pdf](http://www.dlhoffman.com/publiclibrary/software/hot_chips_papers/hc96/hc8_pdf/4.1.pdf)
5. VLSI Technology Now Shipping ARM810 // EE Times. [http://www.eetimes.com/document.asp?doc\\_id=1208831](http://www.eetimes.com/document.asp?doc_id=1208831).
6. Cortex-M0 Specification Summary // ARM Limited. <http://arm.com/products/processors/cortex-m/cortex-m0.php?tab=Specifications>.

УДК УДК 004.413 (045)

Мельник К.В. к.т.н., доц., Жигаревич О.К., Мороз С.М.,  
Луцький національний технічний університет

## ДОСЛІДЖЕННЯ ТА РОЗРОБКА ГРУПОВОГО ОРГАНАЙЗЕРУ

**Мельник К.В., Жигаревич О.К., Мороз С.М.** Дослідження та розробка групового органайзеру. У статті розкрито та важливість надання користувачеві зручного та надійного засобу для керування власним часом та поставленими перед собою завданнями. Запропоновано вирішення даного питання шляхом використання програм-органайзерів. Запропоновано авторське бачення розв'язання даної проблеми шляхом застосування розробленої програми.

**Ключові слова:** органайзер, груповий органайзер, система, інформаційна технологія, автоматизована система, програмне забезпечення.

**Мельник К.В., Жигаревич О.К., Мороз С.М.** Исследование и разработка группового органайзера. В статье раскрыто важность предоставления пользователю удобного и надежного средства для управления собственным временем и поставленными перед собой задачами. Предложено решение данного вопроса путем использования программ-органайзеров. Предложено авторское видение решения данной проблемы путем применения разработанной программы.

**Ключевые слова:** органайзер, групповой органайзер, система, информационная технология, автоматизированная система, программное обеспечение.

**Melnyk K., Zhigarevich O., Moroz S.** Research and development of the group organizer. The article deals with the importance of providing the user with a convenient and reliable means to manage their own time and set goals. Provides a solution to this issue through the use of software-organizers. Invited to the author's vision to solve this problem by applying the developed program.

**Keywords:** organizer, group organizer system, information technology, automated system software.

**Постановка проблеми.** В наш час багато ділових людей заклопотані в своїх повсякденних справах. Зустрічі, робота, відпочинок, свята, все це дуже втомлює людину і вона перестає бути хазяїном свого часу. Таким людям потрібна організація, тому почали використовуватися щоденники. Але щоденник не може повідомити про зустріч або, наприклад день народження близької людини автоматично і ним не зручно користуватися, оскільки не можна редагувати інформацію. Знайшовся вихід з появою ПК. З розвитком програмного забезпечення, розробники приділили увагу цій проблемі і стали розроблятися спеціальні програми – органайзери. Вони надають змогу користувачам створювати розклади на кілька місяців вперед, задавати певні події, про які вони будуть повідомлені органайзером пізніше. Більшість програм мають зручний інтерфейс, а також щось на зразок адресної книжки.[1, 2]

Основною метою створення програми органайзеру – це надання користувачеві зручного та надійного засобу для керування власним часом. Основним елементом є календар та записник, де й будуть зберігатися усі дані користувача. Пам'ять людини обмежена, а тому може потребувати нагадування про настання певних подій, тобто система керування часом повинна мати таймер, який буде відслідковувати усі події і в разі необхідності попереджати користувача про настання події. Користувачеві необхідно надати зручний засіб для обліку виконаних задач, та перегляду їх характеристик.

**Аналіз досліджень.** За результатами дослідження було визначено, що найпотужнішим, в плані можливостей, органайзером є Microsoft Outlook, основними його недоліками є надмірна складність та багато можливостей, що непотрібні більшості користувачів. Наприклад поштові клієнти зараз витісняє Google Mail. Інші розглянуті програми мають одну вагому перевагу, вони розповсюджуються безкоштовно.

Після дослідження програм-аналогів було визначено, що всі програми-аналоги жорстко прив'язані до платформи та середовища, а значить не мають достатньої гнучкості.

Тому про розробці даної програми робиться акцент саме на гнучкість, тобто можливість для користувача працювати зі своєю програмою на будь-якій платформі та на будь-якій системі.

Виходячи з цього **мета** нашої статті полягає в описі розробки програми-органайзеру для використання широкого кола споживачів.

**Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.** Основне призначення програм органайзерів – допомога людині (працівнику) в організації

власного часу на робочому місці. Розроблювана програма має забезпечувати відповідність нормам режимів, прийнятим на законодавчому рівні.

Розподіл норми робочого часу протягом конкретного календарного періоду називається режимом робочого часу. Елементами режиму виступають:

- 1) час початку і закінчення роботи;
- 2) час і тривалість перерв;
- 3) тривалість і правила чергування змін.

На законодавчому рівні встановлюються лише окремі елементи режиму робочого часу. Так, Указом Президента України від 26 квітня 1995 р. "Про введення на території України регіональних графіків початку робочого дня" передбачено введення рішеннями виконкомів обласних рад регіональних (місцевих) графіків початку робочого дня для однозмінних підприємств всіх форм власності.

Режим робочого часу передбачає 5-ти або 6-денний робочий тиждень, початок і кінець робочого дня, час і тривалість обідньої перерви, число змін протягом облікового періоду.

Відповідно до ст. 57 КЗпП час початку і закінчення щоденної роботи (зміни) передбачається правилами внутрішнього трудового розпорядку і графіками змінності.

Органайзер (англ. organizer) – спочатку під цим терміном розуміли невелику книгу, що містить календар, адресну книгу і блокнот, що служить для організації інформації про персональні контакти і події. З розвитком інформаційних технологій книга стала замінюватися кишеньковими персональними комп'ютерами, комп'ютерними програмами та онлайн-органайзерами, що володіють додатковими функціями: нагадування про майбутні події, захист і синхронізація інформації.

У 21 столітті під органайзером частіше розуміється програмне забезпечення для ПК і мобільних пристроїв. Їх безумовними перевагами є практично необмежений обсяг даних, що вводяться, встановлення автоматичних нагадувань. Форми органайзера можуть бути різними. Є, приміром, вузькоспеціалізовані органайзери: органайзер рецептів, органайзер фінансів, органайзер музичної колекції.

Організатор є засобом управління часом. Попереднє планування справ допомагає підвищити ефективність будь-якої діяльності, як особистої, так і професійної. Тому і людину, яка консультує організації та приватних осіб у сфері підвищення ефективності управління часом, теж іноді називають «органайзер». У США навіть існує Національна Асоціація Професійних Органайзерів (NAPO).

Комп'ютерна програма-органайзер - це комп'ютерна програма, що відноситься до прикладного програмного забезпечення і призначена для накопичення інформації користувача, а потім оперативного пошуку по ній, організації справ і контролю за їх виконанням, відстеження визначених користувачем подій. Є однією з форм персонального органайзера.

Функції типовою комп'ютерної програми-органайзера пов'язані із забезпеченням роботи наступних підрозділів:

- календар;
- менеджер контактів (користувальницька адресно-телефонна книга);
- записна книжка і листки-нотатки (аналог паперових листків-липучок);
- події, прив'язані до певної дати і часу (наприклад, свята або зустрічі);
- планувальник завдань (завдань) для контролю за їх самостійним або стороннім виконанням;

- нагадувачі-будильники про певні користувачем події.

Деякі програми-органайзери можуть як не мати якого-небудь з перерахованих підрозділів, так і забезпечувати додаткову функціональність, наприклад, дозволяти працювати з електронною поштою, таким чином виконуючи функції поштового клієнта.[4]

Для забезпечення доступу до інформації з будь-якого комп'ютера в будь-якій точці світу, а також для організації спільної роботи, багато персональні інформаційні менеджери мають клієнт-серверів архітектури. Як клієнт може використовуватися як спеціальна програма, так і веб-інтерфейс. Іноді доступ до інформації може бути забезпечений обома способами. Наприклад в якості клієнта Microsoft Exchange Server може використовуватися і програма Microsoft Outlook і веб-інтерфейс Outlook Web Access.



Груповий електронний органайзер - ефективне рішення для управління колективом за допомогою комп'ютера, що дозволяє створювати завдання як для себе, так і для підлеглих і активно контролювати їх виконання.

Груповий органайзер може дозволяти іншим користувачам активно контролювати і коригувати виконання задачі, змінювати її основні параметри.

Для пошуку найкращих рішень, необхідно слідувати детальному процесу з ціллю отримання аналізу вимог проекту і розробки проекту для його відповідності цим вимогам. Якщо цей процес включає аналіз і проектування системи з об'єктно-орієнтованої точки зору, це називається процесом об'єктно-орієнтованого аналізу і проектування (object-oriented analysis and design (OOAD) process). Програмісти знають, якою б простою не здавалась проблема, час, витрачений на аналіз і проектування, може зберегти нескінченні часи в результаті відмови від погано спланованої розробки системи в процесі її реалізації.

OOAD – це узагальнений термін для ідей, що лежать в основі процесу, який ми використовуємо для аналізу проблеми і виробки методу для її вирішення.

Перед тим як створювати програму для вирішення конкретної задачі, необхідно досконало зрозуміти цю задачу і спланувати шляхи її вирішення. Коли програма розпочинає свій життєвий цикл, в рівному степені важливо розуміти типи, що мають місце в розпорядженні стандартних блоків і використовувати основні правила побудови програми.[5]

Розроблений програмний продукт має назву "Time Manager". Це віртуальний органайзер, призначений для ведення щоденника и виводу повідомлень про події. Тобто, користувач має змогу вводити дані про певні події в таблицю, за допомогою календаря і цієї таблиці він має змогу проглянути, події, що на нього чекають в майбутньому або ті, що виконані в минулому.

Мінімальні вимоги до апаратного забезпечення:

- Процесор Intel Pentium III 500МГц;
- 128Мб оперативної пам'яті;
- 200Мб вільного місця на жорсткому диску;

Вимоги до програмного забезпечення:

Будь-яка з перелічених операційних систем з графічним інтерфейсом:

- X Window System (Unix / Linux);
- Mac OS X;
- Microsoft Windows.

Для запуску програми достатньо в графічному режимі виконати файл ZOrganizer. Програма проста у використанні, тому нею легко користуватися навіть на інтуїтивному рівні. Дані таблиці на кожен день зберігаються в окремих файлах. Один файл на один день.

Надійна робота продукту забезпечена завдяки коректній взаємодії модулів. Вони реалізовані незалежно один від одного. Завдяки інкапсуляції зміна поведінки одного модуля не впливає на інші, якщо не змінювати інтерфейсу.

Для запуску програми достатньо виконати файл "TimeManager.exe".

Після першого запуску буде створено Xml файл, його першопочатковий вміст відображає основу документу а також буде відображене головне вікно програми рис 1.

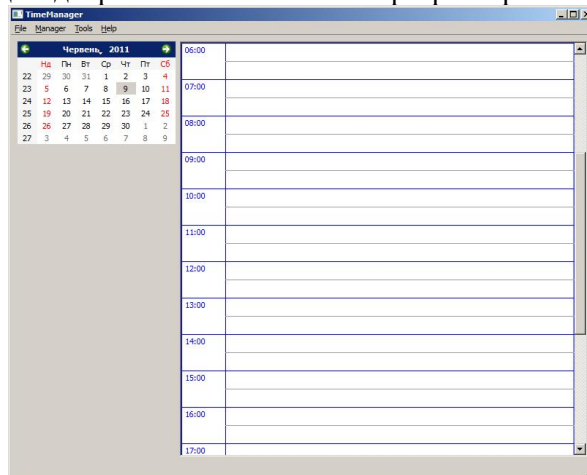


Рис. 1. Головне вікно програми

Після запуску у списку будуть відображатися задачі на поточний день.

Щоб перейти на інший день, треба просто натиснути на інший день в календарі. Таким чином можна проглядати, що заплановано на наступні дні, натискаючи відповідну дату на календарі.

Для створення нової задачі необхідно у меню «Manager» вибрати пункт «New Task» рис.2.

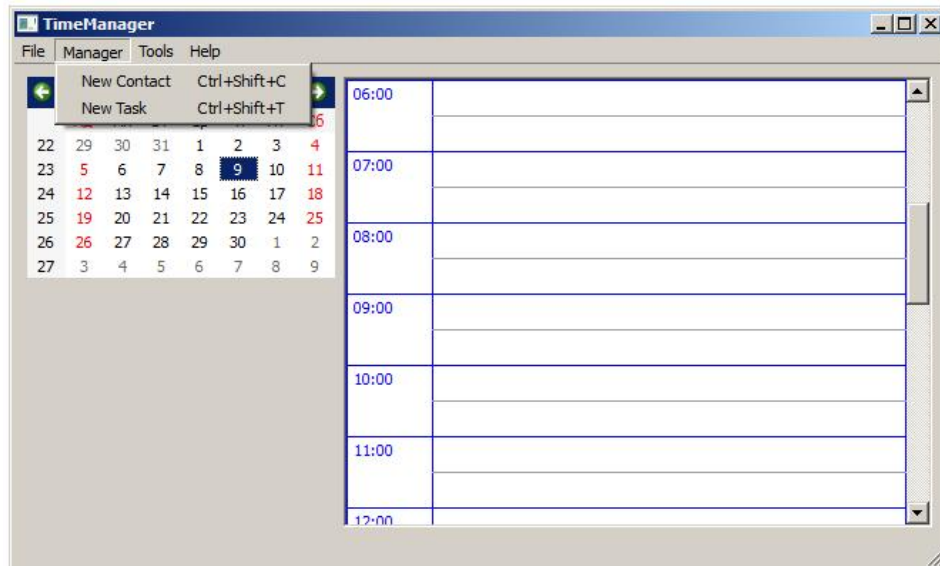


Рис. 2. Меню створення задачі

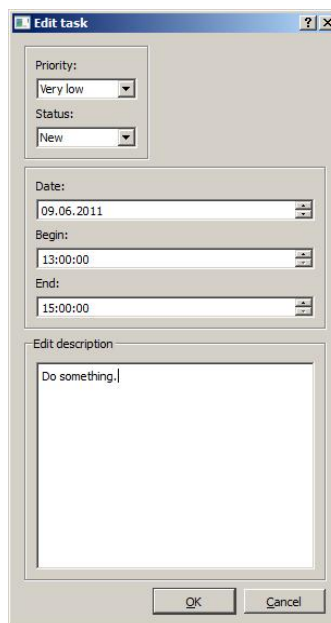


Рис. 3. Діалогове вікно створення задачі

Тут можна налаштувати задачу, визначивши її пріоритет, статус рис.3. Дата за замовчуванням та, що була вибрана у календарі головного вікна. Натиснувши відміну «Cancel» діалог пропаде та у моделі нічого не зміниться. Натиснувши ОК, у моделі створиться новий об'єкт Task, модель дасть команду парсеру занести новий об'єкт до файлу та відобразиться у таблиці, якщо встановлена дата співпадає з датою на календарі.

**Висновки та перспективи подальшого дослідження.** Програми-органайзери стають дедалі популярнішими для ділових людей, оскільки не можливо утримувати весь той об'єм задач у



пам'яті зразу. Людина не може про все пам'ятати і є ризик забути про найважливіші справи. Саме для таких людей було розроблено «Time Manager».

Для знаходження оптимального рішення було проведено дослідження предметної області про режими роботи працівників, було проведено аналіз задач і можливостей, що мають забезпечувати подібні програми. Завдяки порівнянню аналогів винайдено недоліки, які компенсує розроблена програма.

Визначено, що окрім базових задач органайзеру дуже важливе місце посідає переносимість і можливість користувачеві працювати зі своїми даними на різних платформах за різними системами. Це забезпечують веб-органайзери, але в них є серйозний недолік – необхідно обов'язкове підключення до інтернету.

Після виявлення вимог було проведено проектування системи. За допомогою розробки варіанту використання роботи користувача із системою визначено які саме можливості має забезпечити органайзер та як вони будуть використані користувачем.

Була розроблена архітектура з використанням партнерів проектування. Спроектвана взаємодія інтерфейсу з даними.

Для розробки обрано мову програмування C++. Для забезпечення кросплатформеності найкращою і найпотужнішою бібліотекою виявилась Qt. Також вона забезпечує класи для роботи з XML. Саме в цьому форматі зберігаються дані користувача. XML забезпечує гнучкість, він не залежить від платформи і робота з ним можлива на будь-якій системі. Також, завдяки XML забезпечується групова робота з органайзером, для цього достатньо використати файл з даними іншого користувача, наприклад для того, щоб дати йому нові задачі або перевірити результат виконання інших задач. За допомогою методів ООП було забезпечено незалежність модулів, що дає базу для подальшого розширення і удосконалення.

1. Струасуп Б. Мова програмування C++. – К.: Біном, 2008. – 1104 с.
2. Мейерс С. Ефективне використання C++. 55 правильних способів покращити структуру та код ваших програм – М.: ДМК Прес, 2006. – 300 с.
3. Шилдт Г. Довідник програміста по C/C++, третє видання. : Пер. з англ. – М.: видавничий дім «Вільямс», 2003. – 432
4. Немет Е., Снайдер Г., Сібасс С., Хейн Т. UNIX: керівництво системного адміністратора. Для професіоналів. Третє видання – СПб.: Пітер; К.: Видавницька група BHV, 2005. – 925 с. Пуцов В. І. Теоретичні засади практичної андрагогіки / В. І. Пуцов // Професійний розвиток педагогічних працівників: практична андрагогіка : наук.-метод. посіб. / за заг. ред. В. І. Пуцова, Л. Я. Набоки. – К. : [б. в.], 2007. – 228 с.
5. То Гамма Е., Хелм Р., Джонсон Р., Вліссідес Дж. Прийоми об'єктно-орієнтованого проектування. Паттерни проектування. М.: Пітер, 2007. – 366 с.

УДК 629.113(071):004.01:004.04

Каганюк О.К. к.т.н. доц., Панчук Р.Я.  
Луцький національний технічний університет

## РИНКОВА ПРИВАБЛИВІСТЬ ХМАРНОГО СЕРВЕРА ТРАНСПОРТНОЇ ЛОГІСТИКИ

**Каганюк О.К., Панчук Р.Я. Ринкова привабливість хмарного сервера транспортної логістики.** Представляється ринкова привабливість використання хмарного сервера транспортної логістики, приклад використання хмарного сервера для контролю за маршрутними засобами міста. Представлена економічна ефективність даної системи.

**Ключові слова :** Ринкова привабливість, транспортна логістика, хмарний сервер, МАК.

**Kaganiuk A.K., Panchouk R.Ja. Marketability cloud services transport logistics.** It seems the market appeal of using cloud services transport logistics, an example of using the cloud server to control the routing means city. Submitted by economic efficiency of the system.

**Keywords:** market attractiveness, transport logistics, cloud services, МАК .

**Каганюк А.К., Панчук Р.Я. Рыночная привлекательность облачного сервиса транспортной логистики.** Представляется рыночная привлекательность использования облачного сервиса транспортной логистики, пример использования облачного сервера для контроля за маршрутными средствами города. Представлена экономическая эффективность данной системы.

**Ключевые слова:** Рыночная привлекательность, транспортная логистика, облачный сервис, МАК.

Основним завданням транспортної логістики, як і логістики промислових підприємств, є збільшення прибутку транспортних організацій. Цього можливо досягти за рахунок координації транспортного обслуговування споживачів за їх замовленнями, в яких містяться умови поставок. Все це дає змогу отримати конкурентні переваги на ринку і зменшити витрати.

Завдання, які вирішує логістично-транспортна система і розробку її стратегії можна поділити на три групи. Перша з груп пов'язана з формуванням ринкових зон обслуговування, прогнозом матеріалопотоків, їх обробкою в системі обслуговування та іншими роботами в оперативному управлінні і регулюванні матеріалопотоку. Друга група – завдання із розробки системи організації транспортного процесу (план перевезень, план розподілу виду діяльності, план формування вантажопотоків, графік руху транспортних засобів і ін.). Третя група – це управління запасами на підприємствах, фірмах, складських комплексах, розміщення запасів і їх обслуговування транспортними засобами, інформаційними системами.

Оптимізація і вирішення цих завдань, що особливо актуально в умовах ринку, залежить від конкретної ситуації, умов і вимог до ефективної логістичної системи, а також від проблем, пов'язаних із забезпеченням виробництва сировиною і напівфабрикатами, усуненням вузьких місць в технології доставки різних видів продукції в пункти виробництва, складування і збуту.

Для ефективного використання транспортних потоків в місті пропонується використовувати хмарний сервіс «ЗЕЛЕНА ХВИЛЯ» (інтелектуальна інфраструктура дорожнього руху в реальному масштабі часу .на основі використанні глобальних систем позиціонування навігації (GPS, GPRS), дорожніх контролерів, засобів радіочастотної навігації і інше.

**Актуальність** наукової роботи полягає в тому, що процеси роздільного і несинхронізованого розвитку двох компонентів: реального світу та кіберпростору в теперішній час здобуває векторний формат, направлений на створення структурованих і взаємо інтегрованих компонентів КіберГеоСистем (КГС). Остання еволюціонує в теперішній час шляхами створення сервісів точного та оптимізованого моніторингу і оптимального управління складним реальним світом на основі моделей взаємно однозначного відображення всіх земних процесів і явищ

Дана система хмарних серверів має дуже великі перспективи самого різноманітного впровадження. Проте багато фірм не ризикують вкладати кошти у технології що не пов'язані напряму з інтернетом, як хмарні обчислення, збереження інформації на віддалених серверах та ін. Проте використання хмарних сервісів для транспортної логістики є досить прибутковою справою. В даній статті показано ринкову привабливість даного сервера на принципі одного міста [1].

Найбільшою **проблемою** є несприйняття більшістю людей таких технологій з транспорт логістикою. Дані технології є надзвичайно мало розповсюдженими та розрекламованими у світі. Даний сектор є надзвичайно перспективним при збільшенні уваги до нього, що можливо лиш при

збільшенні використання схожих систем у повсякденні. При створенні готових макетів ринкової ефективності даних систем ми можемо залучити сюди кошти, що призведе до автоматичного процесу зростання даної галузі[1].

Головна **ціль** даної наукової роботи полягає в тому щоб доказати ефективності використання хмарних серверів для транспортної логістики. Що дозволить запустити лавиноподібний процес в розробці та використанні хмарних серверів для самих різноманітних напрямів.

Основними **критеріями** є саме ринкова ефективність даних серверів транспортної логістики. Адже саме це і є головним чинником який показує економічну привабливість для інвестицій даних проектів. Також важливим критерієм є вартість усієї системи та її обслуговування, адже запустити таку систему це ще не вся справа.

**Як приклад системи використання хмарного сервера транспортної логістики** можуть слугувати такі системи як система «МАК» компанії «Візор» та сервіс «EasyWay». Детальніше розглянемо систему «МАК» її ринкову привабливість.

Базою для програмної платформи МАК була вибрана операційна система FreeBSD, як сучасна операційна система, що динамічно розвивається та має ряд важливих переваг порівняно із іншими операційними системами.

FreeBSD – безкоштовна, Вам не потрібно витратити зайві кошти на купівлю дорогих серверних операційних систем.

Відкритий програмний код FreeBSD дає можливість будь-якому користувачеві та спеціалісту виявити помилку та виправити її. Системи Unix багато років використовуються мільйонами фахівців.

Ліцензія BSD дозволяє вільно модифікувати код операційної системи відповідно до вимог Вашого бізнесу.

FreeBSD спроектована як безпечна система, оскільки в самій архітектурі системи передбачене розмежування доступу до ресурсів. Не потрібно витратити ресурси на додаткові антивірусні програми.

Призначення "МАК"

Мультимедійний Автоматизований Комплекс (МАК) призначений для забезпечення моніторингу та керування громадським транспортом, оплати проїзду та виконання контролю всіх необхідних систем транспортного засобу. Система МАК побудована за модульним принципом і допускає різні конфігурації, залежно від потреб клієнта.

МАК – це програмно-апаратний комплекс, що складається з бортових пристроїв, які встановлені на громадському транспорті та програмної платформи МАК, на якій побудовані всі рішення по наданню послуг супутниковим GPS моніторингом та керуванню в реальному часі.

Перелік підсистем, їх призначення що реалізовані в комплексі МАК:

а) підсистема моніторингу руху громадського транспорту- забезпечує моніторинг, контроль та управління в реальному часі графіком руху ГТ з використанням супутникової системи визначення координат GPS;

б) підсистема мультимедійної інформації- забезпечує оголошення зупинок по точках GPS з супутника, показ відео зупинки та відео реклами, забезпечення послуг для людей з частковою або повною втратою зору або слуху;

в) підсистема оплати проїзду- забезпечує оплату проїзду в громадському транспорті за допомогою електронного квитка та електронної соціальної картки;

г) підсистема безпеки проїзду - забезпечує контроль швидкості руху в громадському транспорті та зв'язок з диспетчером;

д) підсистема збору і обробки інформації- забезпечує контроль всіх необхідних параметрів роботи транспортного засобу;

е) підсистема відео нагляду.

Впровадження системи МАК дає суттєвий економічний ефект що дозволяє оптимізувати роботу транспорту будь-якого міста, АТП та комунальних служб.

Так впровадження модуля «Розрахунок графіка руху» скорочує затрати праці по роботі над розробкою графіка руху для цілого міста.

Можна представити яку титанічну роботу потрібно зробити для міста з 1000 одиниць транспорту, щоб оптимально з виконанням «правильних» обідів з дотриманням Кодексу законів

про працю України, розрахувати графіки руху міського транспорту. Потрібно врахувати що графіки руху ГТ є для робочих та вихідних днів, тобто:

$$1\ 000 * 2 = 2\ 000$$

ще потрібно врахувати що графіки є сезонні: літо, зима:

$$2\ 000 * 2 = 4\ 000$$

а в місті продовжується життя, відкриваються заводи, супермаркети, школи і постійно потрібно займатися розробкою нових маршрутів та графіків руху.

Так, якщо впровадити модуль «Оплата проїзду» на підприємстві «Електротранспорту» в м. Луцьку це надасть можливість вивільнити кондукторів, що дасть економічний ефект на суму мінімум 220 000 грн. в місяць.

Впровадження цього модуля передбачає купівлю населенням міста карток Mifare, місто з населенням 200 000 чол. купить карток орієнтовано 100 000 номіналом 30 грн., а це наперед оплачувана послуга, що буде використовуватися протягом місяця як оплата проїзду.

$$100\ 000 * 30\text{грн.} = 3\ 000\ 000\ \text{грн.}$$

Можна відкривати власний банк, бо цих 3 000 000 грн. будуть завжди на рахунку фірми оператора послуги![4].

Аналіз транспортного ринку перевезень

В наведеній нижче таблиці Табл.1 подано результати пасажирських перевезень за три роки на підприємстві ВАТ «ХПАС».

Табл. 1

Місяць	Об'єм перевезень пасажирів, Q, чол.		
	2011 рік	2012 рік	2013 рік
Січень	50181	44403	31229
Лютий	37513	33756	29164
Березень	31223	38869	37881
Квітень	39046	43059	48514
Травень	44459	42388	39233
Червень	48700	48811	47834
Липень	59764	59717	58181
Серпень	57330	54485	54314
Вересень	40592	56728	45626
Жовтень	50401	47306	49311
Листопад	46296	40395	44437
Грудень	50181	45827	49725

Також наведені дані соціологічного опитування.

Опитування користувачів проводилось протягом тижня в дві зміни по 7,5 годин кожна, з 6.30 до 14.00 годин дня і з 14.00 годин дня до 21.30. Три групи, в кожній із яких було три студенти проводили опитування на зупинках в різних частинах міста. Групи отримали інструкції, самі

обирати головні і другорядні зупинки і проводити інтерв'ю в різний час. В цілому було проведено опитування більше 1200 чоловік[5].

Опитування було сфокусоване на населенні, яке працює, а тому воно було виділене в окреме спостереження. Так, кількість працюючих чоловіків, що були охоплені спостереженням, не набагато перевищує кількість працюючих жінок: 51,4% і 40,4% відповідно Рис.1.

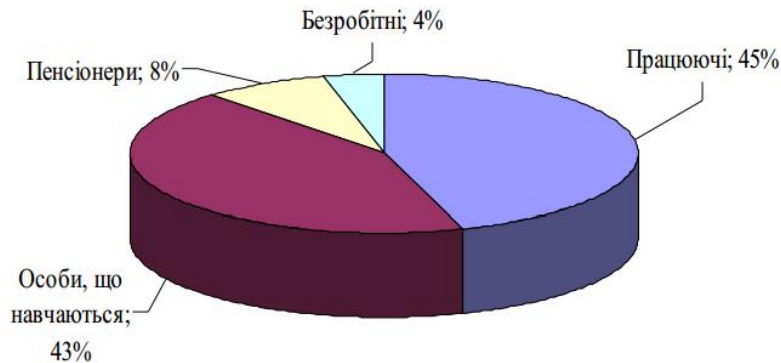


Рис.1. Соціальна структура опитаних.

Опитування користувачів показали, що жителі м. Чернігова користуються транспортом, головним чином для того, щоб доїхати до роботи (45,4% опитаних користувачів), для відвідування друзів та родичів (58%), на навчання (36,8%) і відпочинок (24,8%). На питання «До місця призначення доїжджаєте одним чи декількома видами транспорту?» респонденти відповіли наступним чином: одним – 78%; декількома – 22%, що свідчить про відсутність прямих маршрутів з віддалених мікрорайонів. У зв'язку з тим, що величина доходів досліджених категорій користувачів громадським транспортом різна, то було одержано наступні результати: 64% з числа опитаних пенсіонерів користуються

виключно тролейбусами, так як цій категорії надається пільговий проїзд; 80% з числа тих, хто навчається, для пересування використовують автобуси та маршрутні таксі. Проте, 71% з числа безробітних користуються автобусами та маршрутними таксі. Таким чином, покращення роботи транспорту загального користування (ТЗК) принесе користь людям з більш низьким рівнем доходів.

Населення, яке працює, в значній мірі, залежить від ефективності роботи громадського транспорту. В зв'язку з цим, їх потреби були проаналізовані більш детально. Одночасно біля 66% працюючих при поїзді використовують автобуси і маршрутні таксі хоча б частково. Для 12% працюючих немає принципового значення, яким транспортом добиратися. Вони чекають, «що приїде першим». Маршрутні таксомотори використовуються найчастіше у порівнянні з тролейбусами і автобусами великої місткості. Понад 30% користується маршрутними таксомоторами, 21,5% - автобусами, щоб доїхати до місця призначення, і лише 9,1% віддають перевагу виключно тролейбусам.

#### Висновок.

1. Використання хмарного сервера для транспортної логістики несе за собою багато нововведень та покращень. Дана система досить привабливою на ринку в вагу того що вона дозволяє збільшувати якість перевезень, та дозволяє віддалено за усім наглядати, вчасно вносячи необхідні корективи.
2. Було розглянуто систему відслідковування руху маршрутних таксі, проте це не єдина сфера застосування даної технології. Її можна використовувати для транспортування вантажів, для відслідковування стану роботи інших систем та ін.
3. У майбутньому, практично усі транспортні потоки будуть реєструватись на хмарному сервері. Це дозволить значно підвищити пропускну здатність доріг, дасть можливість в режимі онлайн регулювати ними. Тож можна сказати що дана технологія є досить потребою на ринку та буде лиш розширюватись на різні сфери.

Дані матеріали приведені на основі опитувань різних міст України та відображені відповідними таблицею та інфографікою.

1. Формування ринкової привабливості. Режим доступу : [http://dspace.udpu.org.ua:8080/jspui/bitstream/6789/104/1/Formuvannia\\_rynkovoi\\_pryvablyvosti\\_yak\\_neobkhidna\\_u\\_mova\\_zabezpechennia\\_konkurentospromozhnosti\\_pidpriemstva.pdf](http://dspace.udpu.org.ua:8080/jspui/bitstream/6789/104/1/Formuvannia_rynkovoi_pryvablyvosti_yak_neobkhidna_u_mova_zabezpechennia_konkurentospromozhnosti_pidpriemstva.pdf)
2. Ринкова привабливість: сутність, характеристики. Режим доступу : <http://www.economy.nayka.com.ua/?op=1&z=3083>
3. Транспортна логістика. Режим доступу : [http://pidruchniki.ws/1356061550939/logistika/transportna\\_logistika](http://pidruchniki.ws/1356061550939/logistika/transportna_logistika)
4. Система «МАК» Режим доступу : <http://www.mak.lutsk.ua/>
5. Наукова стаття. Базилюк А.В., Національний транспортний університет,Київ, «МЕТОДИКА SERVQUAL ЯК ІНСТРУМЕНТ ОЦІНЮВАННЯ ЯКОСТІ ПОСЛУГ З ПАСАЖИРСЬКИХ ПЕРЕВЕЗЕНЬ МІСТА».с. 219-230

УДК 371.13.001.76

Мельник В.М. к.ф.-м.н. доц., Клімчук Н.О.  
Луцький Національний технічний університет

## ОСНОВНІ КОНЦЕПЦІЇ ОБ'ЄКТНО-ОРІЄНТОВАНИХ БАЗ ДАНИХ ТА ЇХ ПРОБЛЕМАТИКА

**Мельник В.М., Клімчук Н.О. Основні концепції об'єктно-орієнтованих баз даних та їх проблематика.** У статті розкрито основні вимоги, яким повинна задовольняти база даних, щоб бути кваліфікованою як об'єктно-орієнтована база даних. Було розглянуто проблематику та протиріччя, що виникають на шляху узгодження специфікації об'єктно-орієнтованої моделі та пошуку єдиної системи керування об'єктно-орієнтованою базою даних. Також проаналізовано існуючі вимоги до об'єктно-орієнтованих баз даних.

**Ключові слова:** властивість, клас, специфікація моделі даних, системи керування, об'єктно-орієнтовані бази даних.

**Мельник В.М., Клімчук Н.А. Основные концепции объектно-ориентированных баз данных и их проблематика.** В статье раскрыты основные требования, которым должна удовлетворять база данных, чтобы быть квалифицированной как объектно-ориентированная база данных. Была рассмотрена проблематика и противоречия, возникающие на пути согласования спецификации объектно-ориентированной модели и поиска единой системы управления объектно-ориентированной базой данных. Также проанализированы существующие требования к объектно-ориентированным базам данных.

**Ключевые слова:** свойство, класс, спецификация модели данных, системы управления, объектно-ориентированные базы данных.

**Melnyk V., Klimchuk N. Basic concepts of object-oriented databases and their problems.** The article shows the basic requirements that must satisfy the database to be classified as an object-oriented database. We considers problems and conflicts that arise in the way of agreeing the specification of object-oriented model and the search for a unified system of management of object-oriented database. Also, were analyzed existing requirements for object-oriented databases.

**Keywords:** property, class, specification data model, management systems, object-oriented databases.

**Постановка проблеми.** Об'єктно-орієнтовані бази даних (ООБД) – це нове, третє, покоління баз даних, що привертає все більшу увагу як дослідників і розробників, так і потенційних користувачів з прикладних областей. Це зумовлено тим, що багато інформаційних систем вже «доросли» до такого рівня, що вимагають використання відповідних баз даних, які зможуть справлятися зі складно структурованими даними.

Системи другого покоління – реляційні бази даних, не можуть задовольнити цієї потреби. Для прикладу візьмемо видавничу систему за допомогою якої клієнт може створити макет газети, а потім роздрукувати її. Для роботи такого додатку потрібно зберігати фрагменти тексту, графіку, піктограми і багато інших типів елементів даних, що зустрічаються у більшій частині гіпертекстових середовищ. Підтримка таких елементів даних системами другого покоління зазвичай пов'язана з немалими труднощами. [8]

Напрямок ООБД виник порівняно давно. Публікації з'являлися вже в середині 1980-х років. Однак найбільш активно цей напрямок розвивається в останні роки [4, с.194]. З одного боку це пояснюється розвитком і впровадженням в практику об'єктно-орієнтованого підходу в цілому. Але з іншого боку, інтуїтивно ясно, що максимальний ефект можна отримати саме від використання об'єктно-орієнтованих баз даних. Утім, на даний час не сформована єдина думка про те, що собою являє об'єктно-орієнтована система, не кажучи уже про об'єктно-орієнтовані бази даних. Немає загальної згоди відносно якоїсь одної існуючих моделей даних, відсутня теоретична база.

Як зазначають багато дослідників, своїм успіхом реляційна модель даних багато в чому зобов'язана тому, що спиралася на строгий математичний апарат теорії множин, відносин і логіки першого порядку [4, с.197]. Основні труднощі об'єктно-орієнтованого моделювання даних виникають через те, що такого розвиненого математичного апарату, на який могла б спиратися загальна об'єктно-орієнтована модель даних, не існує. Великою мірою і досі несформована базова об'єктно-орієнтована модель. З іншого боку, в роботах Майєра стверджується, що загальна об'єктно-орієнтована модель даних в класичному сенсі і не може бути визначена з причини непридатності класичного поняття моделі даних до парадигми об'єктної орієнтованості [4, с.198].

За допомогою об'єктно-орієнтованих баз даних вирішуються дві проблеми. По-перше, складні інформаційні структури виражаються в них краще, ніж у реляційних базах даних, а по-друге, усувається необхідність транслювати дані з того формату, що підтримується в системах керування базами даних (СКБД). Наприклад, у реляційній СКБД розмірність цілих чисел може становити 11 цифр, а у використовуваній мові програмування – 16. Програмістові прийдеється враховувати цю ситуацію [1].

Проте суттєвим недоліком об'єктно-орієнтованих баз даних є їхні тісні зв'язки із застосовуваною мовою програмування. До даних, що зберігаються в реляційній СКБД, можуть звертатися будь-які додатки, наприклад, Java-об'єкт, поміщений в об'єктно-орієнтовану базу даних, буде становити інтерес лише для додатків, написаних на Java [1].

До теперішнього моменту нам невідома будь-яка мова програмування ООБД, яка була б спроектована цілком заново, починаючи з нуля. Природним підходом до побудови такої мови було використання (з необхідними розширеннями) деякої існуючої об'єктно-орієнтованої мови. Практично усі розробники усвідомлюють, що існує потреба підтримки не тільки мови програмування ООБД, а й розвиненої мови запитів. Система повинна підтримувати легко освоюваний інтерфейс, прямо доступний кінцевому користувачеві в інтерактивному режимі [4, с. 202].

**Аналіз досліджень.** Проаналізувавши літературу на тему баз даних, в розділах з описами моделей даних – ієрархічна, реляційна, об'єктно-орієнтована моделі даних, останній приділяється найменше уваги. Це наслідок того, що відсутня теоретична база, не говорячи вже про єдину специфікацію об'єктно-орієнтованої моделі даних.

У період з 1989 по 1995 рр. авторські групи, включаючи відомих фахівців в області баз даних, підготували і опублікували три документи, які відображали точки зору авторів щодо перспектив розвитку технології баз даних. З легкої руки авторів ці документи отримали назву маніфестів, що, загалом, відображало їх суть: у кожному з документів проголошувався набір ідей і вимог, на яких, на думку авторів, повинні бути базуватися системи баз даних наступного покоління. Саме ці три документи – Маніфести, було взято за теоретичну базу для написання даної статті.

Цікаво відзначити відмінності між колективами авторів кожного з маніфестів. «Маніфест систем об'єктно-орієнтованих баз даних» (Перший маніфест) написаний академічними дослідниками; майже всі вони є професорами різних університетів. Стиль Першого маніфесту – дуже м'який і помірно рекомендаційний.

Автори документу «Системи баз даних третього покоління: Маніфест» (Другого маніфесту) були представниками індустрії. Другий маніфест написаний в більш жорсткому стилі і багато в чому спрямований на захист інвестицій великих компаній-виробників програмного забезпечення SQL-орієнтованих СУБД. Звичайно, Другий маніфест багато в чому являв собою реакцію індустрії на революційні пропозиції Першого маніфесту.

«Третій маніфест» був одночасно найбільш консервативним і найбільш радикальним. Консервативність Третього маніфесту полягає в тому, що його автори усіма силами стверджують необхідність і достатність використання в системах баз даних наступного покоління класичної реляційної моделі даних. Радикальність полягає в тому, що автори повністю заперечують підходи, пропоновані в перших двох маніфестах, розцінюючи їх як необґрунтовані, погано опрацьовані, надлишкові і навіть шкідливі (за винятком однієї спільної ідеї про потреби забезпечення розвиненої системи типів); фактично, автори повністю відкидають технологію, створену індустрією баз даних за останніх 25 років, і пропонують повернутися до витоків реляційної моделі даних, тобто до початкових статей Е. Кодда [2].

В 1991 році була сформована група ODMG – Object Database Management Group (група управління об'єктними базами даних). Дана група об'єднувала розробників об'єктних СУБД та інших зацікавлених організацій для розробки стандарту об'єктно-орієнтованих СУБД. В 1993 році був опублікований стандарт ODMG-1. Остання версія стандарту була випущена в 1999 році – ODMG-3. Архітектура об'єктної моделі в стандарті ODMG-3 передбачала наявність єдиного сховища даних, до якого користувач може здійснювати різні види доступу. Сховище даних доступно з мови визначення даних, мови запитів і ряду мов маніпулювання даними.



Центральною в архітектурі є модель даних, що представляє організаційну структуру, в якій зберігаються всі дані, керовані ООСУБД. Мова визначення об'єктів, мова запитів і мови маніпулювання розроблені таким чином, що всі їхні можливості спираються на модель даних. Архітектура допускає існування різноманітних реалізаційних структур для зберігання модельованих даних, але важливою вимогою є те, що всі програмні бібліотеки і всі інструментальні засоби забезпечуються в об'єктно-орієнтованих рамках і повинні зберігатися в узгодженні з даними.

Проте, об'єктна модель, що представлена в стандарті ODMG-3 має ряд недоліків, зокрема: об'єктна модель недостатньо обґрунтована. Виникає відчуття, що в цю модель прагнули помістити властивості реалізаційних моделей всіх комерційних ООСУБД, що існували до моменту написання стандарту; поділ мов визначення даних, об'єктних запитів і мов маніпулювання, звичайно, відображає прагнення розробників стандарту до ортогоналізації моделі, але дуже ускладнює її цілісне розуміння; мова об'єктних запитів є простою з концептуальної точки зору, але дуже складною для використання початківцями.

На ринку ООСУБД присутні три заслужені системи, кожна з яких існує близько 20 років, – Versant компанії Versant Corporation, ObjectStore, що належить з 2003 р. компанії Progress Software, і Objectivity/DB компанії Objectivity Inc. У всіх цих системах підтримується об'єктно-орієнтована модель даних ODMG-3, але архітектури систем значно різняться.

На сьогоднішній день ведуться багаточисленні експериментальні розробки: люди реально будують системи. Деякі з них являються тільки прототипами, а деякі – вже стали комерційними продуктами. З кожним роком збільшується число публікацій і реалізованих комерційних та експериментальних систем. Та це ніяк не впливає на прогресування процесу формування єдиної моделі об'єктно-орієнтованих баз даних.

На даний момент, коли теперішнє суспільство технічно ще не готове перейти до використання об'єктно-орієнтованих баз даних та середовищ їх управління, потрібно реально оцінити та проаналізувати можливості існуючих СУБД. Можливо, це зможе сприяти подальшій еволюції реляційних баз даних до нового, третього покоління – об'єктних баз даних.

Виходячи з цього, **мета** статті полягає у аналізі вимог та властивостей, які представлені у трьох Маніфестах, володіючи якими база даних та середовище керування базами даних вважатимуться об'єктно-орієнтованими. На основі визначених властивостей і вимог у даній статті, буде проводитись дослідження існуючих баз даних та їх середовищ створення на предмет об'єктної орієнтованості.

#### **Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.**

Об'єктно-орієнтовані системи беруть свій початок з об'єктно-орієнтованих мов програмування. Основна ідея, що об'єднує ці дві області, полягає в тому, щоб відгородити користувача від конструкцій, що пов'язані з апаратним забезпеченням, таких як біти і байти. Замість цього, користувач має справу з об'єктами і операціями над цими об'єктами, які більше відповідають своїм аналогам в реальному світі. [6, с.944]

Посилаючись на вміст Першого Маніфесту, властивості і вимоги, якими повинна володіти об'єктно-орієнтована база даних, можна розділити на три категорії:

1. **обов'язкові** («золоті» правила) (властивості, що мають забезпечувати у системі об'єктно-орієнтованість);
2. **необов'язкові** (ті, які будуть покращувати систему);
3. **відкриті** (можливість вибору проектувальником рішення певного завдання різними способами).

Таблиця 1 – Категорії властивостей та характеристик ООБД, запропоновані у «Маніфесті систем об'єктно-орієнтованих баз даних»

<b>Обов'язкові можливості («золоті» правила)</b>	
1.	Система повинна бути СУБД:
1.1	стабільність;
1.2	управління вторинною пам'яттю;
1.3	паралелізм;
1.4	відновлюваність;

1.5 засоби забезпечення незапланованих запитів.
2. Система повинна бути об'єктно-орієнтованою, використовуючи при цьому сучасні об'єктно-орієнтовані мови програмування: <ul style="list-style-type: none"><li>2.1 складні об'єкти;</li><li>2.2 ідентифікація об'єктів;</li><li>2.3 інкапсуляція;</li><li>2.4 типи чи класи;</li><li>2.5 успадкування;</li><li>2.6 перекриття методів спільно з пізнім зв'язуванням;</li><li>2.7 розширюваність;</li><li>2.8 обчислювальна повнота.</li></ul>

Продовження таблиці 1

<b>Необов'язкові можливості</b>
1. множинне успадкування 2. перевірка і виведення типів 3. розподілюваність 4. проектні транзакції 5. версії
<b>Відкриті можливості</b>
1. парадигма програмування 2. системне представлення 3. система типів 4. однорідність

Система об'єктно-орієнтованих баз даних повинна задовольняти двом критеріям: повинна бути СУБД і при цьому являтися об'єктно-орієнтованою системою, на максимально можливому ступені знаходитись на рівні сучасних об'єктно-орієнтованих мов програмування. Реляційні системи баз даних не задовольняють правилам з пункту 2.1 по пункт 2.8. [7]

У другому Маніфесті представлено три основні принципи, якими потрібно керуватися при створенні систем третього покоління.

*Принцип 1. Крім традиційних послуг по управлінню даними, СУБД третього покоління забезпечать підтримку більш «багатих» структур об'єктів і правил. Більш «багата» структура об'єктів характеризує засоби, що необхідні для збереження і маніпулювання нетрадиційними елементами даних – тексти, зображення, просторові дані. Творцям додатків потрібно забезпечити можливість задавати групу правил, що стосуються елементів даних, записів і наборів.*

*Принцип 2. СУБД третього покоління повинні включати в себе СУБД другого покоління. Іншими словами, системи другого покоління внесли вирішальний вклад в двох областях: непроцедурний доступ та незалежність даних. І ці досягнення не повинні бути відкинени системами третього покоління. Проте існує точка зору, що є додатки, яким ніколи не знадобиться виконувати запити через властиву їм простоту доступу до СУБД.*

*Принцип 3. СУБД третього покоління повинні бути відкритими для інших підсистем. Інакше кажучи, будь-яка СУБД, що розраховує на широку сферу застосування, повинна бути оснащена мовою четвертого покоління (4GL), різноманітними інструментами підтримки прийняття рішень, дружнім доступом із багатьох мов програмування, дружнім доступом із популярних підсистем, інтерфейсами з графічними бізнес-пакетами, можливістю запуску додатків з бази даних на другій машині і з розподіленої СУБД.*

Крім цього, представлені 13 пропозицій, в яких вимоги до нових систем обговорені більш детально:

1. Система типів СУБД третього покоління повинна бути «багатою» і різноманітною.
2. Наслідування – хороша ідея.
3. Функції, в тому числі процедури і методи баз даних, і інкапсуляція – хороші ідеї.
4. Унікальні ідентифікатори (UID) записів повинні задаватися СУБД тільки в тому випадку, коли недоступний визначений користувачем первинний ключ.
5. Правила (тригери, обмеження) стануть одною із ключових характеристик майбутніх систем.
6. Всі програмуючі доступи до баз даних повинні виконуватись через не процедурну мову доступу високого рівня.
7. Повинно бути по крайній мірі два способи специфікації наборів: за допомогою перерахування членів і шляхом використання мови запитів для задання членів.
8. Суттєвою є наявність оновлюваних уявлень.
9. Показники продуктивності не мають нічого спільного з моделями даних і не повинні в них проявлятися.

10. СУБД третього покоління повинні бути доступні для різноманітних мов програмування високого рівня.
11. Мова з підтримкою стабільних даних – хороша ідея.
12. SQL, як універсальний спосіб вираження запитів.
13. Запити і відповіді на них повинні формувати нижній рівень комунікацій між клієнтом і сервером.

Третій маніфест стосується майбутнього систем управління даними і СУБД. Назва маніфесту обумовлена тим, що він слідує за двома попередніми маніфестами. Автори даного маніфесту твердо впевнені в тому, що в будь-якій спробі рухатись вперед, для того щоб витримати випробування часом, потрібно відкинути мову запитів SQL та реляційну модель взагалі. Проте вказано що потрібно робити із спадщиною SQL та реляційною моделлю. Автори Третього маніфесту винесли на розсуд читачів власні конкретні приписи та заборони, а також вельми наполегливі пропозиції, що стосуються систем управління даними і СУБД.

Перед тим як представити винесені пропозиції у Третьому маніфесті, доцільно уточнити певні поняття, які трактуються авторами в даному документі.

Поняття **домен** і **тип даних** – синонімічні поняття.

Узагальнено значення домену називають **скалярними значеннями** або просто **скалярами**.

Терміни **операція** і **функція** – синонімічні.

**Кортеж** – це елемент відносин, тобто рядок таблиці.

Таблиця 2 – Приписи, заборони та пропозиції, що представлені у Третьому Маніфесті

Приписи	Заборони	Наполегливі пропозиції
1	2	3
<b>Реляційної моделі</b>		
1. Домени 2. Типізовані скаляри 3. Скалярні оператори 4. Реальне уявлення 5. Істинність значень 6. Конструктор типів 7. Оператор порівняння 8. Кортежі 9. Відносини 10. Скалярні і кортежні змінні 11. Змінні відносин 12. Транзакції 13. Операції створення/руйнування 14. Реляційна алгебра 15. Імена і явні значення відносин 16. Реляційні функції 17. Присвоєння відносин і кортежів 18. Порівняння 19. Обмеження цілісності 20. Предикати відносин і баз даних 21. Каталог	1. Ніякого впорядкування атрибутів 2. Ніякого впорядкування кортежів 3. Ніяких дублікатів кортежів 4. Ніяких невизначених значень 5. Ніяких помилок, пов'язаних з порожніми відносинами 6. Ніяких конструкцій внутрішнього рівня 7. Ніяких операцій на рівні кортежів 8. Ніяких зіставних стовпців 9. Ніяких придушень перевірки доменів 10. Не SQL	1. Можливі ключі для похідних 2. Системно-генеровані ключі 3. Посилальна цілісність 4. Висновок можливих ключів 5. Запити з квотою 6. Транзитивне замикання 7. Параметри кортежів і відносин 8. Значення за замовчуванням 9. Міграція SQL

22. Розробка мови		
-------------------	--	--

Продовження таблиці 2

1	2	3
<b>Ортогональні</b>		
1. Перевірка типів на стадії виконання 2. Просте спадкування (умовне) 3. Множинне успадкування (умовне) 4. Обчислювальна повнота 5. Явні границі транзакцій 6. Вкладені транзакції 7. Агрегати і порожні множини	1. Ніяких ідентифікаторів об'єктів 2. Ніяких «публічних змінних екземпляра» 3. Ніяких «захищених змінних екземпляра» або «друзів»	1. Спадкування типів 2. Конструктори типів-колекцій 3. Конвертація в і з відносин 4. Однорівневе зберігання

**Висновки та перспективи подальшого дослідження.** Отже, зважаючи на вищевказані пропозиції та вимоги Маніфестів, можна дійти до загального висновку, що об'єктно-орієнтовані бази даних повинні зберігати всі можливості об'єктно-орієнтованої мови програмування і забезпечувати створення і довготривале зберігання об'єктів у зовнішній пам'яті. Об'єктно-орієнтовані СУБД повинні забезпечувати довготривале зберігання об'єктів, адекватне внутрішнє представлення об'єктів, що відносяться до різних типів даних, паралелізм в обробці різних типів об'єктів, виконання всіх методів, успадкованих об'єктом. Такі бази даних називають об'єктно-орієнтованими в повному розумінні об'єктної парадигми.

В об'єктних СУБД велике майбутнє, перш за все, тому, що структура об'єктних баз даних більш точно відповідає предметній області, чим структура реляційних баз даних. Надзвичайно важливо прийти до спільної згоди щодо визначення системи об'єктно-орієнтованих баз даних та єдиної специфікації об'єктної моделі баз даних, оскільки це нове покоління, що може прийти на заміну реляційній моделі. На шляху до цієї мети, виникає багато проблем, серед них можна виділити три характерні риси сучасного стану досліджень у цій галузі:

- відсутність загальноприйнятої моделі даних;
- відсутність єдиної формальної теорії;
- активна експериментаторська діяльність.

Одною із переваг ООСУБД являється їх можливість інтеграції з об'єктно-орієнтованими мовами програмування. В результаті всі складові інформаційної системи зможуть конструюватись з єдиних позицій і в єдиному стилі.

В даний час перспектива об'єктних моделей баз даних дуже велика. Існує багато галузей діяльності людини в якій саме об'єктні моделі можуть якісно і з меншою проблематичністю допомогти. Та, на жаль, об'єктно-орієнтовані бази даних ще не настільки використовувані, щоб перерости і витіснити реляційні бази даних. Не останню роль у практичній відсутності досліджень з ООБД у нашій країні відіграє часто невинуватим критика об'єктно-орієнтованого програмування в цілому.

1. <http://www.ua5.org/database/118-obktno-orntovan-bazi-danikh.html>
2. Дейт К.Дж. Введение в системы баз данных. - К.: Диалектика, 1998. - 784 с.
3. <http://www.simulation.kiev.ua/dbis/lection30.html>
4. Кузнецов С.Д. Основы современных баз данных.
5. <http://ukrbukva.net/print/page,1.35438-Manifest-sistem-ob-ektno-orientirovannyh-baz-dannyh.html>
6. Хомоненко А.Д. Базы данных: учебник для вузов по технич. и экономич. спец. / [А. Д. Хомоненко, В. М. Цыганков, М. Г. Мальцев] ; под ред. А. Д. Хомоненко. - Санкт-Петербург : КОРОНА принт, 2002. - 672 с.
7. «Манифест систем объектно-ориентированных баз данных» М. Атkinson, Ф. Бансилон, Д. ДеВитт, К. Диттрих, Д. Майер, С.Здоник, Перевод: М.Р. Когаловский, Источник: журнал Системы Управления Базами Данных # 4/1995, издательский дом «Открытые системы» Новая редакция: Сергей Кузнецов, 2009 г.
8. Системы баз данных третьего поколения: Манифест. (Второй Манифест)
9. Третий Манифест, Х. Дарвин и К. Дэйт, Перевод: М.Р. Когаловский, Источник: журнал Системы Управления Базами Данных #1/1996, Издательский дом «Открытые системы».



УДК 004.773:004.051(045)

Мельник В.М. к.ф.-м.н. доц.

Луцький національний технічний університет

## ВИКОРИСТАННЯ ЗВИЧАЙНИХ СОКЕТІВ API ДЛЯ ФАЙЛОВИХ СИСТЕМ ТИПУ HADOOP

**Мельник В.М. Використання звичайних сокетів API для файлових систем типу Hadoop.** Hadoop – це реалізація (каркас) для програмування розподілених обчислень з відкритим вихідним кодом яка базується на моделі MapReduce. Вона спочатку інтегрується з Hadoop Distributed File System (HDFS), тобто розподіленою файловою системою на рівні користувача. У статті описується оболонка-агностик API для файлової системи Hadoop (HFAA), щоб дати змогу їй інтегруватися з будь-якою розподіленою файловою системою через TCP-сокетну взаємодію. За допомогою такого API HDFS можна замінити будь-якою розподіленою файловою системою, такою як PVFS, Ceph, Lustre або іншою, дозволяючи прямим способом здійснювати співставлення з точки зору періодизації продуктивності і масштабованості. На відміну від попередніх спроб підвищення взаємодії системи Hadoop з новими файловими системами представлений сокет API усуває необхідність налаштування Hadoop-реалізації для Java і повертає реалізацію обов'язків її на саму файлову систему. Таким чином, розробники, бажаючи інтегрувати їх нову (іншу) файлову систему з Hadoop не несуть відповідальності за розуміння деталі внутрішньої роботи Hadoop в під'єднаній файловій системі.

В даному випадку початкова реалізація HFAA використовується для заміни HDFS файловою системою PVFS, популярною в високопродуктивних обчислювальних середовищах. Порівняно з альтернативним методом інтеграції з PVFS (інтерфейсу POSIX-ядра), HFAA збільшує пропускну здатність запису і читання.

**Ключові слова:** Сокет API, Hadoop файлова система, розподілена обчислювальна реалізація Hadoop в Java

**Мельник В.М. Использование обычных сокетов API для файловых систем типа Hadoop.** Hadoop – это реализация (каркас) для программирования распределенных вычислений с открытым исходным кодом, основанная на модели MapReduce. Она сначала интегрируется с Hadoop Distributed File System (HDFS), т.е. распределенной файловой системой на уровне пользователя. В статье описывается оболочка-агностик API для файловой системы Hadoop (HFAA), чтобы дать возможность ей интегрироваться с любой распределенной файловой системой через TCP-сокетное взаимодействие. С помощью такого API HDFS можно заменить любой распределенной файловой системой, такой как PVFS, Ceph, Lustre или другой, позволяя прямым способом осуществлять сопоставление с точки зрения периодизации производительности и масштабируемости. В отличие от предыдущих попыток повышения взаимодействия системы Hadoop с новыми файловыми системами представлен API-сокет, устраняющий необходимость настройки Hadoop-реализации для Java и возвращающий реализацию ее обязанностей на саму файловую систему. Таким образом, разработчики, желающие интегрировать их новую (другую) файловую систему с Hadoop не несут ответственности за понимание детали внутренней работы Hadoop в подключенной файловой системе.

В данном случае начальная реализация HFAA используется для замены HDFS файловой системой PVFS, популярной в высокопроизводительных вычислительных средах. По сравнению с альтернативным методом интеграции с PVFS (интерфейса POSIX-ядра), HFAA увеличивает пропускную способность записи и чтения.

**Ключевые слова:** Сокет API, Hadoop файловая система, распределенная вычислительная реализация Hadoop в Java

**Melnyk V. Using a conventional socket API for hadoop file systems.** Hadoop is an implementation (frame) for distributed computing programming with open source code, based on MapReduce model. It initially is integrated with the Hadoop Distributed File System (HDFS) at the user level. In this article the Hadoop Filesystem Agnostic API (HFAA) is introduced to allow Hadoop to integrate with any distributed file system over TCP sockets. With this API, HDFS can be replaced by distributed file systems such as PVFS, Ceph, Lustre, or others, thereby allowing direct comparisons in terms of performance and scalability. Unlike previous attempts at augmenting Hadoop with new file systems, the socket API presented here eliminates the need to customize Hadoop's Java implementation, and instead moves the implementation responsibilities to the file system itself. Thus, developers wishing to integrate their new file system with Hadoop are not responsible for understanding details of Hadoop's internal operation.

In this case, an initial implementation of HFAA is used to replace HDFS with PVFS, a file system popular in highperformance computing environments. Compared with an alternate method of integrating with PVFS (a POSIX kernel interface), HFAA increases write and read throughput.

**Keywords:** socket API, Hadoop file system, Distributed computing, Hadoop's Java implementation

### Вступ

До складу *Hadoop Common* входять бібліотеки управління файловими системами, що підтримуються Hadoop, сценарії створення необхідної інфраструктури та управління розподіленою обробкою даних, для вигідності виконання яких створений спеціалізований спрощений інтерпретатор командного рядка, що стартує із оболонки операційної системи через відповідну команду.

Hadoop MapReduce являється програмним каркасом для програмування розподілених обчислень в рамках парадигми MapReduce [1]. Розробники додатка для Hadoop MapReduce



необхідно реалізувати базовий обробник, який на кожному вузлі обчислення кластера забезпечує перетворення вихідних пар «ключ-значення» в проміжний набір пар «ключ – значення» (клас, який реалізує інтерфейс Mapper, носить ім'я за функцією вищого порядку Map), і обробник, який зводить проміжний набір пар в результуючий, скорочений набір (клас, що реалізує інтерфейс Reducer). Каркас передає на вхід такого скорочення відсортовані виведення з базових обробників. Зведення складається з трьох фаз: *shuffle* (масування, виділення потрібної секції виводу), *sort* (сортування. Групування за ключами виведення від розподільників до сортування, яке необхідне в випадку, коли різні атомарні обробники повертають набір із однаковими ключами. При цьому, правила сортування на даній фазі можуть бути задані програмно і використовувати будь-які особливості внутрішньої структури ключів) і, власне, *reduce* (скорочення списку) – для отримання результуючого набору. Для деяких видів обробки стиснення не потрібне, і каркас повертає в цьому випадку набір відсортованих пар, отриманих базовими обробниками.

Hadoop MapReduce дозволяє створювати завдання як з базовими обробниками, так і зі стисненнями [1,2], написаними без використання Java: утиліти *Hadoop streaming* дозволяють використовувати в якості базових обробників і стиснень (зверток) будь-який файл виконання, який працює зі стандартним вводом-виводом операційної системи (наприклад, утиліти командної оболонки UNIX), є також SWING-сумісним прикладний інтерфейс програмування *Hadoop pipes* на C++. Також, в склад дистрибутивів Hadoop входять реалізації різних конкретних базових обробників і зверток, що найбільш типічно використовуються в розподіленій обробці даних.

В перших версіях Hadoop MapReduce включав спеціальний засіб планування завдань (*JobTracker*) і вже з версії 2.0 ця функція перенесена в YARN. З цієї версії модуль Hadoop MapReduce реалізований поверх YARN. Програмні інтерфейси в більшості збережені, однак повна зворотня сумісність не досягнута (тобто для запуску програм, написаних для попередніх версій API для роботи в YARN в загальному потрібна їх модифікація або рефакторинг, і тільки при певних обмеженнях можливі варіанти зворотної двійкової сумісності).

**HDFS** (*Hadoop Distributed File System*) – це файлова система, призначена для збереження файлів великих розмірів, поблоково розподілених між вузлами кластера обчислення. Всі блоки в HDFS (крім останнього блоку файлу) мають однаковий розмір, і кожен блок може бути розміщений на декількох вузлах, розмір блоку і коефіцієнт реплікації (кількість вузлів, на яких повинен бути розміщений кожен блок) визначаються в налаштуваннях на рівні файлу. Завдяки реплікації забезпечується стійкість розподіленої системи до збоїв окремих вузлів. Файли в HDFS можуть бути записані лише один раз (модифікація не підтримується), а запис в файл в одну мить часу може вести тільки один процес. Організація файлів в просторі імен – ієрархічна: наявний кореневий каталог, підтримується створення підкаталогів, в одному каталозі можуть бути розміщені файли і підкаталоги. Одже, Hadoop це базова основа з відкритим кодом, реалізуюча модель MapReduce для паралельного програмування [1,2]. Каркас Hadoop складається з двигуна MapReduce і файлової системи на рівні користувача. Для переносимості і простоти установки обидва компонента написані на Java і вимагають всього лиш стандартного обладнання. В останні роки, Hadoop став популярним в промисловості та академічних колах [3]. Наприклад, на кінець 2010 року, Yahoo було більш ніж 43 000 вузлів, що працюють на Hadoop як для застосування в області досліджень так і у виробництві [4].

HDFS управляє ресурсами зберігання в кластері Hadoop, забезпечуючи глобальний доступ до будь-якого файлу [5,6] і реалізується двома службами: один центральний вузол – NameNode і багатьох вузлів – DataNodes. NameNode несе відповідальність за підтримання дерева каталогів HDFS. Клієнти звертаються до NameNode для виконання спільних операцій файлової системи, таких як відкриття, закриття, перейменування і видалення. NameNode не зберігає власних даних HDFS, а скоріше всього підтримує процес відображення між іменем файлу HDFS і вузлом NameNode, список блоків у файлі, і вузлом/вузлами DataNode, на якому/яких ці блоки зберігаються.

Хоча HDFS зберігає файл даних в розподіленому режимі, файл метаданих зберігається за допомогою централізованої служби вузла NameNode. Як це властиво для дрібних кластерів, ця конструкція запобігає Hadoop від масштабування за рахунок ресурсів єдиного вузла NameNode. Попередній аналіз CPU-процесора і необхідної пам'яті для NameNode виявив, що ця служба має обмеження в пам'яті. Великий NameNode, наприклад, з 60 GB оперативної пам'яті може зберігати

не більше 100 мільйонів файлів в середньому розміром 128 MB (еквівалент двом HDFS блокам). Крім того, з 30% цільовим навантаженням на процесор CPU для служби малої затримки подібний NameNode може підтримати кластер з 100 000 користувачів для зчитування, проте тільки 10 тисяч користувачів з операцією запису [7].

За покращення масштабованості (цю ціль також розділяє HDFS "Федерація", наближення з використанням декількох незалежних просторів імен, що зберігаються на окремих NameNodes [8]), існують й інші мотивації заміни HDFS. Наприклад, альтернативні файлові системи дозволяють операції запису і перезапису будь де (підвищення гнучкості додатків) і підтримують віддалені пересилання для віддалених додатків через мережу як необмежену смугу (збільшення продуктивності передачі даних). Поряд з масштабованістю, продуктивністю і набором функцій підтримки, одним з ключових драйверів для використання альтернативних файлових систем є лише те, що ці файлові системи вже широко розгорнуті. У багатьох високопродуктивних обчислювальних середовищах Hadoop (тобто реалізація MapReduce) є найновішою реалізацією у найрізноманітніших підходах програмування додатків, які будуть підтримуватися з акцентом на компетентність існуючої інфраструктури.

Останні роботи описують досліджували заміни HDFS з іншими розподіленими файловими системами. На сьогодні Hadoop інтегрований з Amazon S3 [9], CloudStore [10], Ceph [11], GPFS [12], Lustre [13] і PVFS [14]. Кожна з цих реалізацій вимагає код спеціального призначення для інтеграції з Hadoop, і вимагає від розробника необхідності знання внутрішньої діяльності як Hadoop так і цільової файлової системи.

В даній статті дається оцінка трьом різним підходам заміни HDFS на інші розподілені файлові системи. В першому методі використовується драйвер POSIX для безпосереднього вмонтування розподіленої файлової системи на кожному вузлі кластера. Цей драйвер зазвичай надається з файловою системою і призначений для дозволу легкої інтеграції з будь-якою немодифікованою програмою або утилітою. Однак, при використанні Hadoop страждає продуктивність зберігання системи через обмеження інтерфейсу POSIX. Наприклад, Hadoop не в змозі запросити файлову систему для отримання інформації місця знаходження файлу, і, таким чином, не може спланувати розрахунки щоб мінімізувати передачу мережевих даних.

Другий підхід, код співвставки, що розширює реалізацію Hadoop Java (шляхом забезпечення модифікації абстрактного класу зручної файлової системи системою Hadoop) для прямого розташування іншої файлової системи в просторі користувача. Такий співвкладений код є специфікою конкретної файлової системи, тобто вставка для PVFS працює тільки з PVFS-системою.

Враховуючи вищеописані обмеження, третій спосіб включає розроблений агностик API файлової системи Hadoop (Hadoop Filesystem Agnostic API – *HFAA*) для подолання масштабованих обмежень HDFS і надання можливості прямої інтеграції з альтернативними файловими системами. HFAA забезпечує універсальний, загальний інтерфейс, який дозволяє Hadoop співпрацювати з будь-якою файловою системою, підтримуючу мережеві сокети, а, особливо, файлові системи без єдиної точки відмови і більшою загальною цілеспрямованістю семантики. Його конструкція переміщає обов'язки інтеграції за межі Hadoop, і не вимагає від користувача або розробника базових знань framework для системи Hadoop MapReduce. По суті, інтеграція Hadoop зводиться до одного API.

### **Архітектура Hadoop**

Framework для системи Hadoop реалізована у вигляді двох основних служб: двигуна Hadoop MapReduce і HDFS. Вони, як правило, використовується спільно, хоча при бажанні кожен з них може працювати незалежно. Наприклад, користувачі сервісу Amazon Elastic MapReduce можуть використовувати драйвер Hadoop MapReduce в поєднанні з власним сервісом збереження для Amazon (S3) [9].

В системі Hadoop, драйвер MapReduce реалізується двома програмними сервісами: робочий трекер (JobTracker) і трекер завдань (TaskTracker). Централізований сервіс JobTracker'а працює на виділеному вузлі кластера і несе відповідальність за розщеплення вхідних даних на частини для обробки незалежною картою і зменшенням задач (шляхом координатії на рівні користувача з файловою системою), планування кожного завдання на вузлі кластера для його виконання, моніторингу процесу виконання, отримуючи сигнали підтвердження від вузлів кластера, і

відновлення після збоїв за допомогою повторного запуску завдань. На кожному вузлі кластера примірник служби TaskTracker'a отримує карту, знижує задачі з JobTracker'a, виконує завдання і повідомляє про цей стан назад у JobTracker.

HDFS забезпечує глобальний доступ до будь-якого файлу в розділювальному просторі імен через кластер Hadoop [5,6]. HDFS реалізується двома службами: NameNode і DataNode. Реалізація HFAA сумісна з OrangeFS (вилкою PVFS) і порівнює його продуктивність по відношенню до інших підходів. Нарешті, в розділі 6 обговорюються споріднені роботи в нашому підході.

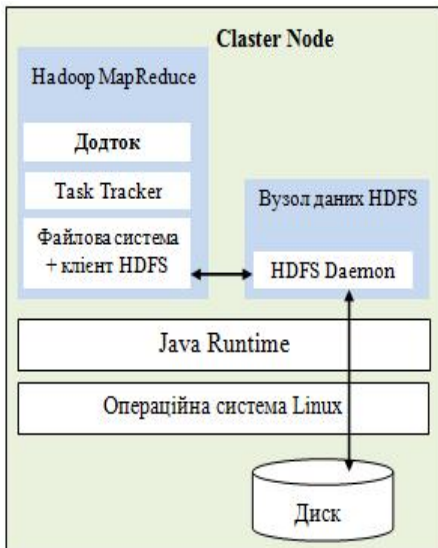


Рис. 1 – Кластерний вузол Hadoop

Іменовані вузли NameNode відповідає за підтримання дерева каталогів HDFS, і являється централізованою службою в кластері, працюючому з єдиним вузлом. Клієнти звертаються до іменованого вузла для виконання звичайних операцій файлової системи, таких як відкриття, закриття, перейменування і видалення. Цей вузол не зберігає самих даних HDFS, а скоріше підтримує відображення між іменем файлу HDFS, списком блоків у файлі і вузлом/вузлами даних, на якому/яких ці блоки зберігаються.

До того ж, до централізованого іменованого вузла всі інші вузли кластера обслуговують вузол даних. Кожен вузол даних зберігає блоки HDFS (64MB, фрагмент пам'яті одного логічного файлу) від локальних або віддалених клієнтів. Кожен блок зберігається у вигляді окремого файлу в локальній файлової системі вузла, який використовує власну файловою систему, таку як ext4. Блоки створюються або знищуються на вузлі даних за запитом іменованого вузла, який перевіряє і обробляє запити від клієнтів. Хоча іменовані вузли управляє простором імен, клієнти все ж безпосередньо спілкуються з вузлом даних для читання або запису даних на рівні блоків HDFS.

На рисунку 1 показана архітектура стандартного вузла Hadoop кластера, що використовується для обчислень і збереження. Двигун Map-Reduce (працюючий у віртуальній машині Java) виконує додаток користувача. Коли додаток зчитує або записує дані, запити передаються через клас `Hadoop org.apache.hadoop.fs.FileSystem`, який забезпечує стандартний інтерфейс для розподілених файлової систем, в тому числі HDFS за замовчуванням. Клієнт HDFS є все ж відповідальним за отримання даних від розподіленої файлової системи через з'єднання з DataNode з необхідним блоком. У загальному випадку DataNode працює на тому ж вузлі, а тому не потрібно ніякого зовнішнього мережевого трафіку. DataNode, також працюючи у віртуальній машині Java, доступується до даних, що зберігаються на локальному диску, використовуючи звичайний файл функцій вводу/виводу.

#### Масштабовані файлові системи

Існує багато розподілених файлової систем, створених для масштабування через великій комп'ютерний кластер. Деякі з них є відкритим вихідним кодом, у тому числі Ceph [15], Lustre [16], і PVFS [17], в той час як інші є власними, в тому числі GPFS [18], файлової системи Google [19,20], і PanFS [21].

Функція	HDFS	Lustre	Ceph	PVES
Простір імен	Централізований	Централізований	Розподілений	Розподілений
Витримки помилок	Звернення до файла та авто-оновлення	Апаратна залежність	Звернення до файла та авто-оновлення	Апаратна залежність
Симантики запису	Єдиний запис (забезпечується підтримка додавання інформації до файла)	Запис і перезапис будь-де з участю більшості POSIX-семантик		
Доступність	Звичайні API Hadoop або драйвер ядра (відмінний від POSIX)	Звичайний (Lustre/ Ceph/PVFS) API або драйвер ядра, забезпечений POSIX-подібною функціональністю		
Розвернута модель	Вузли, що використовуються для обчислень та збереження	Розподілені вузли, що використовуються для обчислень та збереження (призначені служби)		

**Таблиця 1 – Порівняння HDFS та інших розподілених файлових систем**

У таблиці 1 наводиться порівняння системи HDFS та інших розподілених файлових систем з відкритим вихідним кодом, включаючи Ceph, Lustre, і PVFS. Основні відмінності включають обробку файлових метаданих в централізованому або розподіленому режимі, реалізація відмовостійкості по причині апаратного RAID (нижче рівня файлової системи) або на базі реплікації програмного забезпечення і накладень без POSIX обмежень, таких як файл-записи `-once` (тільки один раз). У широкому сенсі розуміння HDFS була розроблена спеціально для підтримки одного виду програмної моделі – MapReduce і, таким чином, має спрощений дизайн, сильно оптимізований в напрямку до потокового доступу. В протилежність їй, інші паралельні розподілені файлові системи були розроблені для високопродуктивних обчислювальних середовищ, в основу яких положені кілька моделей програмування, таких як MPI, PVM, OpenMP і, таким чином, вони повинні підтримувати більш широкий спектр можливостей.

#### **4. Агностик API файлової системи Hadoop (HFAA API)**

Агностик API для файлової системи Hadoop (HFAA) забезпечує простий спосіб взаємодії Hadoop з файловими системами, відмінними HDFS. Вона орієнтована на розробників цих альтернативних файлових систем, тобто, тих, хто має уявлення про свою розподілену файлову систему, але не є обізнаними у внутрішній роботі системи Hadoop. HFAA усуває необхідність в розширеннях користувача для реалізації Hadoop в Java і забезпечує зв'язок з будь-якою файловою системою через TCP сокети. Архітектура HFAA в загальному добре описана і для обговорення проектування розподілені і паралельні файлові системи, відмінні від співвідних з HDFS (наприклад, PVFS, Ceph, або Lustre) будемо називати комбінованим іменем – нові розподілені файлові системи (NewDFS).

Цей API призначений для розгортання в типовому середовищі Hadoop, де кожен вузол кластера відповідає за обчислення (через Hadoop MapReduce) і зберігання (через HDFS). Коли HDFS замінюється всім необхідним "NewDFS", кожен вузол буде як і раніше відповідати за обчислення і збереження. HDFS daemon буде відключена, а замість неї буде налаштована NewDFS daemon.

Щоб включити Hadoop MapReduce для зв'язку з NewDFS daemon, архітектура HFAA додає дві програмні компоненти для системи: клієнт HFAA і сервер HFAA, – як показано на рисунку 2. Клієнт HFAA інтегрується зі стандартною Hadoop MapReduce framework і перехоплює запити читання/запису, які зазвичай надходять в HDFS. Потім клієнт направляє запити через мережеві сокети на сервер HFAA, які, як правило, будуть працювати на тому ж вузлі, але може бути віддалений. Сервер HFAA відповідає за взаємодію з NewDFS, отримуючи мережеві запити від клієнта HFAA для виконання операцій файлової системи (створення, відкриття, видалення тощо) і відповідає за результати даних/метаданих з NewDFS.

Клієнт HFAA створений в Java інтегрований з Hadoop framework і може бути повторно використаний з будь-яким NewDFS. На відміну від цього, сервер HFAA написаний на будь-якій

іншій мові, на якій реалізований NewDFS, відповідно, і повинен бути переписаний для кожної нової з'єднаної файлової системи. Ми можемо побачити, що цей компонент наданий розробниками NewDFS, які не мають ніякої потреби в детальному знанні внутрішньої роботи Hadoop. Детальна інформація проектування та експлуатації клієнта і сервера HFAA описані нижче.

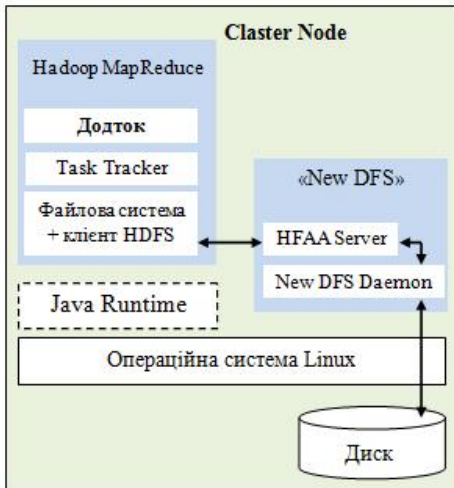


Рис. 2 – Вузол кластера HFAA

#### Інтеграція клієнта з Hadoop

Реалізація Hadoop в Java включає абстрактний клас під назвою `org.apache.hadoop.fs.FileSystem`. Кілька файлових систем розширюють і реалізують цей абстрактний клас, у тому числі HDFS (файлову систему за замовчуванням), Amazon S3, і Kosmos File System. Клієнт HFAA інтегрується з Hadoop шляхом реалізації цього класу файлової системи, по суті, змінюючи інтерфейс HDFS з еквівалентною функціональністю на інтерфейс для взаємодії з будь-якими NewDFS через TCP сокети.

Функція	Опис
Відкрити (Open)	Відкрити файл для читання.
Створити (Create)	Створити файл для запису.
Додати (Append)	Додати дані в існуючий файл.
Переіменувати (Rename)	Змінити ім'я файла або/чи перемістити файл в іншу директорію.
Видалити (Delete)	Видалити файл/директорію разо з їх інформацією.
Статус виводу списку (ListStatus)	Вивід списку файлів поточної директорії.
Створити директорію (Mkdir)	Створити директорію за вказаним шляхом в файловій системі. Наддиректорії мусять уже бути створені і існувати на цей

	момент в файловій системі.
Статус виводу атрибутів (GetFileStatus)	Виявляє метадані (атрибути: об'єм, локалізація блоку, спосіб відповіді, власник, ...) для вказаного файлу чи директорії.
Запис (Write)	Операція запису реалізована за допомогою нового класу OutputStream.
Читання (Read)	Операція читання реалізована за допомогою нового класу OutputStream.

**Таблиця 2 – Операції HFAA**

Ключова функціональність HFAA реалізує включення, подані в таблиці 2. Для згенерації протоколу простий запит команди і відповідні типи кодів та їх дані/метадані обмінюються між клієнтом HFAA і сервером HFAA у вигляді рядкового формату. Групи запитів і відповідей є гнучкими і простими до організації роботи, тому що більшість мов програмування пропонують бібліотеки для роботи з рядками. Повний технічний опис протоколу HFAA знаходиться в [22].

#### **Інтеграція сервера з NewDFS**

Сервер HFAA функціонально взаємодіє з NewDFS. Він може приймати мережеві запити від клієнта HFAA для виконання операцій файловою системою (наприклад, створювати, відкривати, видаляти і т.д.) і відкликається на відповідні дані/метадані, отримані від NewDFS. У цій архітектурі функції інтерфейсу між клієнтом HFAA і сервером HFAA стандартизовані, але функції інтерфейсу між HFAA сервером і NewDFS daemon залишаються для реалізатора. Таким чином, сервер HFAA може бути інтегрований в самій NewDFS daemon, або залишений як автономна програма.

Незалежно від того, чи є він автономний чи інтегрований з демона NewDFS, базова структура сервера HFAA полягає в наступному. Сервер HFAA прослуховує тісно переплетені вхідні запити TCP, проглядаючи кожен запит як незалежну нитку або завершальний процес. Сервер HFAA повинен бути в змозі реагувати на численні запити одночасно, оскільки типовий Hadoop вузол паралельно обробляє декілька завдань MapReduce, і тому що кожне завдання має принаймні один файл даних і кілька файлів метаданих (відстеження статусу виконання завдання), що відкриваються в будь-який момент часу. У кожному процесі є багато обробників запитів для кожної функції API Hadoop (наприклад, getFileStatus(), listStatus()). Обробники запитів обслуговують запити повністю до завершення, а потім виходять з них.

#### **6. Роботи, пов'язані з даною тематикою**

Інтеграції функціональності MapReduce з Hadoop з файловими системами, відмінними від HDFS в останні роки був проявлений значний інтерес. Така інтеграція типічно наслідуює один з двох можливих підходів: розширення вбудованого абстрактного класу файлової системи для Hadoop зі вставкою коду для підтримки бажаної розподіленої файлової системи, або з використанням POSIX-драйвера щоб експонувати розподілену файлову систему в базову операційну систему, а потім безпосередньо використовувати цей ресурс з середини Hadoop.

Приклади альтернативних розподілених файлових систем, які були успішно використані з Hadoop MapReduce включають: Amazon S3 [9], Ceph [11], CloudStore/KosmosFS [10], GPFS [12], PVFS [14], і Lustre [13]. Багато з цих файлових систем можуть бути використані в інтеграції одного з наведених підходів: підходу інтеграції POSIX-драйвера або підходу вставки коду для підвищення продуктивності. Ключова різниця між усіма цими підходами і підходу, обраного для HFAA, полягає в тому, що HFAA був розроблений для підтримки постійності в реалізації Hadoop,

і було висунуто спрощений набір реалізованих відповідальних моментів в напрямку розподіленої файлової системи. На відміну від цього, всі попередні інтегровані проекти включали в себе свою власну відмінність, унікальний набір модифікацій і виправлень в базі коду Hadoop. Для отримання детальної додаткової інформації з проектування HFAA та протоколу необхідно звертатися до роботи [22].

### **Обговорення результатів**

Архітектура HFAA клієнта і сервера, описані вище, були реалізовані і протестовані з Hadoop 0.20.204.0 і OrangeFS 2.8.4, – розподіленою файловою системою типу PVFS. Клієнт HFAA був розроблений в Java і скомпільований в Hadoop framework. HFAA сервер був написаний в C, для того щоб надати перевагу існуючому коду PVFS. Для зручності реалізації сервер HFAA не був безпосередньо інтегрований в daemon PVFS збереження. Швидше, він був реалізований як автономний процес, схожий за стилем на існуючу програму AdminTools, що поставляється з новою файловою системою. Ці утиліти пропонують команди в стилі POSIX (наприклад, ls, touch) для підтримки та адміністрування PVFS.

Для багатогранної оцінки використовувалися чотири однорідні сервери. Кожен сервер складався з процесора Intel Xeon X3430, оперативної пам'яті 4 ГБ, і жорсткий диск SATA 500 Гб. Для зберігання даних HDFS було виділено розділ на зовнішньому краю диска в 100 ГБ (тобто, найшвидша область) або дані PVFS. Використання лише невеликої області диска виявляє різницю пропускну здатності при доступі між найшвидшим і самим повільним секторами враховуючи фізичне розташування диска і зводиться до мінімуму – нижче 8% в експериментах. Кожен сервер був налаштований з Ubuntu 10.10 Linux-дистрибутива і програмного стека (Hadoop + розподілена файлова система), яка змінюється в експерименті. Всі вузли були пов'язані з допомогою свіча Gigabit Ethernet в закритій мережі.

В таблиці 3 подано три різні конфігурації програмного забезпечення кластера, що використовувалися для експериментів: Hadoop з HDFS, Hadoop з PVFS і Hadoop з PVFS що використовує HFAA. Для збереження однаковості експериментів, у всіх конфігураціях використовувався один "старший" сервер для планування завдань, і три "допоміжні" сервери – для обчислення і збереження файлів даних. Розташування файлових метаданих змінювалося між експериментами.

Перша конфігурація, позначена Hadoop + HDFS – це стандартна Hadoop архітектура, де головний вузол стартує централізований JobTracker та послуги іменованого вузла (NameNode – для управління завданнями і простором імен файлової системи) і кожен підлеглий вузол обумовлює працювати власний примірник TaskTracker і послуги вузла даних (Data-Node – для обчислень і розподіленого збереження).

У другій конфігурації, що названа Hadoop + PVFS, HDFS була видалена і замінена PVFS. Драйвер POSIX-клієнта, що забезпечувався PVFS, використовується для монтування розподіленої файлової системи на кожному вузлі, і Hadoop налаштований на використання "локальної" директорії файлової системи безпосередньо через інтерфейс RawLocalFileSystem. У цій конфігурації Hadoop обмежується інтерфейсом POSIX-драйвера, і не може отримати інформаційної локальності з наголошеної файлової системи.

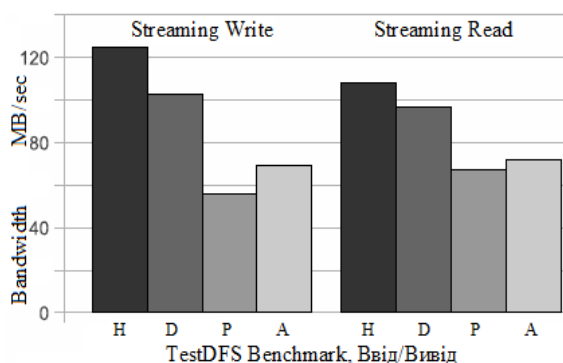
Архітектура PVFS розроблена з 3-х компонентів: сервери метаданих (зберігають метадані), сервери вводу-виводу (для зберігання даних), і клієнти. Для узгодженості з конфігурацією HDFS, головний вузол в конфігурації Hadoop + PVFS використовується тільки для запуску сервера метаданих, а всі інші підпорядковані вузли працюють як сервер метаданих і сервер вводу/виводу. Таким чином, як і в конфігурації HDFS, тільки 3 підпорядковані вузли зберігають дані файлу. Однак, на відміну від HDFS, сервер(и) метаданих (еквівалент іменованих вузлів) розподіляється по всьому суцільному кластеру.

В третій конфігурації Hadoop + HFAA + PVFS, змінюється інтерфейс між Hadoop і PVFS. POSIX-драйвер і PVFS-клієнт видаляються, і замінюються новою клієнт-серверною архітектурою HFAA як це буде обговорюватися нижче. Як і в попередній конфігурації, кожен вузол виконує сервер метаданих, але тільки підпорядковані вузли запускають сервер вводу/виводу з метою відтворення оригінальної HDFS-архітектури.



Вузол	Hadoop+HDFS	Hadoop+PVFS	Hadoop+HFAA+PVFS
Головний (1)	H: JobTracker D: ІменованійВузол (метадані)	H: JobTracker P: POSIX-драйвер P: Клієнт P: СерверМетаданих (метадані)	H: JobTracker A: HFAA-клієнт A: HFAA-сервер P: СерверМетаданих (метадані)
Підлеглий (3)	H: TaskTracker D: ВузолДаних(дані)	H: TaskTracker P: POSIX-драйвер P: Клієнт P: СерверМетаданих (метадані) P: СерверВ-В(дані)	H: TaskTracker A: HFAA-клієнт A: HFAA-сервер P: СерверМетаданих (метадані) P: СерверВ-В(дані)

**Таблиця 3: Позначення конфігурації кластера для програмного забезпечення: Н = Hadoop MapReduce, D = HDFS, P = PVFS, A = HFAA**



**Рис. 3 – Пропускна здатність на запис і читання при доступі до файлу для варіантів, поданих в таблиці 3**

Кожна із трьох кластерних конфігурацій багатогранно протестовані з додатком TestDFSIO для MapReduce, який поставляється з Hadoop framework. Цей тест виконує паралельні операції потокового читання або потокового запису для багатьох файлів в рамках кластера. Кожен окремий файл зчитується або записується одним завданням відображення програми TestDFSIO, і кожен підлеглий кластер може виконувати кілька завдань відображення одночасно. Так як кожне завдання відображення виконується в межах кластера, одиничні завдання зменшують агрегативність завдань і статистику звітів продуктивності виконання. У цих експериментах TaskTracker на кожному підлеглому вузлі виконує єдине завдання відображення за допомогою TestDFSIO: читання або запис єдиного файлу на 5 Гб в глобальній файлової системі, і в цілому 15 Гб доступних даних в сукупності по всьому кластеру. Слід звернути увагу, що через те що реплікація природньо не підтримувалася PVFS, вона була відключена у всіх конфігураціях.

На малюнку 3 подані результати TestDFSIO для трьох кластерних конфігурацій для робочих операцій читання і запису. Ці результати порівнюються з результатом обробки сировинного диска повністю поза середовищем Hadoop, який був виміряний за допомогою утиліти dd, зконфігурованої для роботи з потоковим доступом. При стандартній конфігурації Hadoop + HDFS досягається 82% від середньої пропускної здатності для запису і 87,3% від середньої пропускної здатності для читання порівняно з пропускною здатністю для необробленого диска. На відміну від цього, при конфігурації Hadoop + PVFS досягається тільки 46% від середньої пропускної здатності для запису і 60,3% від середньої пропускної здатності для читання в порівнянні з пропускною здатністю для необробленого диска. Нарешті, конфігурація Hadoop + HFAA + PVFS дозволяє досягти 54,9% від середньої пропускної здатності для запису і 64% від середньої пропускної здатності для читання в порівнянні з вихідною пропускною здатністю для необробленого диска, показуючи при цьому, як HFAA може зберегти оригінальну пропускну здатність файлової системи і зробити її доступною для Hadoop.

Одержані результати виявляють функціональні можливості інтерфейсу HFAA. Однак, використання цього API з PVFS не підходить, а також не перевершує рівні вводу/виводу HDFS, підтримується 67,2 % з пропускної здатності для запису HDFS і 74,1% з пропускної здатності для читання HDFS. Інтеграція PVFS з HFAA досягає ~23% приросту пропускної здатності для запису і



~7% приросту пропускної здатності для читання в порівнянні з використанням PVFS як локальної файлової системи через інтерфейс POSIX-ядра.

### Висновки

Агностик API файлової системи Hadoop розширює Hadoop з простим протоколом зв'язку, що дозволяє інтегруватися їй з будь-якою файловою системою, яка підтримує сокети TCP. Цей новий API усуває необхідність у налаштуванні реалізації Hadoop в Java, і замість цього переміщає функції відповідальності реалізації на саму файлову систему. Таким чином, розробники, бажаючи інтегрувати їх нову файлову систему з Hadoop не несуть відповідальності за розуміння деталей внутрішньої роботи в Hadoop.

Надаючи файловій системі агностик API майбутня робота може безпосередньо досліджувати компроміси продуктивності з використанням програмної моделі Map-Reduce з файловими системами, що підтримують операції записів в будь-якому місці, розподілених серверів метаданих та інших переваг. Крім того, HFAA дозволяє Hadoop інтегруватися більш легше в існуючих обчислювальних високопродуктивних середовищах, де заміщення здійснюється вже відомими розподіленими файловими системами.

1. Hadoop. <http://hadoop.apache.org>, 2012.
2. J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. In OSDI'04: Proceedings of the 6-th Symposium on Operating Systems Design & Implementation, page 10, Berkeley, CA, USA, 2004.
3. Powered by Hadoop. <http://wiki.apache.org/hadoop/PoweredBy>, 2012.
4. S. Radia. HDFS federation - apache Hadoop india summit. <http://www.slideshare.net/huguk/hdfs-federation-hadoop-summit>, 2011.
5. HDFS (Hadoop distributed file system) architecture. [http://hadoop.apache.org/common/docs/current/hdfs\\_design.html](http://hadoop.apache.org/common/docs/current/hdfs_design.html), 2012.
6. K. Shvachko, H. Kuang, S. Radia, and R. Chansler. The Hadoop distributed file system. In Proceedings of the 2010 IEEE 26-th Symposium on Mass Storage Systems and Technologies (MSST), MSST '10, pages 1 – 10, Washington, DC, USA, 2010.
7. K. V. Shvachko. HDFS scalability: The limits to growth. In login: The Magazine of USENIX, volume 35, pages 6 – 16, Apr. 2010.
8. K. V. Shvachko. Apache Hadoop: The scalability update. In login: The Magazine of USENIX, volume 36, pages 7 – 13, June 2011.
9. Amazon S3. <http://wiki.apache.org/hadoop/AmazonS3>, 2012.
10. kosmosfs - kosmos distributed filesystem. <http://code.google.com/p/kosmosfs/>, 2012.
11. C. Maltzahn, E. Molina-Estolano, A. Khurana, N. A. J., S. A. Brandt, and S. A. Weil. Ceph as a scalable alternative to the Hadoop distributed file system. In login: The Magazine of USENIX, volume 35, pages 38 – 49, Aug. 2010.
12. R. Ananthanarayanan, K. Gupta, P. Pandey, H. Pucha, P. Sarkar, M. Shah, and R. Tewari. Cloud analytics: do we really need to reinvent the storage stack? In Proceedings of the 2009 conference on Hot topics in cloud computing, HotCloud'09, Berkeley, CA, USA, 2009.
13. Using Lustre with Apache Hadoop. [http://wiki.lustre.org/images/1/1b/Hadoop\\_wp\\_v0.4.2.pdf](http://wiki.lustre.org/images/1/1b/Hadoop_wp_v0.4.2.pdf), Jan. 2010.
14. W. Tantisiriroj, S. W. Son, S. Patil, S. J. Lang, G. Gibson, and R. B. Ross. On the duality of data-intensive file system design: reconciling HDFS and PVFS. In Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, SC '11, pages 67:1 – 67:12, New York, NY, USA, 2011.
15. S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn. Ceph: a scalable, high-performance distributed file system. In Proceedings of the 7th symposium on Operating systems design and implementation, OSDI '06, pages 307 – 320, Berkeley, CA, USA, 2006.
16. Lustre file system. <http://www.lustre.org>, 2012.
17. Parallel virtual file system, version 2. <http://www.pvfs.org>, 2012.
18. F. Schmuck and R. Haskin. Gpfs: A shared-disk file system for large computing clusters. In Proceedings of the 1-st USENIX Conference on File and Storage Technologies, FAST '02, Berkeley, CA, USA, 2002.
19. S. Ghemawat, H. Gobiuff, and S.-T. Leung. The Google file system. In SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles, pages 29 – 43, New York, NY, USA, 2003.
20. M. K. McKusick and S. Quinlan. GFS: Evolution on fast-forward. Queue, 7:10:10 – 10:20, August 2009.
21. B. Welch, M. Unangst, Z. Abbasi, G. Gibson, B. Mueller, J. Small, J. Zelenka, and B. Zhou. Scalable performance of the panasas parallel file system. In Proceedings of the 6th USENIX Conference on File and Storage Technologies, FAST'08, pages 2:1 – 2:17, Berkeley, CA, USA, 2008.
22. A. J. Yee. Sharing the love: A generic socket API for Hadoop mapreduce. Master's thesis, University of the Pacific, 2011.

УДК 004.254(045)

Мельник В.М.к.ф.-м.н. доц., Багнюк Н.В. к.т.н. доц., Мельник К.В. к.т.н. доц.  
Луцький національний технічний університет

## АЛГОРИТМИ ПЛАНУВАННЯ ВІДОМИМ ОБМІНОМ ДАНИХ В ОПЕРАЦІЙНИХ СИСТЕМАХ ДЛЯ МУЛЬТИСОКЕТНИХ БАГАТОЯДЕРНИХ СЕРВЕРІВ

**Мельник В.М., Багнюк Н.В., Мельник К.В. Алгоритми планування відомим обміном даних в операційних системах для мультисокетних багатоядерних серверів.** Для основних виробників чіпів відомі всі новітні багатоядерні мікропроцесори. Мультисокетні системи, які побудовані на базі таких процесорів зазвичай використовуються для виконання різних серверних додатків. Типічно кожен процесор в такій системі розділяє кеш-пам'ять або на рівні L2 або на рівні L3. В залежності від програми, яка виконується в системі, передача кеш-пам'ять-кеш-пам'ять (КП-КП) між сокетами може впливати на загальну продуктивність системи. Дана робота представляє нову оптимізацію планування операційної системи (ОС) з метою зменшення впливу подібної міжсокетної передачі між кеш-пам'яттю і кеш-пам'яттю.

Спостерігаючи передачу між кеш-пам'яттю і кеш-пам'яттю поміж кожною парою ниток для кожного кванта планування та застосування чотирьох різних алгоритмів було придумано новий графік потоків для наступного запланованого кванта. Цей новий графік потенційно скорочує міжсокету передачу між кеш-пам'яттю і кеш-пам'яттю для наступного кванта планування. Досліджувався вплив цих алгоритмів на 18 реальних тестах. За допомогою тестів вивчалася міжсокетна передача між кеш-пам'яттю і кеш-пам'яттю, де спостерігалось зменшення більш як на 99,3% на деяких тестах і, в середньому, від 5,5% до 24% в залежності від використовуваного алгоритму планування.

**Ключові слова:** алгоритми планування, кеш-пам'ять, передача КП-КП, продуктивність, управління процесами планування, багатопроцесорний сервер, нитки.

**Мельник В.М., Багнюк Н.В., Мельник Е.В. Алгоритмы планирования известным обменом данных в операционных системах для мультисокетных многоядерных серверов.** Для основных производителей чипов известны все новейшие многоядерные процессоры. Мультисокетные системы, построенные на базе таких процессоров обычно используются для выполнения различных серверных приложений. Типично каждый процессор в такой системе разделяет кэш-память или на уровне L2 или на уровне L3. В зависимости от программы, которая выполняется в системе, передача кэш-память-кэш-память (КП-КП) между сокетами может влиять на общую производительность системы. Данная работа представляет новую оптимизацию планирования операционной системы (ОС) с целью уменьшения воздействия подобной межсокетной передачи между кэш-памятью и кэш-памятью.

Наблюдая передачу между кэш-памятью и кэш-памятью между каждой парой нитей для каждого кванта планирования и применения четырех различных алгоритмов было придумано новый график потоков для следующего запланированного кванта. Этот новый график потенциально сокращает передачу между кэш-памятью и кэш-памятью между сокетами для последующего кванта планирования. Исследовалось влияние этих алгоритмов на 18 реальных тестах. С помощью тестов изучалась междусокетная передача между кэш-памятью и кэш-памятью, где наблюдалось уменьшение более чем на 99,3% в некоторых тестах и, в среднем, от 5,5% до 24% в зависимости от используемого алгоритма планирования.

**Ключевые слова:** алгоритмы планирования, кэш-память, передача КП-КП, производительность, управление процессами планирования, многопроцессорный сервер, нити.

**Melnyk V.M., Bagnyk N.V., Melnyk K.V. Data knowledge exchange algorithms using in the multi-socket operating systems for multicore servers.** Major chip manufacturers have all introduced multi-core microprocessors. Multi-socket systems built from these processors are routinely used for running various server applications. Typically, each processor in such a system shares a cache at either the L2 or L3 level. Depending on the application that runs on the system, intersocket cache-to-cache transfers can affect overall performance. This paper presents a new operating system (OS) scheduling optimization to reduce the impact of such inter-socket cache-to-cache transfers.

By observing the pattern of cache-to-cache transfers between every pair of threads for each scheduling quantum and applying four different algorithms, we come up with a new schedule of threads for the next quantum. This new schedule potentially cuts down the inter-socket cache-to-cache transfers for the next scheduling quantum. We studied the impact of these algorithms on 18 real-world benchmarks. For the benchmarks we studied, inter-socket cache-to-cache transfers are down by as much as 99.3% on some benchmarks and, on average, between 5.5% and 24% depending on the scheduling algorithm employed.

**Keywords:** Scheduling Algorithms, cache memory, cache memory, cache-to-cache transfers, performance, process scheduling management, multiprocessor server, threads.

**Постановка наукової проблеми.** Багато комерційних серверних додатків сьогодні працюють на мультисокетних багатоядерних серверах. Ці програми, як правило, страждають від різних видів недостатків кешу. Більшість з цих недоліків можна уникнути, збільшивши або розмір кешу, кеш асоціативність або розмір рядка. Однак є й інші види недоліків зокрема так звані недоліки в комунікації, практично недоліки міжсокетної комунікації, які притаманні додаткам і не можуть бути легко вирішені шляхом обмежування параметрів кешу. Це залежить від кількості

спільного використання програми та шляху, яким додаток був написаний. Щоб покращити цю проблему, один із способів – це переписати програму. Інший спосіб полягає в спостереженні за предметами спільного використання та адаптації планувальника завдань операційної системи (ОС), щоб мінімізувати вплив недоліків зв'язку між сокетами. У цій роботі представлено таку оптимізацію планування ОС, яка відстежує сокетні трансфери КП-КП для кожного кванта планування, а потім застосовує чотири різні алгоритми вирішення, де слід запланувати кожен потік для наступного кванта планування. Встановлено, що для різних тестів, які вивчалися, різні алгоритми і їх зміни зменшували міжсокетну КП-КП передачу, в середньому, від 5,5 до 24% і до 99,3% для деяких тестів. Наскільки відомо, жодна з поточних комерційних ОС не оптимізовує власний планувальник завдань з урахуванням недоліків зв'язку, то далі розмова піде про алгоритми планування, їх складність і ймовірність, з якою вони вносять вклад в додатки, а також методика використання для оцінки цих алгоритмів та результати роботи.

**Аналіз досліджень.** Алгоритми планування. Різні алгоритми планування можуть допомогти зменшити міжсокетну КП-КП передачу. Розглянемо один простий і три складні комплексні алгоритми в цій роботі. Для всіх алгоритмів ми виділяємо той, що відповідає лічильнику підтримки апаратної продуктивності, для розрахунку міжсокетної КП-КП передачі. Для кожного ядра ми призначаємо 480-бітний регістр, розділений на п'ятнадцять 32-бітних пакетів між ниткою, що виконується на даному ядрі і ниток, які протікають на всіх інших п'ятнадцяти ядрах (наша максимальна мікропроцесорна (МП) конфігурація – це система з чотирма сокетами, де кожен з сокетів об'єднує чотири ядра). Ми можемо зменшити ширину регістра шляхом спрощення КП-КП передач, замість того, щоб рахувати кожен передачу [1]. У кожному запланованому кванті для кожного запиту зчитування, що впливає на КП-КП передачу, ведеться спостереження, яка з ниток пересилає дані, і збільшується лічильник, що відповідає цій нитці. Наприкінці запланованого кванта ОС зчитує лічильники для кожної нитки, а потім застосовує один з чотирьох алгоритмів для прийняття рішення про планування для наступного кванта планування. Ця інформація лічильника може бути збережена в ОС як частина контексту нитки. Алгоритми описуються в наступних підрозділах.

#### *Послідовність вирішення проблеми*

Проблема, яку потрібно вирішити, можна сформулювати наступним чином. Розглянемо 16 ниток, що виконуються на системі з чотирма сокетами, де кожен з сокетів об'єднує чотири ядра. Представимо орієнтований граф з 16 вершинами, де кожна з вершин представляє ядро. Ребро з вершини  $i$  в вершину  $j$  представляє КП-КП передачу як результуючу передавання даних від процесора  $i$  до процесора  $j$  (див. рис. 1). Крайове навантаження  $W_{ij}$  являє собою кількість КП-КП передач. Проблема полягає в тому, щоб згрупувати разом ці вершини з максимальною сумою країв так, щоб глобальний максимум суми ребер був максимальний. Ми представляємо простий алгоритм, а потім більш складний покращений алгоритм, з подальшим удосконаленням покращеного алгоритму, а потім з невеликим штрихом поліпшення, так що він може скоро вийти. Зверніть увагу, що найбільш складний алгоритм намагається згрупувати нитки тільки на основі даного кванта планування. Це не може виявитися оптимальним для наступного етапу. Це надихає на думку про вивчення алгоритмів застосування для удосконалення найбільш складного алгоритму.

#### **Вклад основного матеріалу й обґрунтування отриманих результатів дослідження.**

**Алгоритм 1.** У цьому алгоритмі вибирається одна нитка для кожного сокета, що дасть найбільший вклад шляхом переходу до наступного сокета. Наприклад, якщо взяти систему з чотирма сокетами, то вирішуємо заздалегідь, що переміщаємо першу нитку від сокета 0 до сокета 1, іншу нитку – з сокета 1 в сокет 2, наступну нитку – з сокета 2 в сокет 3, а ще наступну – з сокета 3 до сокета 0. Отже рух між сокетами вирішено заздалегідь. Для кожного сокета розраховуємо, яка з чотирьох ниток дасть найбільший вклад шляхом переходу до наступного сокета. Після цього вибираємо і переміщаємо цю конкретну нитку в наступний квант планування. В кінці кожного кванта планування (крім першого) обчислюється вклад, отриманий від нової "карти" розподілу ниток в порівнянні з вихідною і представляється цей вклад в результаті запропонованого алгоритму.

#### *Складність*

$$O(p*N*K) + O(p*N) = O(p*N*K)$$

$p$  = номер кванта планування;  $N$  = Кількість ниток;  $K$  = кількість ядер на сокет (4 в нашому випадку)

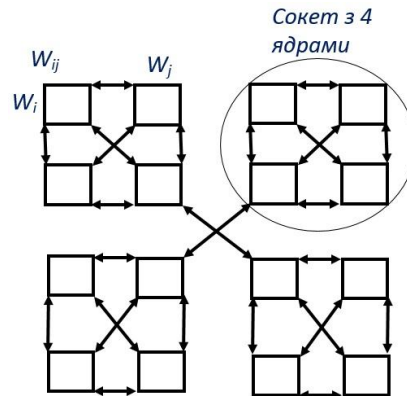


Рис. 1. Кластерний граф (міжкластерний рух не повністю показаний на графі)

Попередній номер складності, в основному, для того щоб розрахувати внесок для кожної з  $N$  ниток шляхом об'єднання його КП-КП передачі даних зі всіма  $K$  нитками свого власного сокета, і тими, що на наступному сокеті. Потім береться різниця між ними, щоб обчислити внесок, який можна отримати, переміщуючи цю нитку до наступного сокета. Це має бути зроблено для кожних фази/кванта планування. Все це стає  $O(p*N*K)$  (примітка: вся складність представляється для всіх квантів планування; альтернативно, можна видалити член умови  $p$  і спостерігати складність на кожен квант). Після розрахунку внеску потрібно порівняти внески від всіх  $K$  ниток у сокеті для того, щоб вирішити, яка нитка має максимальний внесок і спрямувати цю нитку до наступного сокета. Це має бути зроблено для кожного сокета і для кожного кванта планування. Це стає  $O(p*N)$ . Таким чином, в цілому, складність є  $O(p*N*K)$ . Цей алгоритм може бути легко розпаралелений. Якщо кожне ядро виконує частину алгоритму, то складність для кожного кванта планування/фази стає  $O(K^2)$ , де  $K$  є постійним (по суті,  $O(1)$ ).

*Варіант 1 алгоритму 1*

Є два варіанти для першого алгоритму. Обидва варіанти пов'язані зі спостереженням фазової поведінки КП-КП передачі і дають відповідні переваги в його реалізації. У першому варіанті кластеризується певна кількість  $L$  квантів планування в один квант, а потім приймається рішення планування для наступної  $L$  кількості квантів планування. Для кожної нитки обчислюється вклад руху цієї нитки до наступного кластера для кожного з  $L$  квантів планування, а потім об'єднуються вклади для всіх  $L$  квантів. Виходячи з цього вкладу вибирається одна з чотирьох ниток в кластері, яка приносить найбільший вклад при переході на наступний кластер. Цей процес повторюється для кожного кластера (двох кластерів для двосокетної системи і чотирьох кластерів для системи з чотирма сокетами). Після цього отримується нове "відображення" ниток і використовується для наступних  $L$  квантів планування.

*Складність*

$$O(p*N*K) + O((P/L)*N) = O(p*N*K);$$

$p$  = кількість квантів планування;  $N$  = кількість ниток;  $K$  = кількість ядер на сокет (в нашому випадку 4);  $L$  = кількість планування квантів за якими ведеться спостереження одночасно для того, щоб прийняти рішення на наступні  $L$  квантів. Аргумент складності такий же, як вихідний, за винятком того, що число фаз спадає з коефіцієнтом  $L$  (на другий відрізок часу). Таким чином, загальна складність становить  $O(p*N*K)$ .

*Варіант 2 алгоритму 1*

У другому варіанті також ведеться спостереження за поведінкою в межах фази, але використовується "алгоритм вивчення", з метою дізнатися про які-небудь загальні закономірності, що відбуваються в  $L$  квантів планування, як описано вище. Отримується відображення ниток для кожного  $L$  кванта планування та формується для них бітова картина. Є стільки бітових карт, скільки є сокетів в системі. Якщо нитка  $i$  запланована на сокеті  $0$ , то біт встановлений у положенні  $i$  на бітовій карті відповідає сокету  $0$ . Визначається, чи є загальний шаблон (загальний шаблон

зображений з двох логічно помножених  $\square$ AND $\square$  послідовних бітових карт і число бітів, встановлених в результуючому шаблоні повинно бути більше ніж 2), що проходить від одного кванта до іншого в бітових картах. Якщо загальний шаблон виконується, то вклад буде  $i*(1/L)$  для цього шаблону, в якому  $i$  – це число кванта планування в  $L$ -розгалуженні квантів планування. Це дає більший вклад для "останніх" шаблонів, що утворюються, і менший вклад для "старих" шаблонів. Також надається вклад ( $i*(1/(L*L))$ ) для кожної з вихідних (операція заперечення  $\square$ AND $\square$ ) бітових карт, які отримуються для кожного з  $L$  квантів планування. Потім шаблони впорядковуються в залежності від вкладу, що присвоюється даному шаблону. Після сортування переглядається відсортований список і можна спробувати сформувану карту, використовуючи найбільш ті шаблони, що частіше зустрічаються. Отримана карта використовується для наступного  $L$  планування квантів.

**Складність**

$$O(p*N*K)+O(p*N)+O((p/L)*((S^2*L+S*L)^2))+O((p/L)*(S^2*L+S*L))=O(p*N*K)+O((p/L)*((S^2*L+S*L)^2));$$

$p$  = кількість квантів планування;  $N$  = кількість потоків;  $K$  = кількість ядер на сокет (4 в нашому випадку);  $L$  = кількість квантів планування, що спостерігається одночасно для прийняття рішення на наступні  $L$  квантів;  $S$  = кількість сокетів.

Перші дві умови (терми) є такими ж, як і в варіанті 1. Третя і четверта умови застосовуються для  $\square$ алгоритму вивчання $\square$ . Для алгоритму вивчання виконується карта з урахуванням ідентифікаторів ниток, запланованих на кожний сокет (в бітовій карті встановлюється біт в положенні ID для нитки). Отримуються  $S$ -бітні карти, утворені для кожного кванта планування, де  $S$  – це кількість сокетів. Кожна з  $S$ -бітних карт логічно перемножується з  $S$ -бітними картами наступного кванта планування, щоб отримати відповідний шаблон, що приводить до  $O(S^2)$ . Крім цього, також включаються  $S$ -бітові карти для кожного кванта планування із вкладом ( $i*(1/(L*L))$ ), якбуло описано раніше, і це, в свою чергу, приводить до  $O(S)$ . Це повторюється для всіх  $L$  квантів планування за якими ведеться спостереження. Всі відповідні шаблони вставляються в відсортованому порядку і систематизується тип сортування для вставки. Таким чином, складність стає  $O((S^2*L+S*L)^2)$ . (Примітка: для простоти призначається тип вставки. Насправді, можна було б це робити за допомогою двійкового дерева пошуку, щоб відстежувати треки входу або які-небудь інші ефективні структури, і в цьому випадку алгоритм стає набагато ефективнішим.). Це повинно повторюватися для всіх  $p/L$  фаз і, таким чином, загальна складність стає  $O((p/L)*((S^2*L+S*L)^2))$ . Після такого введення в мульти-карту слід зробити лінійний хід від шаблонів узгодження до отримання нового відображення ниток. Це має бути зроблено для всіх  $(p/L)$  фаз, а результат очікується в четвертому діапазоні.

**Алгоритм 2**

Цей алгоритм є дещо складнішим в порівнянні з першим. В кінці кожного кванта збираються лічильники від усіх ядер, а потім вибираються всі можливі комбінації груп з чотирьох ниток ( $N@4=N$ ) і підсумовується КП-КП передача між парами (там буде 12 можливих пар (напрямлених ребер) КП-КП передачі між чотирма нитками). Потім відсортовуються суми в порядку спадання. Після сортування, вибирається перша група з чотирьох ниток з найвищою сумою і сплановується на сокет 0, а потім вибирається наступна група з чотирьох ниток з високою сумою і сплановується на сокет 1, і так далі, поки не закінчиться планування ниток на всіх сокетах. Тим не менш, це не є оптимальною евристикой, тому що використовувався спрощений алгоритм для вибору групи з чотирьох ниток з відсортованого списку.

**Складність**

$$O(p*N@4) + O(p*((N@4)^2)) + O(p*(N@4)) = O(p*((N@4)^2));$$

$p$  = кількість квантів планування;  $N$  = кількість ниток;  $N@4=N$ -комбінованість 4.

Ці три умови є наступними. Перша умова відповідає за формування  $N@4$  парних взаємодій для всіх  $p$  квантів планування. Друга умова вставляє попарно суми в відсортованому порядку. Встановлюється сортування для вставки, яке повинно бути зроблено для кожного кванта планування. Отже,  $O(p*((N@4)^2))$ . Третя умова здійснює лінійну апроксимацію, щоб підібрати нове відображення на наступний квант планування в спрощеному варіанті. Це знову-таки має бути зроблено для всіх  $p$  квантів планування. Отже,  $O(p*(N@4))$ .

**Варіант 1 алгоритму 2**

Існують два варіанти другого алгоритму. Обидва вони пов'язані зі спостереженнями фазової поведінки КП-КП передачі і стимулюють його відповідні переваги.

У першому варіанті (аналогічно як для алгоритму 1), кластеризується певна кількість  $L$  квантів планування в один квант, а потім приймається рішення планування для наступної  $L$  кількості квантів планування. Для всієї можливої комбінації з чотирьох ниток ( $N@4$ ) об'єднується попарно взаємодія для всіх  $L$  квантів планування, а потім нормалізується ця сума наступним чином:

$$\text{Нормована сума} = (\text{Сума} / (\text{max\_pair\_wise\_interaction} - \text{min\_pair\_wise\_interaction})) * (\text{max\_pair\_wise\_interaction});$$

Після цього відсортується кожен можливий кластер відповідно до нормованої суми. Після сортування відсортований список переглядається, а потім формуються кращі два або чотири кластери (для дво- або чотирьох-сокетних систем) використовуючи спрощений описаний вище алгоритм.

*Складність*

$$O((p/L)*N@4) + O((p/L)*((N@4)^2)) + O((p/L)*(N@4)) = O((p/L)*((N@4)^2));$$

$p$  = кількість планування квантів;  $N$  = кількість потоків;  $N@4$  =  $N$ -комбінованість 4;  $L$  = кількість квантів планування за якими одночасно ведеться спостереження, щоб прийняти рішення на наступні  $L$  квантів. Ці три умови є такими ж, як в алгоритмі 2 без зміни. Зараз просто слід запустити алгоритм для  $P/L$  квантів планування, тому що ведеться спостереження за  $L$  квантами одночасно.

*Варіант 2 алгоритму 2*

У другому варіанті (за аналогією з алгоритмом 1) також ведеться спостереження за базовою фазовою поведінкою, але використовується "алгоритм вивчення" щоб дізнатися про загальні закономірності, що відбуваються в  $L$  квантах планування, як описано вище. Отримується карта ниток для кожного кванта планування за допомогою спрощеного алгоритму, а потім застосовується алгоритм вивчення (описаний в алгоритмі 1), щоб дізнатися про які-небудь загальні закономірності між відображеннями. Грунтуючись на цих загальних шаблонах отримується нове відображення, яке буде використовуватися для наступних  $L$  квантів планування.

*Складність*

$$O(p*N@4) + O(p*((N@4)^2)) + O(p*(N@4)) + O((p/L)*((S^2*L + S*L)^2)) + O((p/L)*(S^2*L + S*L)) = O(p*((N@4)^2)) + O((p/L)*((S^2*L + S*L)^2))$$

$p$  = кількість квантів планування;  $N$  = кількість ниток;  $L$  = кількість квантів планування, за якими ведеться спостереження одночасно для прийняття рішення на наступні  $L$  квантів;  $S$  = кількість сокетів.

Перші три умови є такими ж, як описано в основному алгоритмі і його варіанті. Четверта і п'ята умови пов'язані з алгоритмом вивчення і були пояснені в варіанті 2 алгоритму 1.

*Алгоритм 3*

Цей алгоритм є покращенням у порівнянні з алгоритмом 2. Тут в кінці кожного кванта відмічаються лічильники від усіх ядер, а потім вибираються всі можливі комбінації групи з чотирьох ниток ( $N@4$ ) і підсумовується вся КП-КП передача між парами (буде сформовано 12 можливих пар (напрямлених ребер) КП-КП передачі між чотирма нитками). Потім відсортовуються суми в порядку спадання. Після сортування переглядається відсортований список, утворюючи всі комбінації груп  $M$  кластерів (де  $M$  являє собою число сокетів в системі), а потім вибирається ця група для  $M$  кластера з максимальною сумою спареної взаємодії поміж усіма  $M$  кластерами. Це може не виявити кращу комбінацію спарення для сполучення для наступного кванта (так як на даному етапі відсутні знання про майбутнє, а тільки можна спостерігати минуле або базову фазову поведінку, щоб покращити умови).

*Складність*

$$O(p*N@4) + O(p*((N@4)^2)) + O(p*((N@4)^2)) = O(p*((N@4)^2));$$

$p$  = кількість квантів планування;  $N$  = кількість ниток;  $N@4$  =  $N$ -комбінованість 4.

Перший доданок – це формування  $N@4$  парних взаємодій для всіх  $p$  квантів планування. Другий доданок вставляє попарні суми в відсортованому режимі. Самостійно встановлюється сортування вставки. Це має бути зроблено для кожного кванта планування. Отже,  $O(p*((N@4)^2))$ . Третя складова для перегляду відсортованого списку ( $N@4-1$ ) рази, щоб отримати глобально

кращу групу  $M$  кластерів. Це має бути зроблено для кожного з  $p$  квантів планування. Отже, отримуємо доданок  $O(p*((N\textcircled{4})^2))$ .

*Варіант 1 алгоритму 3*

Є два варіанти в цьому алгоритмі, як з алгоритмами 1 і 2. Обидва вони ґрунтуються на дослідженні базової фазової поведінки. У першому варіанті (за аналогією з алгоритмом 1) кластеризується певна  $L$  кількість квантів планування в один квант, а потім прийняття рішення планування для наступного  $L$  номера квантів планування. Для всіх можливих комбінацій з чотирьох ниток ( $N\textcircled{4}$ ) слідує об'єднання парно у взаємодію для всіх  $L$  квантів планування, а потім ця сума нормалізується:

$$\text{Нормована сума} = (\text{Сума} / (\text{max\_попарн\_вз} - \text{min\_попарн\_вз})) * (\text{max\_попарн\_вз});$$

Потім відсортовується кожен можливий кластер відповідно до нормованої суми. Після сортування переглядається відсортований список ( $N\textcircled{4}-1$  разів), а потім утворюються кращі два або чотири кластери (для дво- або чотирьох-сокетних систем) з використанням алгоритму 3.

*Складність*

$$O((p/L)*N\textcircled{4}) + O((p/L)*((N\textcircled{4})^2)) + O((p/L)*((N\textcircled{4})^2)) = O((p/L)*((N\textcircled{4})^2));$$

$p$  = кількість квантів планування;  $N$  = кількість ниток;  $L$  = кількість квантів планування,  $N\textcircled{4}$  =  $N$ -комбінованість 4,  $L$  = кількість квантів планування, що спостерігається одночасно для прийняття рішення на наступні  $L$  квантів.

Ці три умови є так само, як і в алгоритмі 3, без зміни, але тепер потрібно стартувати алгоритм для  $(P/L)$  квантів планування, тому що спостереження ведеться одночасно за  $L$  квантами.

*Варіант 2 алгоритму 3*

У другому варіанті (за аналогією з алгоритмом 1) також досліджується базова поведінка фази, але використовується "алгоритм вивчення", щоб дізнатися, які загальні закономірності відбуваються в  $L$  кванті планування, як описано вище. Зображаємо нитки для кожного кванта планування з використанням алгоритму 3 (без зміни), а потім застосуємо алгоритм вивчення (описаний вище в алгоритмі 1), щоб дізнатися про які-небудь загальні закономірності між відображеннями. Опираючись на ці загальні моделі отримується нове відображення, яке буде використовуватися для наступних  $L$  квантів планування.

*Складність*

$$O(p*N\textcircled{4}) + O(p*((N\textcircled{4})^2)) + O(p*((N\textcircled{4})^2)) + O((p/L)*((S^2*L + S*L)^2)) + O((p/L)*(S^2*L + S*L)) = O(p*((N\textcircled{4})^2)) + O((p/L)*((S^2*L + S*L)^2))$$

$p$  = кількість квантів планування,  $N$  = кількість ниток,  $L$  = кількість квантів планування, що спостерігається одночасно для прийняття рішення на наступні  $L$  квантів,  $S$  = кількість сокетів. Перші три умови є такими ж, як описано в основному алгоритмі і його варіанті представлення. Четвертий і п'ятий доданки пов'язані з алгоритмом вивчення, як описано в варіанті 2 алгоритму 1.

*Алгоритм 4*

Цей алгоритм є таким же, як алгоритм 3, але використовує невелику гнучкість для того, щоб вийти з алгоритму раніше, поки буде досягнутий такий же результат, як в алгоритмі 3. Коли здійснюється аналіз відсортованого списку, то відстежується доданок для максимальної попарної взаємодії для всіх комбінацій  $M$  кластерів проаналізованих досі (де  $M$  – кількість сокетів в системі). Коли досягається точка, в якій не прослідковується ніяка можливість перевищення максимуму, що спостерігався досі, то перед цим треба вийти з алгоритму. То ж, як швидко буде здійснено вихід з алгоритму – стає дуже залежним показником. Скільки буде зекономлено за рахунок швидкого виходу – подано в таблиці 4.

*Складність*

$$O(p*N\textcircled{4}) + O(p*((N\textcircled{4})^2)) + O(p*((N\textcircled{4})*E)) = O(p*((N\textcircled{4})^2))$$

$p$  = кількість квантів планування,  $N$  = кількість ниток,  $N\textcircled{4}$  =  $N$ -комбінованість 4,  $E$  = фактор передуючого виходу.

Перші два члени такі самі, як і в алгоритмі 3. Третій доданок для переходу до  $E$  раз відсортованого списку, для отримання глобальної кращої групи кластерів  $M$ . Коефіцієнт  $E$  дуже малий, часто менший за  $0.2*N\textcircled{4}$  (таблиця 4). Це має бути зроблено для кожного з  $p$  квантів планування. Отже, доданок набуває вигляду  $O(p*((N\textcircled{4})*E))$ .

*Варіант 1 алгоритму 4*

Є два варіанти цього алгоритму. Обидва вони засновані на спостереженні базової фазової поведінки.

У першому варіанті (за аналогією з алгоритмом 1), ми кластеризуємо певне  $L$  число квантів планування в один квант, а потім приймаємо рішення планування для наступного  $L$  числа квантів планування. Для всіх можливих комбінацій з чотирьох ниток ( $N \odot 4$ ) ми об'єднуємо попарну взаємодію для всіх  $L$  квантів планування, а потім нормалізувати цю суму:

$$\text{Нормована сума} = (\text{сума} / (\text{max\_попарн\_вз} - \text{min\_попарн\_вз})) * (\text{max\_попарн\_вз});$$

Потім відсортовується кожен можливий кластер відповідно до нормованої суми. Після сортування, аналізується відсортований список ( $E$  раз), а потім утворюється кращі два або чотири кластери (для дво- або чотирьох-сокетних систем) з використанням алгоритму 4.

*Складність*

$$O((p/L)*N \odot 4) + O((p/L)*(N \odot 4)^2) + O((p/L)*(N \odot 4)*E) = O((p/L)*(N \odot 4)^2)$$

$p$  = кількість квантів планування,  $N$  = кількість ниток,  $N \odot 4$  =  $N$ -комбінованість 4,  $E$  = фактор передуючого виходу,  $L$  = кількість квантів планування що спостерігається в один і той же час для прийняти рішення на наступні  $L$  квантів.

Ці три умови є так само, як в алгоритмі 4 без зміни, але слід стартувати алгоритм для  $(P/L)$  квантів планування, тому що спостереження ведеться за  $L$ -квантами одночасно.

*Варіант 2 алгоритму 4*

У другому варіанті (за аналогією з алгоритмом 1) також ведеться спостереження за базовою фазовою поведінкою, але використовується "алгоритм вивчення" щоб дізнатися про будь-які загальні закономірності, що відбуваються в  $L$  квантів планування. Зображаються нитки для кожного кванта планування з використанням алгоритму 4 (без зміни), а потім застосовується алгоритм вивчення (вже описаний), щоб дізнатися про які-небудь загальні закономірності між відображеннями. Опираючись на ці загальні підходи отримується нова карта на наступні  $L$  квантів планування.

*Складність*

$$O(p*N \odot 4) + O(p*(N \odot 4)^2) + O(p*(N \odot 4)*E) + O((p/L)*(S^2*L + S*L)^2) + O((p/L)*(S^2*L + S*L)) = O(p*(N \odot 4)^2) + O((p/L)*(S^2*L + S*L)^2)$$

$p$  = кількість квантів планування,  $N$  = кількість ниток,  $N \odot 4$  =  $N$ -комбінованість 4,  $E$  = фактор передуючого виходу,  $L$  = кількість квантів планування що спостерігається в один і той же час для прийняти рішення на наступні  $L$  квантів,  $S$  = кількість сокетів.

Перші три умови є такими ж, як описано в основному алгоритмі і його варіанті. Четвертий і п'ятий доданки відображають алгоритм вивчення, що роз'яснюється в варіанті 2 алгоритму 1.

### Методологія

Таблиця 1. Параметри тестування

Тестування	К-сть сокетів	Довжина інструкцій (в мільйонах)	Тестування	К-сть сокетів	Довжина інструкцій (в мільйонах)
Бізнес-обробка	2	74	SPEC WEB 05	4	140
Бізнес-обробка	4	149	Робочий стіл мультикористувачів	2	67
SPEC JBB 2000	2	145	Робочий стіл мультикористувачів	4	135
SPEC JBB 2000	4	145	робоче навантаження 1 для бази даних (БД)	2	72
SPEC JBB 2005	2	107	робоче навантаження 1 для БД	4	145
SPEC JBB 2005	4	120	робоче навантаження 2 для БД	4	145
SPEC JBB 2005-2	2	124	робоче навантаження 3 для БД	4	140
SPEC JBB 2005-2	4	124	робоче навантаження 4 для БД	2	68
SPEC WEB 05	2	70	робоче навантаження 4 для БД	4	136

Було використано внутрішній симулятор AMD для запуску 18 серверних слідів для дво- і чотирьох-сокетної конфігурації для збору статистики передачі КП-КП серед усіх пар ниток. Ці статистичні дані потім подаються в програму, яка імітує алгоритми планування, описані вище, і



дає загальний вклад (зниження відсотків у КП-КП передачі) в сумі досягнутих кожним алгоритмом. Критерії перераховано в таблиці 1, а параметри конфігурації – в таблиці 2. Допускається 5 мс кванти планування на процесорі частотою 2 ГГц. У середньому, моделювання проводилося для 28 квантів.

Таблиця 2. Параметри архітектури

Параметри архітектури	Значення
Кількість сокетів	2 або 4
Кількість ядер на сокет	4
Л1 ДКеш (на ядро)	64 КВ, 2-шляхи, 64-байтовий блок
Л2 Кеш (на ядро)	512 КВ, 16-шляхів, 64-байтовий блок
Л3 Кеш (спільний кеш – 1 на сокет)	6МВ, 48-шляхів, 64- байтовий блок
Протокол кеш-когерентності	MOESI
Мережа внутрішнього зв'язку	Лінія для 2 сокетів (сокет0 <-> сокет1), Квадрат для 4 сокетів (сокет0 <-> сокет1 <-> сокет2 <-> сокет3 <-> сокет0)

### Результати

Результати для двохсокетної і чотирьохсокетної конфігурацій для всіх чотирьох алгоритмів наведені в таблиці 3. У позначенні, наприклад, A1p1, A – позначає алгоритм, 1 – позначає число алгоритмів, p – позначає базову фазову поведінку, і 1 – позначає алгоритм вивчення. Для базової фазової зміни спостерігаються одночасно два кванти планування для того, щоб прийняти рішення для наступних двох квантів.

Таблиця 3. Процентне зниження для міжсокетної КП-КП передачі на алгоритм

№ серії	Тест	A1	A1p	A1pl	A2	A2p	A2pl	A3	A3p	A3pl	A4	A4p	A4pl
1	Бізнес-обробка (2)	14	22	16	18	2	22	18	2	22	18	2	22
2	Бізнес-обробка (4)	35	37	34	47	19	49	47	19	49	47	19	49
3	SPEC JBB 2000 (2)	-0.7	-0.5	-0.5	1	-1	1	3	-1	3	3	-1	3
4	SPEC JBB 2000 (4)	1.6	0.7	-0.6	2	0	2	3	0	3	3	0	3
5	SPEC JBB 2005 (2)	5.9	2.5	2.5	12	6	12	15	6	15	15	6	15
6	SPEC JBB 2005 (4)	5.5	3.9	3.8	10	1	10	11	1	10	11	1	10
7	SPEC JBB 2005-2 (2)	6.9	0.3	0.6	6	-3.5	5.3	8.3	-3.3	8.8	8.3	-3.3	8.8
8	SPEC JBB 2005-2 (4)	9.3	8.4	9	10	-1	10.4	11	-1	11.1	11	-1	11.1
9	SPEC WEB 05 (2)	-13	-8.4	-7.1	-6	-24	-5.4	-2	-24	0	-2	-24	0
10	SPEC WEB 05 (4)	32	29	27	45	13.7	41.4	48	13.8	46	48	13.8	46
11	Робочий стіл мультикористувачів (2)	18	19	17.4	13.8	10.1	12.3	25	10.1	25	25	10.1	25
12	Робочий стіл мультикористувачів (4)	3.4	3	-14	15	-22	12.3	25	-19	25	25	-19	25
13	Робоче навантаж-я 1 для БД (2)	5.7	-11	-11	99.3	-12	99	99.3	-11	99	99.3	-11	99

14	Робоче навантаж-я 1 для БД (4)	42	38	28	99.2	-20	99	99.2	-18	99	99.2	-18	99.2
15	Робоче навантаж-я 2 для БД (4)	2	1.4	0.8	2.2	0.96	1.6	3.64	0.96	2.47	3.64	0.96	2.47
16	Робоче навантаж-я 3 для БД (4)	-21	-21	-21	0	-44	0	0	-38	0	0	-38	0
17	Робоче навантаж-я 2 для БД (4)	3.3	3.4	3.4	3	2.4	3.3	5.6	2.4	6.4	5.6	2.4	6.4
18	Робоче навантаж-я 4 для БД (4)	-11	-15	-16	3	-27	3.5	4.1	-26	4.98	4.1	-26	4.98
Середнє		7.7	6.2	4	21.1	-5.5	21	23.5	-4.8	23.88	23.5	-4.8	23.78

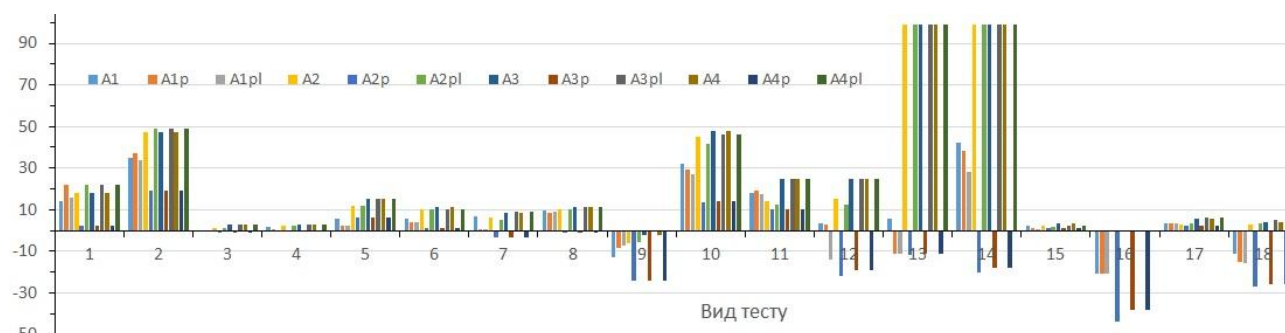


Рис. 2. Відсоткове зниження в міжсокетній КП-КП передачі на алгоритм (у тесті вісь Х визначається за серійним номером у таблиці 3, а середнє значення не наводиться)

Для результатів, зображених на малюнку 2 (із таблиці 3) видно, що середній розмір вкладу від алгоритмів (відсоткове зниження в міжсокетній КП-КП передачі) коливається від 5,5% до 24% (будь ласка зверніть увагу, що в графіку не показано алгоритм 4, тому що вклад від нього по суті такий же, як від алгоритму 3). Отримано збільшення КП-КП передачі тому що спарювання знайдено за заданим алгоритмом для заданого числа фаз не може підтримувати наступні кілька фаз. Алгоритм 4 з використанням базового фазового алгоритму вивчення дає нам великий вклад. Алгоритм 2 (скорочений алгоритм) без базового фазового вивчення пропонує майже такий же великий вклад (21%), не будучи дуже складним. Є додатки, в яких для алгоритму вивчення передбачається (наприклад, бізнес-обробка, робоче навантаження 4 для БД з двосокетною та чотирьохсокетною конфігурацією), в середньому, алгоритм 2 (звичайна версія) який виконується досить добре. Робоче навантаження 1 для БД виділяється з максимальним вкладом 99,2%. За браком місця загальний вклад не перераховується під час виконання кожної програми. Але, як кращий варіант, для робочого навантаження 1 для БД (як двох- так і чотирьохсокетного) отримано приблизно вклад в розмірі 10% під час виконання.

Загалом, для малих серверів середньої дальності алгоритм 2 дає хороший компроміс між продуктивністю і складністю. Для великих серверів простий алгоритм може бути кращим вибором; він також надає додатковий вклад, що може бути реалізовано в розподіленій/паралельній формі.

У таблиці 4 ми представляємо відсоткове зниження числа ітерацій циклу, отриманих за рахунок виконання дострокового виходу в алгоритмі 4. Для більшості додатків зменшення відсотків спостерігається на рівнях 80-и або 90-а. Це говорить, що якщо ми хочемо вибрати кращий алгоритм виконання, то слід враховувати допомогу здійснення раннього виходу для реальних додатків.

#### Споріднені дослідження

Алгоритми кластеризації ниток були розглянуті Thekkath і Еггерс [2]. Їх дослідження базується на тому, щоб знайти кращий спосіб угруповання ниток, які з'єднують області пам'яті на тому ж процесорі так, щоб максимізувати кеш-обмін і повторно його використати. Однак, шлях, якого вони дотримуються, є на базі адрес і може бути схильний до помилок, особливо в міграціях обміну. Якщо процеси P0, P1, P2 беруть участь в міграційному обміні, і якщо обирається

P0 і P1 щоб запланувати на той же сокет, а рядок кеш-пам'яті насправді мігрує від P0 до P2, а потім назад на P1, то зниження в міжсокетній КП-КП передачі не спостерігатиметься. У цій роботі відстежується обмін з використанням попарних лічильників і представляється найкращий глобальний алгоритм. В роботі припускається, що число ниток відповідає кількості процесорів, іншими словами, система завжди працює з балансуванням навантаження. Вивчення запропонованих алгоритмів також проходить на більш реальних тестуваннях на базі комерційних серверів.

Наша робота найбільше нагадує роботу, виконану авторами роботи [1], яка також пропонує використання лічильників продуктивності апаратного забезпечення для кластеризації ниток в багато процесорних системах з поєднаним використанням пам'яті. Вони використовували легкий алгоритм кластеризації і повідомили про зниження між віддаленими стійками кеша до 70%. У цій роботі використовуються чотири різних алгоритми планування і повідомляється про скорочення у віддалених недостатках кеш-пам'яті до 99,3%. Крім того, використовується інший підхід, щоб зібрати міжсокетні КП-КП передачі за допомогою лічильників продуктивності. Використання віддалених стійок кеша на діапазон адрес не вважається найкращим, так як це може приводити до проблеми спільного використання пам'яті, описаної вище, а замість цього підраховуються попарні передачі віддаленого кешу. Такий підхід працює для серверів малого і середнього радіусу дії.

Для великих машин (таких як Cray Red Storm, Aprro International Atlas [3], і т.д.), можна використовувати грубо- та дрібнокластерний підхід такий як поєднання двох досить близьких сокетів в один "логічний сокет". Це пов'язано з тим, що, навіть якщо не можна запланувати високовзаємодіючі нитки на одному і тому ж сокеті, то можна запланувати їх на близьких сокетах, щоб зменшити кількість переходів. Припускається, що базовий вузол для всіх КП-КП передач для обраного кластера ниток попадає в цей сокет. Один із способів вирішення цієї проблеми є підтримання в кожному лічильнику головного вузла підрахунку КП-КП передач між кожною парою ниток, а потім взяття основного вузла до уваги для реалізації алгоритму планування.

Таблиця 4. Процентне зниження в ітераціях за алгоритмом 4

Тестування	К-сть сокетів	% спадання (в циклічних підрахунках)	Тестування	К-сть сокетів	% спадання (в циклічних підрахунках)
Бізнес-обробка	2	96	Робочий стіл мультикористувачів	2	86
Бізнес-обробка	4	99	Робочий стіл мультикористувачів	4	99
SPEC JBB 2000	2	50	Робоче навантаження 1 для бази даних (БД)	2	97
SPEC JBB 2000	4	73	Робоче навантаження 1 для БД	4	99.8
SPEC JBB 2005	2	61	Робоче навантаження 2 для БД	4	88
SPEC JBB 2005	4	77	Робоче навантаження 3 для БД	4	99
SPEC JBB 2005-2	2	75	Робоче навантаження 4 для БД	2	71
SPEC JBB 2005-2	4	91	Робоче навантаження 4 для БД	4	97.4
SPEC WEB 05	2	84			
SPEC WEB 05	4	98			

В роботі [4] автори досліджували підходи, щоб виявити блокування обміну в просторі користувача між мультинитковими додатками за допомогою бібліотек анутовання синхронізації на рівні користувача. На базі цієї інформації нитки, використовуючи те ж блокування, мігрують на

тому ж процесорі. Дана робота проводиться в тому ж дусі, але на більш загальному рівні, який може застосовуватися до будь-якого виду обміну області пам'яті.

Багато вчених досліджували підтримку ОС для мінімізації конфлікту кеш-пам'яті і проблеми пропускну здатності розподілених кеш-процесорів L2/L3 [5,6,7]. Однак дана робота спрямована на зменшення впливу недостатків зв'язку, які притаманні додаткам і може бути доповненням до цих робіт.

**Висновки та перспективи подальшого дослідження.** Багато комерційних серверних додатків сьогодні працюють на мультисокетних багатоядерних серверах. Такі програми, як правило, терплять проблеми від міжсокетних недостатків зв'язку, що зменшує їх загальну продуктивність. В роботі представлено операційну систему оптимізації з метою зниження наслідків недостатків міжсокетного зв'язку. Спостерігаючи міжсокетну КП-КП передачу між різними нитками і додаткові включення для рішень планування ОС, міжсокетні недатки зв'язку для деяких додатків значно знижуються. З чотирьох різних алгоритмів вивчених на 18 реальних тестах, простий алгоритм зменшує міжсокетні недатки зв'язку аж до 42% (в середньому = 7,7%) і варіації з трьох інших складних алгоритмів зменшують міжсокетну взаємодію до 99,3 % (в середньому – від 21% до 24%). В цілому, для малих і середніх серверів, спрощений алгоритм пропонує хороший компроміс між продуктивністю і складністю. Для великих серверів спрощений алгоритм може бути кращим вибором між продуктивністю та складністю, а простий алгоритм може бути кращим вибором.

Дана робота може бути поліпшена в декількох напрямках. Можна використовувати базову фазову поведінку на основі дослідження останньої L фази для прийняття рішення планування тільки для наступної фази замість наступних L фаз. Можна також використовувати кращі алгоритми машинного вивчення щоб дізнатися про будь-які зміни фазової поведінки між квантів планування і включити це в рішення планування. Загалом, для будь-якого довгострокового додатка з стійкими закономірностями, апаратне забезпечення може забезпечити зворотний зв'язок з ОС, яка в свою чергу може використовувати цю інформацію для адаптації своєї політики на користь продуктивності додатків.

1. Tam D., Azimi R., Stumm M. Thread Clustering: Sharing-Aware Scheduling on SMP-CMPSMT Multiprocessors. In ACM SIGOPS Operating System Review. June 2007.
2. Thekkath R., Eggers S.J. Impact of sharing based thread placement on multi-threaded architectures. In Int-l. Symp. on Computer Architecture. 1994.
3. [www.top500.org](http://www.top500.org)
4. Sridharan S. *et al.* Thread migration to improve synchronization performance. In Workshop on Operating System Interference in High Performance Applications, 2006.
5. Nakajima J. *et al.* Enhancements for Hyper-Threading technology in the operating system – seeking the optimal micro-architectural scheduling. In Intl. Parallel and Distributed Processing Symp. 2005.
6. Snavely A. *et al.* Symbiotic job scheduling for a simultaneous multithreading processor. In Conf. On Architectural Support for Programming Languages and Operating Systems, 2000.
7. El-Moursy *et al.* Compatible phase coscheduling on a CMP of multi-threaded processors. In Intl. Parallel and Distributed Processing Symp. 2006.

УДК 681.515.8

Здолбіцька Н.В. к.т.н. доц., Здолбіцький А.П., Найдюк Ю.Ю.

Луцький національний технічний університет

## ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИБОРУ РЕЖИМУ ТРЕНУВАННЯ НА БАЗІ ОС ANDROID

**Здолбіцька Н.В., Здолбіцький А.П., Найдюк Ю.Ю. Програмне забезпечення для вибору режиму тренування на базі ОС Android.** Розроблено програмне забезпечення для можливості підбору оптимального та найкращого режиму занять в тренажерному залі, яке працюватиме на ОС Android. Мобільний додаток може служити посібником, довідником та помічником для великої кількості людей, які займаються спортом, слідкують та покращують власне здоров'я.

**Ключові слова:** android, програмне забезпечення, база даних, додаток, activity, sqlite, xml.

**Рис. 3. Літ. 12.**

**Здолбицкая Н.В., Здолбицкий А.П., Найдюк Ю.Ю. Программное обеспечение для выбора режима тренировки на базе ОС Android.** Разработано программное обеспечение для возможности подбора оптимального и лучшего режима занятий в тренажерном зале, которое будет работать на ОС Android. Мобильное приложение может служить пособием, справочником и помощником для большого количества людей, которые занимаются спортом, следят и улучшают свое здоровье.

**Ключевые слова:** android, программное обеспечение, база данных, приложение, activity, sqlite, xml.

**Zdolbitska N., Zdolbitsky A., Naidiyk Y. Software training mode selection based on Android OS.** Software to enable selection of the optimal and best treatment sessions in the gym, which will run on the operating system Android are developed. Mobile application can serve as a guide and assistant guide for a large number of people involved in sports, follow and improve their health.

**Keywords:** android, software, database, application, activity, sqlite, xml.

**Постановка проблеми.** В ході роботи була поставлена задача створення додатка довідника для мобільних пристроїв на операційній системі Android. Основною ідеєю була реалізація можливості вибору індивідуальної програми режиму занять в тренажерному залі.

Мовою програмування Android є Java. Для розробки використовується середовище розробки Android Studio, яке є надбудовою AndroidSDK і вбудованим в SDK емулятором мобільного приладу. Для створення та редагування растрових зображень використовувався Adobe Photoshop з дизайнерського пакету Creative Suite.

Також в процесі розробки документації були використані Enterprise Architect та Microsoft Office.

**Метою роботи** є створення програмного забезпечення для вибору режиму тренування, вивчення принципів роботи операційної системи Android, а також вивчення основних можливостей та функцій системи.

Однією з головних проблем розробки додатків на платформі Android є велика фрагментація приладів з різним розширенням екрана мів та версіями операційної системи.

Фрагментація девайсів збільшується, і разом з цим росте вибір Android. Виробники більш дешевих пристроїв намагатимуться використовувати останні версії Android, а фрагментація сприятиме екосистемі ставати більш глобальною і соціально-економічною.

Враховуючи статистику фрагментацій екранів та версії ОС (рис. 1), виберемо оптимальний пристрій для розроблення додатку.

Фрагментація географічно розширює ринок Android – вона не обмежується лише розсіюванням пристроїв і ОС. Вона також важлива для розуміння, які пристрої зараз актуальні, які девайси можуть бути актуальні для одного регіону, а для іншого навпаки, розробникам потрібно враховувати відмінності в продуктивності мережі в різних регіонах при розробці додатків. Інший фактор – це термін життя батареї; якщо сьогодні термін життя батареї цілком прийнятний, то в майбутньому її може не вистачати.

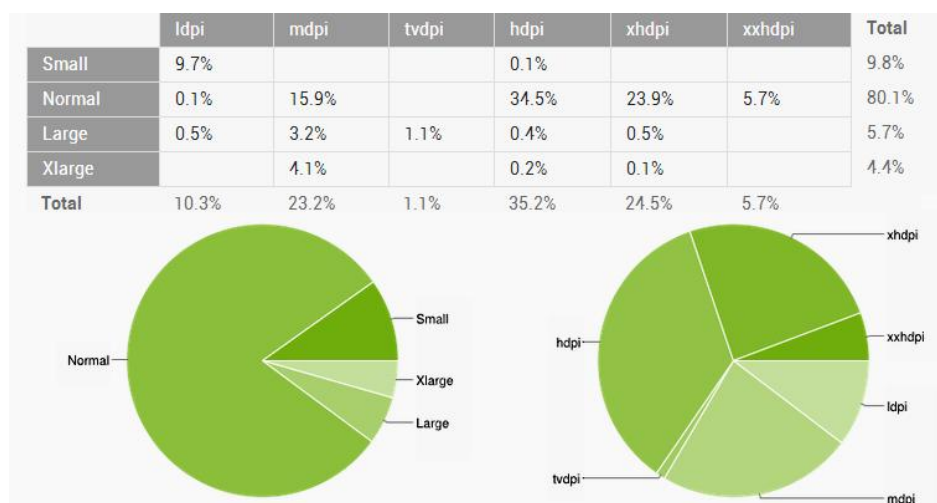


Рис. 1 – Фрагментація екранів

Згідно статистичним даним, при створенні додатку вибрано версію компіляції коду – Jelly Bean 4.1 API16 (37,3%) (мінімальна версія Gingerbread 2.3.3 API 10 (36,3%)). Розробка велася на пристрої з розміром hdpi normal (34,5%), а саме Motorola MB526 та AVD емулятор.

В даній статті створено додаток для вибору програми тренувань в тренажерному залі. Додаток містить довідник вправ, інформацію про калорійність їжі та спортивне харчування, тест вибору оптимальних вправ користувача та їх збереження в файл, секундомір з можливістю послідовного фіксування часу, форму підрахунку ідеальної маси тіла, синхронізацію з тематичними веб-ресурсами та можливості зворотнього зв'язку з розробником.

**AndroidManifest.xml.** Файл маніфесту AndroidManifest.xml надає основну інформацію системі про програму. Кожна програма повинна мати свій файл AndroidManifest.xml. Редагувати файл маніфесту можна вручну, змінюючи XML-код або через візуальний редактор Manifest Editor (Редактор файлу маніфесту), який дозволяє здійснювати візуальне і текстове редагування файлу маніфесту додатки.

Призначення файлу:

- оголошує ім'я Java-пакета додатка, який є унікальним ідентифікатором;
- описує компоненти програми – діяльності, служби, приймачі широкомовних намірів і контент-провайдери, що дозволяє викликати класи, які реалізують кожний з компонентів;
- містить список необхідних дозволів для звернення до захищених частин API та взаємодії з іншими додатками;
- оголошує дозволи, які сторонні додатки зобов'язані мати для взаємодії з компонентами цього додатка;
- оголошує мінімальний рівень API Android, необхідний для роботи програми;
- перераховує пов'язані бібліотеки;

Файл маніфесту інкапсулює всю архітектуру Android додатка, його функціональні можливості і конфігурацію. У процесі розробки програми файл постійно редагується, змінюється його структура, доповнюючись новими елементами і атрибутами.

Кореневим елементом маніфесту є <manifest>. Крім даного елемента обов'язковими елементами є теги <application> і <uses-sdk>. Елемент <application> є основним елементом маніфесту і містить безліч дочірніх елементів, що визначають структуру і роботу програми. Порядок розташування елементів, що знаходяться на одному рівні, довільний. Всі значення встановлюються через атрибути елементів. Крім обов'язкових елементів, згаданих вище, в маніфесті по мірі необхідності використовуються інші елементи.

Початок файлу, а саме налаштування додатку, має вигляд:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.naidiuk.mp" >
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CALL_PHONE"/>
```

```
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.VIBRATE" />
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme"
    android:orientation="vertical" >
    ...
```

**База даних SQLite.** SQLite доступний на будь-якому Android-пристрої, його не потрібно встановлювати окремо.

SQLite підтримує типи TEXT (аналог String в Java), INTEGER (аналог long в Java) і REAL (аналог double в Java). Решта типів слід конвертувати, перш ніж зберігати в базі даних. SQLite сама по собі не перевіряє типи даних, тому можна записати ціле число в колонку, призначену для рядків і навпаки.

Для нашого додатку потрібна досить проста база даних, яка міститиме лише колонку name, у якій будуть вказані назви вправ. Дані вправи виймаються з бази по результатам тесту, який проходить користувач.

Для обробки описаних вище ситуацій нам треба створити клас, який є спадкоємцем для SQLiteOpenHelper. Назвемо його sqllite. Цей клас надасть нам методи для створення або поновлення БД у випадках її відсутності або старіння.

При реалізації цього допоміжного класу ховається логіка, на основі якої приймається рішення про створення або оновлення бази даних перед її відкриттям. Клас SQLiteOpenHelper містить два абстрактних методи:

onCreate() – метод, який буде викликаний, якщо БД, до якої ми хочемо підключитися – не існує.

onUpgrade() – буде викликаний в разі, якщо ми намагаємося підключитися до БД більш нової версії, ніж існуюча.

**Файл стилів styles.xml.** Щоб наш додаток був якісним, можна скористатися спеціальною темою. Тема – це колекції стилів, які забезпечують професійний вид додатка, щоб він був схожий на рідний додаток Android. Сама система Android вже має кілька попередньо встановлених тем, якими можна скористатися в своїх цілях. Для цього достатньо тільки вказати ім'я теми в маніфесті.

Щоб полегшити громіздку роботу над створенням стилів інтерфейсу, було використано веб-ресурс Android Action Bar Style Generator. Так як вид Activity при створенні проекту був вибраний з Action Bar.

Даний ресурс створює набір графічних 9patch елементів та файли стилів, в тому числі style.xml, яким потрібно замінити стандартний.

У генераторі є можливість вибору таких властивостей:

- Style name – назва стилю;
- Style compatibility – сумісність стилю (Holo, App Compat, Sherlock);
- Base theme – базова тема (світла, темна, світла з темним Action Bar);
- Action bar style – стиль Action Bar (монотонний, прозорий);
- Action bar texture – текстурованість Action bar;
- Tab hairline style – лінії відмежування для табів;
- Neutral pressed states – звичайні натиснення (стандартні для версії KitKat);
- Action bar color – колір Action bar;
- Stacked color – колір табів;
- Tab indicator color – колір індикатора табів;
- Popup color – колір меню;
- Accent color – колір вибраного елемента;
- Action mode background color – колір заднього фону;
- Action mode highlight color – колір переднього фону.

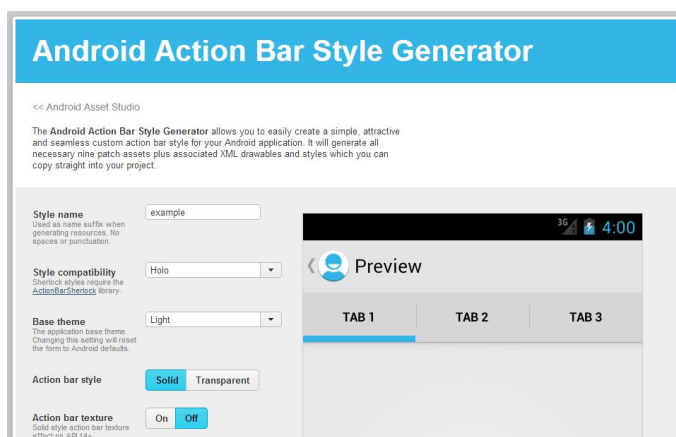


Рис. 2 – Вікно генератора стилів

**Фрагменти в Android.** Існує два основних підходи у використанні фрагментів.

Перший спосіб заснований на заміщенні батьківського контейнера. Створюється стандартна розмітка і в тому місці, де будуть використовуватися фрагменти, розміщується контейнер, наприклад, `LinearLayout`. У коді контейнер заміщається фрагментом. При використанні подібного сценарію в розмітці не використовується тег фрагмента, оскільки його не можна міняти динамічно. Також вам доведеться оновлювати `ActionBar`, якщо він залежить від фрагмента.

Другий варіант – використовуються окремі розмітки для телефонів і планшетів, які можна розмістити в різних папках ресурсів. Наприклад, якщо в планшеті використовується двохпанельна розмітка з двома фрагментами на одній активності, тоді використовується ця ж активність для телефону, але з підключенням іншої розмітки, яка містить один фрагмент. Для перемикання на другий фрагмент запускається друга `Activity`.

У додатку застосовано перший варіант, для відображення кількох табів з контейнерами поверх одного `Activity`.

Клас фрагменту подібно класу активності посилається на `xml` файл інтерфейсу, але має іншу конструкцію в якій метод `onCreate()` замінюється на `onCreateView()`.

**Меню додатка.** За меню відповідає клас `android.view.Menu`. Кожна активність пов'язана з одним об'єктом меню. Саме меню містить пункти меню (клас `android.view.MenuItem`) і підменю (клас `android.view.SubMenu`).

При натисканні кнопки меню на старіших моделях телефонів з'являється набір пунктів меню, що прикріплюється до активності. Меню може містити іконки. Таке меню може містити шість пунктів (як правило). При наявності більше шести пунктів використовується розширене меню – в цьому випадку замість шостого пункту з'являється пункт Параметри (Докладніше). При натисканні даного пункту показується розширене меню зі списком пунктів, які не вмістилися в основній частині меню вибору опцій.

Коли меню відкривається вперше, `Android` викликає метод `onCreateOptionsMenu()`, передаючи як параметр об'єкт меню. Меню можна створювати у вигляді ресурсів в `xml` файлі або використовувати метод `add()`.

У стандартному проекті при виборі звичайного шаблону вже є пункт для меню з одного пункту «Налаштування» і виклик методу для меню.

Розглянемо роботу меню через ресурси. Для створення меню використовуються ресурси, які повинні зберігатися в `xml` файлі. Меню складається з наступних елементів:

– `<menu>` – визначає меню, яке буде містити пункти меню. Елемент `<menu>` повинен бути кореневим елементом в структурі файлу і може містити один або декілька елементів `<item>` і `<group>`;

– `<item>` – створює безпосередньо пункти меню. Даний елемент може мати вкладений елемент `<menu>` для створення підменю;

– `<group>` – при бажанні можна використовувати невидимий контейнер для елементів `<item>`. Це дозволяє досягти деяких ефектів.



Меню додатку, що розробляється, може викликатися лише у MainActivity.java, в якому описано три метода для його керування: onCreateOptionsMenu(), onPrepareOptionsMenu(), onOptionsItemSelected().

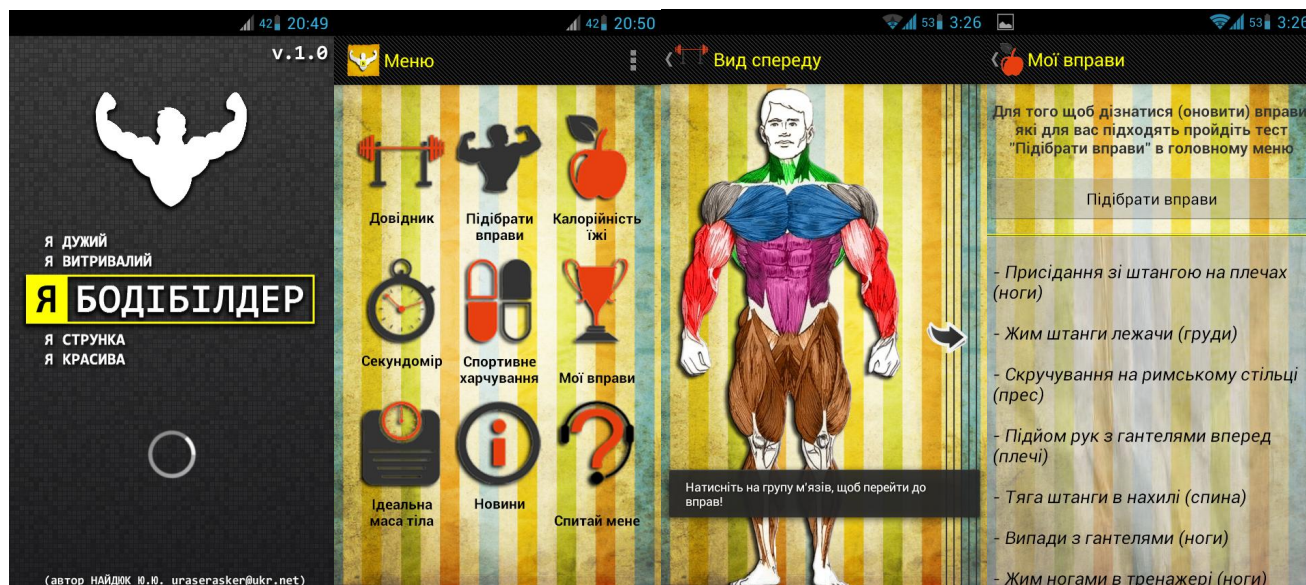


Рис. 3 – Інтерфейс додатка

### Висновки

Оглянувши всі найбільш поширені мобільні операційні системи було обрано ту, яка найбільш поширена, найбільш динамічно розвивається, має найбільший попит на ринку девайсів та найбільш відкрита для сторонніх розробників та програмістів.

Була поставлена задача створення додатка довідника для мобільних пристроїв на операційній системі Android. Основною ідеєю була реалізація можливості вибору індивідуальної програми режиму занять в тренажерному залі.

Додаток представляє собою «електронний довідник» спортсмена, бодібілдера, використовуючи який він може покращити свої показники в спорті, підтримувати задовільний стан здоров'я та вигляд тіла.

Слід відмітити, що даний прототип потрібно розвивати та оновлювати. Перечислимо деякі ідеї для подальшого удосконалення системи:

- врахування в подальшому зміни фрагментації девайсів;
- можливість редагування та самостійного підбору програм та вправ;
- розширення бази даних занять;
- створення онлайн справки;
- введення статистичних вимірів та порівняльних таблиць результатів занять;
- впровадження нових функцій та можливосте й ОС Android.

1. Rumbaugh J. Getting Started – Using Use Cases to Capture Requirements. // Journal of Object Oriented Programming. - 1994. Vol. 7, № 5. P. 8–12.
2. Блэйк Мик Программирование под Android. - СПб.: Санкт-Петербург, 2012. - 496 с.
3. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. 2-е изд. - М.: ДМК Пресс, 2007.
4. Гленфорд Майерс, Том Баджетт, Кори Сандлер Искусство тестирования программ, 3-е издание = The Art of Software Testing, 3rd Edition. - М.: "Диалектика", 2012. - 272 с.
5. Голощапов А.Л. Google Android: Программирование для мобильных устройств. - СПб.: БХВ-Петербург, 2011.
6. Эйблсон Ф. Разработка приложений для платформы Android в среде Eclipse.
7. Жвалевский Андрей Смартфоны Android без напряжения. Руководство пользователя. - СПб.: Санкт-Петербург, 2012. - 224 с.
8. Рето Майер Программирование приложений для планшетных компьютеров и смартфонов. - СПб.: Санкт-Петербург, 2011. - 672 с.
9. Хашими С., Коматинени С., Маклин Д. Разработка приложений
10. <http://developer.android.com/index.html>
11. <http://startandroid.ru/ru/>
12. <http://developer.alexanderklimov.ru/android/>

УДК 004.415.3

Пех П.А. к.т.н. доц., Мельник В.М. к.ф-м.н. доц., Островський М.В.  
Луцький національний технічний університет

## C++BUILDER ПРОЕКТ ІНТЕРПОЛЯЦІЇ ФУНКЦІЙ МНОГОЧЛЕНАМИ НЬЮТОНА ТА ЛАГРАНЖА

**Пех П.А., Мельник В.М., Островський М.В. C++Builder проект інтерполяції функцій многочленами Ньютона та Лагранжа.** В статті запропоновано C++Builder проект для побудови інтерполяційних многочленів Ньютона та Лагранжа.

**Ключові слова:** C++Builder проект, інтерполяція функцій, многочлен Ньютона, многочлен Лагранжа.

**Пех П.А., Мельник В.М., Островський М.В. C++Builder проект інтерполяції функцій многочленами Ньютона та Лагранжа.** В статті пропонується C++Builder проект для побудови інтерполяційних многочленів Ньютона та Лагранжа.

**Ключевые слова:** C++Builder проект, интерполяция функций, многочлен Ньютона, многочлен Лагранжа.

**Pekh P., Melnyk V., Ostrovsky M. C++Builder project for interpolation functions by polynomial Newton and Lagrange.** In this article is designed the C++Builder project to build an interpolation functions by polynomial Newton and Lagrange.

**Keywords:** C++Builder project, interpolation functions polynomial Newton, polynomial Lagrange.

**Постановка задачі.** Нехай невідому функцію  $y = f(x)$  задано своїми числовими значеннями  $y_0 = f(x_0)$ ,  $y_1 = f(x_1)$ , ...,  $y_n = f(x_n)$  у точках  $x_0, x_1, x_2, \dots, x_n$ , які називаються вузлами інтерполяції. Задача інтерполяції функції многочленом  $P_n(x)$  степеня  $n$  полягає у знаходженні многочлена такого вигляду [4]:

$$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2) + \dots + a_n(x - x_0)(x - x_1)\dots(x - x_{n-1}), \quad (1)$$

де  $x_0, x_1, \dots, x_{n-1}$  — задані значення аргументу  $x$ , а  $a_0, a_1, \dots, a_n$  — невідомі коефіцієнти, причому многочлен  $P_n(x)$  у вузлах інтерполяції має приймати такі ж значення, що й задана функція  $y = f(x)$ . Визначити коефіцієнти  $a_0, a_1, \dots, a_n$  і означає розв'язати задачу інтерполяції, оскільки знайдений за таких умов многочлен дозволить обчислити значення заданої функції у довільній точці.

Метою даного дослідження є розроблення засобами середовища C++Builder [1,2,3] проекту, який дозволяє автоматизувати процес побудови інтерполяційних многочленів за формулами Ньютона та Лагранжа.

**Основна частина.** З математичної точки зору задача інтерполяції функції многочленами може бути розв'язана багатьма методами. У даній статті аналізується, як ця задача розв'язується за допомогою першої та другої інтерполяційних формул Ньютона, у яких вузли інтерполяції є рівновіддаленими, а також за допомогою інтерполяційної формули Лагранжа для випадку нерівновіддалених вузлів інтерполяції.

Звичайно, побудова інтерполяційних формул значно спрощується, коли вузли інтерполяції  $x_0, x_1, \dots, x_n$  — рівновіддалені, тобто крок інтерполяції є постійним:

$$h = x_{i+1} - x_i = \text{const} \quad (i = 0, 1, \dots, n-1). \quad (2)$$

В інтерполяційних формулах з рівновіддаленими вузлами інтерполяції використовується поняття скінченних різниць. Назвемо скінченною різницею першого порядку різницю між значеннями функції у сусідніх вузлах інтерполяції:

$$y_1 - y_0 = \Delta y_0; \quad y_2 - y_1 = \Delta y_1; \quad \dots; \quad y_n - y_{n-1} = \Delta y_{n-1}. \quad (3)$$

Зі скінченних різниць першого порядку можна одержати скінченні різниці другого порядку:

$$\Delta^2 y_0 = \Delta y_1 - \Delta y_0; \quad \Delta^2 y_1 = \Delta y_2 - \Delta y_1; \quad \dots; \quad \Delta^2 y_{n-2} = \Delta y_{n-1} - \Delta y_{n-2} \quad (4)$$

Скінченні різниці  $n$ -го порядку визначаються формулами:

$$\Delta^n y_0 = \Delta^{n-1} y_1 - \Delta^{n-1} y_0; \Delta^n y_1 = \Delta^{n-1} y_2 - \Delta^{n-1} y_1; \dots \Delta^n y_i = \Delta^{n-1} y_{i+1} - \Delta^{n-1} y_i, \dots \quad (5)$$

Різниці довільних порядків можуть бути виражені безпосередньо через значення функції:

$$\begin{aligned} \Delta y_i &= y_{i+1} - y_i; \\ \Delta^2 y_i &= \Delta y_{i+1} - \Delta y_i = (y_{i+2} - y_{i+1}) - (y_{i+1} - y_i) = y_{i+2} - 2y_{i+1} + y_i; \\ \Delta^3 y_i &= \Delta^2 y_{i+1} - \Delta^2 y_i = (y_{i+3} - 2y_{i+2} + y_{i+1}) - (y_{i+2} - 2y_{i+1} + y_i) = \\ &= y_{i+3} - 3y_{i+2} + 3y_{i+1} - y_i. \end{aligned} \quad (6)$$

Неважко довести, що для будь-якого цілого значення  $m$  мають місце формули:

$$\begin{aligned} \Delta^m y_i &= y_{i+m} - m y_{i+m-1} + \frac{m(m-1)}{2!} y_{i+m-2} - \\ &- \frac{m(m-1)(m-2)}{3!} y_{i+m-3} + \dots + (-1)^{m-1} m y_{i+1} + (-1)^m y_i. \end{aligned} \quad (7)$$

Скінченні різниці різних порядків зручно розташовувати у вигляді: горизонтальної (табл. 1) або діагональної таблиць (табл. 2).

Таблиця 1 – Горизонтальна таблиця скінченних різниць функції

$x$	$y$	$\Delta y$	$\Delta^2 y$	$\Delta^3 y$
$x_0$	$y_0$	$\Delta y_0$	$\Delta^2 y_0$	$\Delta^3 y_0$
$x_1$	$y_1$	$\Delta y_1$	$\Delta^2 y_1$	
$x_2$	$y_2$	$\Delta y_2$		
$x_3$	$y_3$			

Таблиця 2 – Діагональна таблиця скінченних різниць функції

$x$	$y$	$\Delta y$	$\Delta^2 y$	$\Delta^3 y$
$x_0$	$y_0$			
		$\Delta y_0$		
$x_1$	$y_1$		$\Delta^2 y_0$	
		$\Delta y_1$		$\Delta^3 y_0$
$x_2$	$y_2$		$\Delta^2 y_1$	
		$\Delta y_2$		
$x_3$	$y_3$			

Перша інтерполяційна формула Ньютона для рівновіддалених вузлів інтерполяції має такий вигляд:

$$\begin{aligned} P_n(x) &= y_0 + \frac{\Delta y_0}{h}(x - x_0) + \frac{\Delta^2 y_0}{2! h^2}(x - x_0)(x - x_1) + \\ &+ \frac{\Delta^3 y_0}{3! h^3}(x - x_0)(x - x_1)(x - x_2) + \dots + \frac{\Delta^n y_0}{n! h^n}(x - x_0)(x - x_1) \dots (x - x_{n-1}). \end{aligned} \quad (8)$$

Її можна записати у вигляді більш зручному для практичного використання. Позначимо:

$$\frac{x - x_0}{h} = q. \quad (9)$$

Тоді

$$\frac{(x - x_1)}{h} = \frac{x - (x_0 + h)}{h} = q - 1; \quad (10)$$

$$\text{де } \frac{x-x_2}{h} = q-2; \quad \dots, \frac{x-x_{n-1}}{h} = q-n+1, \quad (11)$$

і формула (8) набуває вигляду:

$$P_n(x) = y_0 + q\Delta y_0 + \frac{q(q-1)}{2!} \Delta^2 y_0 + \frac{q(q-1)(q-2)}{3!} \Delta^3 y_0 + \dots + \frac{q(q-1)(q-2)\dots(q-n+1)}{n!} \Delta^n y_0. \quad (12)$$

Формулу (12) доцільно використовувати для інтерполювання (екстраполювання) функції  $y = f(x)$  в околі початкового значення  $x_0$ , де значення  $q$  є малим за абсолютною величиною.

Друга інтерполяційна формула Ньютона для рівновіддалених вузлів інтерполяції має вигляд:

$$P_n(x) = y_n + \frac{\Delta y_{n-1}}{h} (x-x_n) + \frac{\Delta^2 y_{n-2}}{2! h^2} (x-x_n)(x-x_{n-1}) + \frac{\Delta^3 y_{n-3}}{3! h^3} (x-x_n)(x-x_{n-1})(x-x_{n-2}) + \dots + \frac{\Delta^n y_0}{n! h^n} (x-x_n)(x-x_{n-1})\dots(x-x_1). \quad (13)$$

Запишемо її у вигляді, зручнішому для практичного використання. Позначивши

$$\frac{x-x_n}{h} = q, \text{ дістанемо:}$$

$$\begin{aligned} \frac{x-x_{n-1}}{h} &= \frac{x-(x_n-h)}{h} = q+1; \\ \frac{x-x_{n-2}}{h} &= \frac{x-(x_n-2h)}{h} = q+2; \end{aligned} \quad (14)$$

$$\dots \dots \dots \frac{x-x_1}{h} = \frac{x-[x_n-(n-1)h]}{h} = q+n-1.$$

Після підстановок цих значень у формулу (13) вона набуває вигляду

$$P_n(x) = y_n + q\Delta y_{n-1} + \frac{q(q+1)}{2!} \Delta^2 y_{n-2} + \frac{q(q+1)(q+2)}{3!} \Delta^3 y_{n-3} + \dots + \frac{q(q+1)\dots(q+n-1)}{n!} \Delta^n y_0. \quad (15)$$

Цю формулу рекомендується використовувати для інтерполювання (екстраполювання) функції  $y = f(x)$  для тих значень аргумента, які знаходяться у кінці таблиці.

Нарешті, нехай невідому функцію  $y = f(x)$  задано своїми числовими значеннями  $y_0 = f(x_0), y_1 = f(x_1), \dots, y_n = f(x_n)$  у нерівновіддалених вузлах інтерполяції  $x_0, x_1, x_2, \dots, x_n$ . Тоді задача інтерполяції функції може бути розв'язана за допомогою інтерполяційного полінома Лагранжа:

$$L_n(x) = \sum_{k=0}^n \frac{(x-x_0)(x-x_1)(x-x_2)\dots\Lambda(x-x_{k-1})(x-x_{k+1})\dots\Lambda(x-x_{n-1})(x-x_n)}{(x_k-x_0)(x_k-x_1)(x_k-x_2)\dots\Lambda(x_k-x_{k-1})(x_k-x_{k+1})\dots\Lambda(x_k-x_{n-1})(x_k-x_n)} \cdot y_k \quad (16)$$

Зупинимось на оцінках похибок інтерполяційних формул Лагранжа і Ньютона. Адже недостатньо лише обчислити значення функції у довільній точці, потрібно також вміти оцінити похибку знайденого значення.

Про похибку, що виникає у разі заміни функції  $y = f(x)$  її інтерполяційним многочленом  $P_n(x)$ , можна судити за величиною залишкового члена:

$$R_n(x) = f(x) - P_n(x). \quad (7)$$

Якщо для функції  $y = f(x)$  відомий аналітичний вираз і можна знайти її похідні до  $(n+1)$ -го порядку включно у деякій області  $a \leq x \leq b$ , що містить вузли

інтерполяції  $x_0, x_1, x_2, \dots, x_n$ , то величину залишкового члена  $R_n(x)$  для інтерполяційної формули Лагранжа (3.16) визначають у такий спосіб:

$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} (x-x_0)(x-x_1)\dots(x-x_n), \quad (18)$$

де змінна  $\xi$  лежить усередині відрізка  $[a, b]$ . Позначивши через

$$M_{n+1} = \max_{a \leq x \leq b} |f^{(n+1)}(x)|, \quad (19)$$

отримаємо оцінку для абсолютної похибки інтерполяційної формули Лагранжа:

$$|R_n(x)| \leq \frac{M_{n+1}}{(n+1)!} |(x-x_0)(x-x_1)\dots(x-x_n)|. \quad (20)$$

Якщо вузли інтерполяції  $x_0, x_1, x_2, \dots, x_n$  рівновіддалені, то, вважаючи  $q = \frac{x-x_0}{h}$ ,

отримаємо залишковий член першої інтерполяційної формули Ньютона:

$$R_n(x) = h^{n+1} \frac{q(q-1)(q-2)\dots(q-n)}{(n+1)!} f^{(n+1)}(\xi), \quad (21)$$

де  $\xi$  — деяке проміжне значення між вузлами інтерполяції. Аналогічно, вважаючи

$q = \frac{x-x_n}{h}$ , отримаємо залишковий член другої інтерполяційної формули Ньютона:

$$R_n(x) = h^{n+1} \frac{q(q+1)(q+2)\dots(q+n)}{(n+1)!} f^{(n+1)}(\xi). \quad (22)$$

У практичних розрахунках аналітичний вигляд функції  $y = f(x)$  не завжди відомий. Тоді, припускаючи, що в таблиці скінченних різниць для функції  $y = f(x)$  різниці  $(n+1)$ -го порядку  $\Delta^{n+1}y$  у майже постійні і  $h$  досить мале, а також ураховуючи, що

$$f^{(n+1)}(x) = \lim_{h \rightarrow 0} \frac{\Delta^{n+1}y}{h^{n+1}}, \quad (23)$$

наближено можна вважати:

$$f^{(n+1)}(\xi) \approx \frac{\Delta^{n+1}y_0}{h^{n+1}}. \quad (24)$$

У цьому разі залишковий член першої інтерполяційної формули Ньютона дорівнює:

$$R_n(x) \approx \frac{q(q-1)\dots(q-n)}{(n+1)!} \Delta^{n+1}y_0, \quad (25)$$

а для другої інтерполяційної формули Ньютона залишковий член дорівнює:

$$R_n(x) \approx \frac{q(q+1)\dots(q+n)}{(n+1)!} \Delta^{n+1}y_n. \quad (26)$$

Для реалізації мети дослідження розроблено C++Builder проект, який складається з головної форми Form1, програмний код якої знаходиться у файлі MainFormUnit1, та семи підлеглих форм, призначення яких показано у таблиці 3. На головній формі (рис. 1) розміщене головне меню проекту, яке забезпечує зв'язок з усіма іншими формами. Головне меню включає в себе шість команд, за допомогою яких можна перейти на всі підлеглі форми, кожна з яких вирішує ту чи іншу підзадачу проекту. Наприклад, вибравши першу команду головного меню, ми перейдемо на форму Form2, на якій розв'язується задача інтерполяції функції за допомогою першої інтерполяційної формули Ньютона.

Для створення головного меню проекту використано компонент MainMenu, кожна команда якого (пункт меню) забезпечує перехід на відповідну підлеглу форму, а з кожної підлеглої форми можна повернутись лише на головну.

Зв'язування головної та всіх підлеглих форм досягається тим, що у файлі головної форми MainFormUnit1.cpp включені директиви з іменами файлів усіх підлеглих форм:

```
#include "MainFormUnit1.h"
#include "TabNewtOneUnit2.h"
#include "TabNewtTwoUnit3.h"
#include "TabLagrUnit4.h"
#include "GrafNewt1Unit5.h"
#include "GrafNewt2Unit6.h"
#include "GrafLagrUnit7.h"
#include "ProectContUnit8.h",
```

а у заголовки підлеглих форм включаємо директиву звернення до головної форми. Наприклад, для форми Form2 це виглядає так:

```
#include "TabNewtOneUnit2.h"
#include "MainFormUnit1.h"
```

Звичайно, перехід з головної форми на одну з підлеглих забезпечується програмуванням відповідних елементів управління. Так, перша команда головного меню, яка забезпечує перехід на форму Form2, запрограмована так:

```
void __fastcall TForm1::N11Click(TObject *Sender)
{Form2->Show();}
```

Відповідно до цього коду, клацнувши по першій команді головного меню, ми перейдемо до форми Form2. Команда До головного меню форми Form2, про яку йтиметься далі, має такий програмний код:

```
void __fastcall TForm2::N3Click(TObject *Sender)
{Form2->Close();
Form1->Show();}
```

Виконавши команду **До головного меню** на формі Form2, ми закриємо форму Form2 і перейдемо до головної форми.

Таблиця 3. Найменування та призначення форм і відповідних їм файлів.

з/п	Форма	Найменування файла	Призначення форми
	2	3	4
	Form1	MainFormUnit1	Головна форма проекту
	Form2	TabNewtOneUnit2	Інтерполяція функції за першою інтерполяційною формулою Ньютона
	Form3	TabNewtTwoUnit3	Інтерполяція функції за другою інтерполяційною формулою Ньютона
	Form4	TabLagrUnit4	Інтерполяція функції за інтерполяційною формулою Лагранжа
	Form5	GrafNewt1Unit5	Графік функції за першою інтерполяційною формулою Ньютона
	Form6	GrafNewt2Unit6	Графік функції за другою інтерполяційною формулою Ньютона
	Form7	GrafLagrUnit7	Графік функції за інтерполяційною формулою Лагранжа
	Form8	ProectContUnit8	Зміст задач проекту

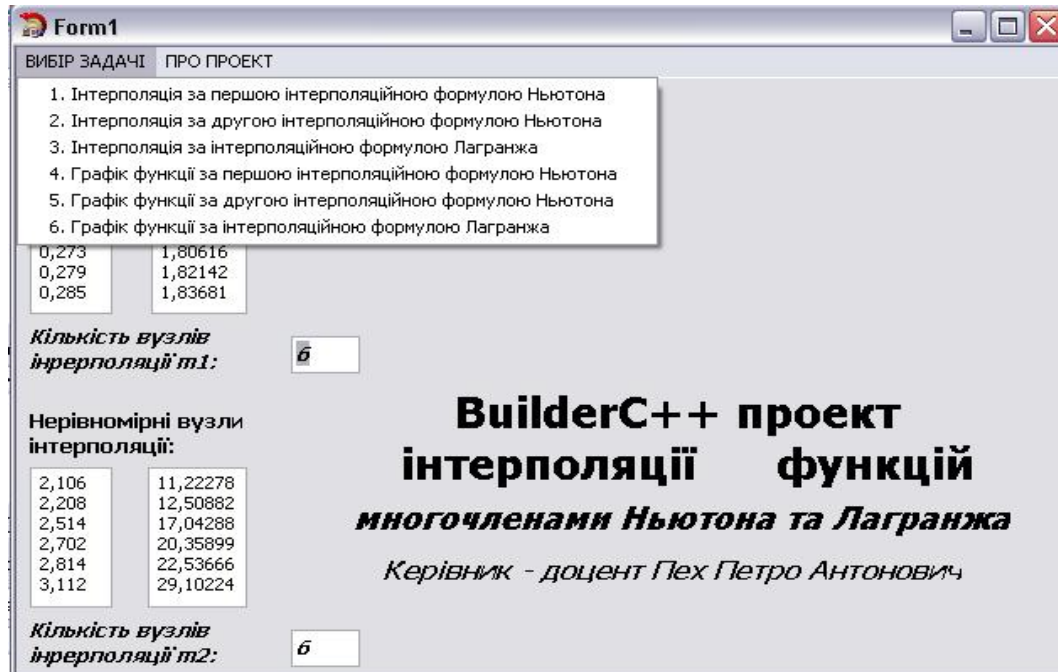


Рис. 1 – Вікно Form1 - головної форми проекту

На рис.2 наведено форму Form2, з результатами обчислень кінцевих різниць та коефіцієнтів інтерполяційного многочлена, а на рис.3 наведено ту ж форму Form2, з виведенням на неї інтерполяційним многочленом, знайденим за першою інтерполяційною формулою Ньютона.

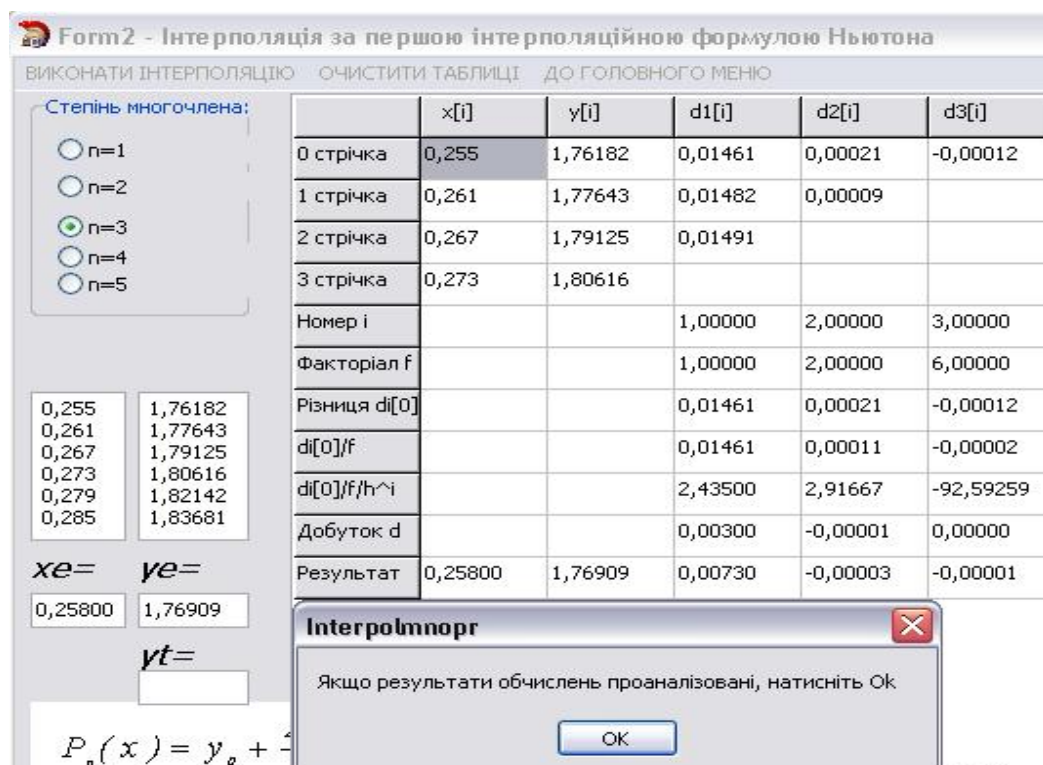


Рис. 2 – Вигляд вікна Form2 з таблицею проміжних та кінцевих результатів



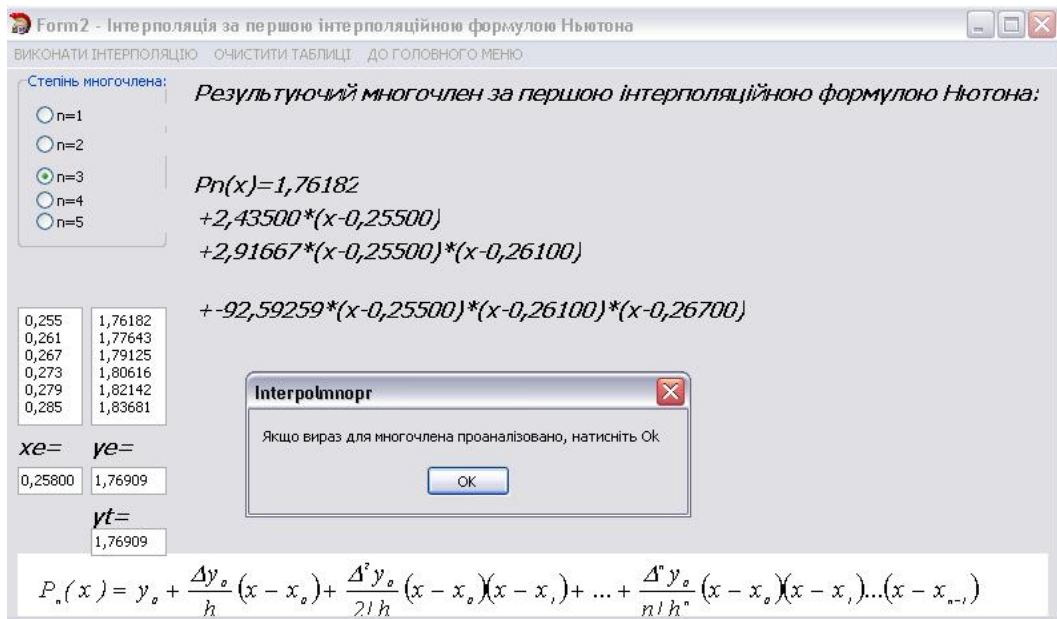


Рис. 3 – Вигляд вікна Form2 з першою інтерполяційною формулою Ньютона

Такі ж результати отримані стосовно другої інтерполяційної формули Ньютона та інтерполяційного многочлена Лагранжа. Передбачена також можливість побудови графіків за кожним зі знайдених многочленів.

#### Висновки

У роботі розроблено C++Builder проект, за допомогою якого засобами сучасних інформаційних технологій розв'язується актуальна задача автоматизації процесу побудови інтерполяційних многочленів за формулами Ньютона та Лагранжа.

1. Архангельский А.Я. Программирование в C++Builder. –М.:ООО «Бином-Пресс», 2010.-1298 с.
2. Дейтел Х.М., Дейтел П.Дж. Как программировать на С. – М.:Издательство Бином, 2006.-912 с.
3. Шпак З.Я. Програмування мовою С. – Львів: Оріяна-Нова, 2006. -432 с.
4. Хвищун І.О. Програмування та математичне моделювання. –К.:Видавничий Дім «Ін Юре», 2007. -544 с.



УДК 004.415.3

Пех П.А. к.т.н. доц., Бондарчук В.О.

Луцький національний технічний університет

## ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ СИНХРОННИХ ТА АСИНХРОННИХ ТЕХНОЛОГІЙ ПРОГРАМУВАННЯ У ПРОЦЕСІ РОЗРОБКИ ВЕБ ДОДАТКІВ

**Пех П.А., Бондарчук В.О. Дослідження ефективності синхронних та асинхронних технологій програмування у процесі розробки веб додатків.** В даній статті відображено концепт взаємодії серверних додатків із CMS системами на прикладі асинхронної технології Node.js та CMS Drupal.

**Ключові слова:** Drupal, Node.js, Open source, API, асинхронність, Javascript, PHP, real-time apps.

**Пех П.А., Бондарчук В.А. Исследование эффективности синхронных и асинхронных технологий программирования в процессе разработки веб приложений.** В данной статье отражен концепт взаимодействия серверных приложений с CMS системами на примере асинхронной технологии Node.js и CMS Drupal.

**Ключевые слова:** Drupal, Node.js, Open source, API, асинхронность, Javascript, PHP, real-time apps.

**Pekh P., Bondarchuk V. The efficacy of sync and async programming technologies in the web applications development.** This article shows the concept of integration server applications with CMS systems on example async technology Node.js and CMS Drupal.

**Keywords:** Drupal, Node.js, Open source, API, async, Javascript, PHP, real-time apps.

**Вступ.** Drupal є прекрасною платформою, але він не може зробити все. Зі зростанням об'ємів навантажень на сайт, зростають і випадки коли движок (Drupal) не справляється. На щастя, всі ці проблеми, пов'язані з оптимізацією і зменшенням навантаження на систему, можна розв'язати за допомогою сторонніх сервісів, такі як Node.js.

Node.js - це платформа з відкритим вихідним кодом для написання серверних і мережеских додатків. В її основі лежить подієво-орієнтоване та асинхронне програмування з неблокуючим введенням/виведенням. Платформа призначена для відокремленого виконання веб-додатків на мові JavaScript. Для виконання JavaScript-коду використовується рушій V8, розроблений компанією Google. Інтеграцію Node.js з Drupal 7 можна провести за допомогою модуля Node.js integration.

Все більш поширеними засобами для оптимізації сайтів стають сторонні API. В свою чергу, API (Application Programming Interface) - це зазвичай (але не обов'язково) метод абстракції між низькорівневим та високорівневим програмним забезпеченням. Одним з найпоширеніших призначень API є надання набору широко використовуваних функцій, наприклад для малювання вікна чи іконок на екрані. Програмісти використовують переваги API у функціональності, таким чином їм не доводиться розробляти все з нуля. API є абстрактним поняттям — програмне забезпечення, що пропонує деякий API, часто називають реалізацією (implementation) даного API. У багатьох випадках API є частиною набору розробки програмного забезпечення, водночас, набір розробки може включати як API, так і інші інструменти/апаратне забезпечення, отже ці два терміни не є взаємозамінювані.

Інтеграція з сторонніми сервісами являє собою досить гнучке рішення, яке дозволяє робити сайти більш потужними без використання купи нового функціоналу всередині веб додатка. Головна проблема – це те, більшість сторонніх API сервісів дуже повільні.

Node.js призначений для відокремленого виконання високопродуктивних мережеских застосунків на мові JavaScript. Функції платформи не обмежені створенням серверних скриптів для веб, платформа може використовуватися і для створення звичайних клієнтських і серверних мережеских програм. Для забезпечення виконання JavaScript-коду використовується розроблений компанією Google рушій V8. Для забезпечення обробки великої кількості паралельних запитів Node.js задіює асинхронну модель запуску коду, засновану на обробці подій в неблокуючому режимі і визначенні обробників зворотніх викликів (callback). В якості способів мультиплексування з'єднань підтримується epoll, kqueue, /dev/poll і select. Для мультиплексування з'єднань використовується бібліотека libev, для створення пулу нитей (thread pool) задіяна бібліотека libeio, для виконання DNS-запитів у неблокуючому режимі інтегрований c-ares. Всі системні виклики, що викликають блокування, виконуються всередині пула нитей і потім, як і обробники сигналів, передають результат своєї роботи назад через неіменовані канали (pipe).

За своєю суттю Node.js схожий на фреймворки Perl AnyEvent, Ruby Event Machine і Python Twisted, але цикл обробки подій (event loop) у Node.js прихований від розробника і нагадує обробку подій у веб-застосунку, що працює в браузері. При написанні програм для Node.js необхідно враховувати специфіку подієво-орієнтованого програмування, наприклад, замість виконання.

Node.js це асинхронна мова програмування, іншими словами, поки в одному потоці виконується db\_select(), програма починає виконання іншого шматка коду в іншому потоці, наприклад db\_query. а після опрацювання обох потоків і отримання response (відповіді), повертає дані, які використовуються у програмі далі.

Це єдиний процес, але він багатопоточний - у порівнянні з PHP, який створює процес для кожного користувача, в Node.js один процес може виконувати паралельні дії для кожного користувача. Це набагато швидше, тому що сервери не повинні витрачати ресурси і час для створення процесів для кожного запита (request), витягати якісь дані з бази даних чи щось подібне. Потрібно просто запустити сервер, завантажити всі дані і чекати на запит від користувача.

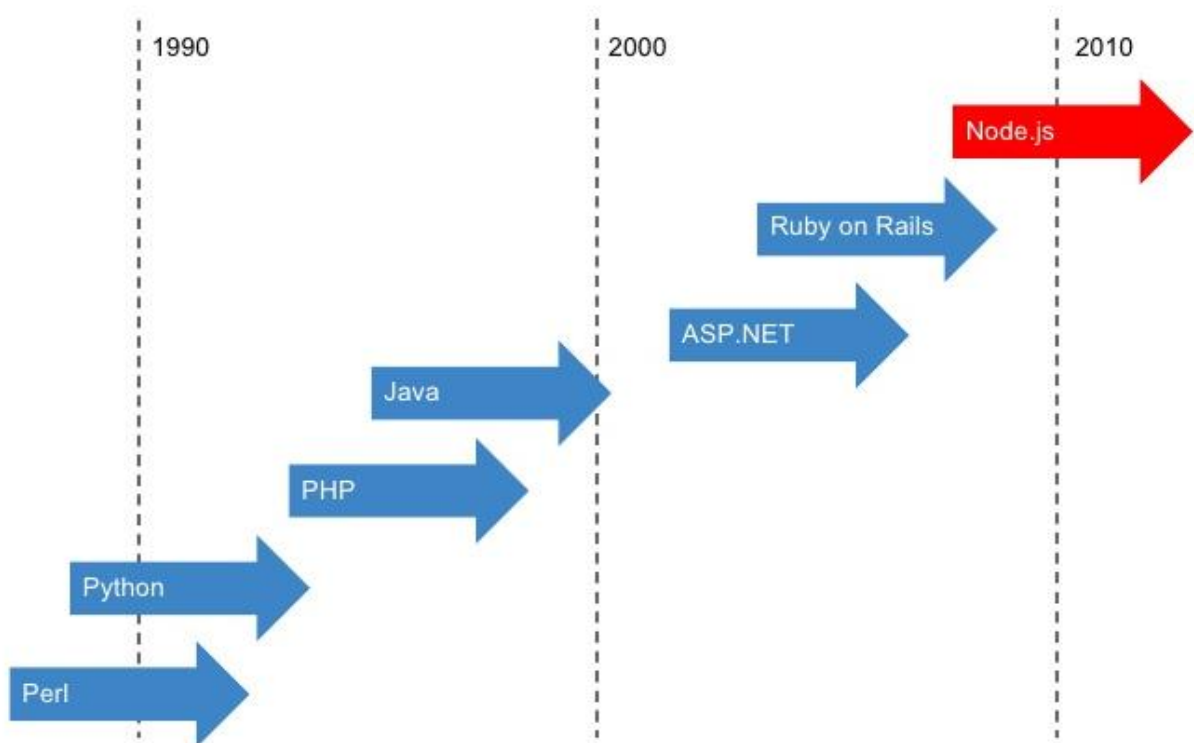


Рис. 1 – Хронологія появи деяких мов програмування та фреймворків.

**Основна частина.** Для налаштування Node.js в Drupal потрібно:

- 1) Встановити всі необхідні пакети для роботи з Node.js та сам Node.js  
`$ sudo apt-get install build-essential git curl openssl libssl-dev`
- 2) Компіляція його вихідних кодів  
`sudo make install`
- 3) Якщо все пройшло без помилок, то вам відкриється доступ до оболонки node.js.  
`$ node`  
`> console.log('Hello world');`  
 Результатом має бути:  
`Hello world`
- 4) Наступним нашим кроком буде встановлення модуля node.js на Drupal  
`drush en nodejs`
- 5) Встановлення всіх необхідних залежностей  
`$ sudo npm install socket.io`

```
$ sudo npm install request
$ sudo npm install express
$ sudo npm install connect
```

- б) Далі необхідно зайти на сторінку налаштування node.js, скопіювати вміст поля “Suggested configuration” і створити файл sites/all/modules/nodejs/nodejs.config.js та додати туди скопійовані налаштування.

**Node.js server configuration builder**

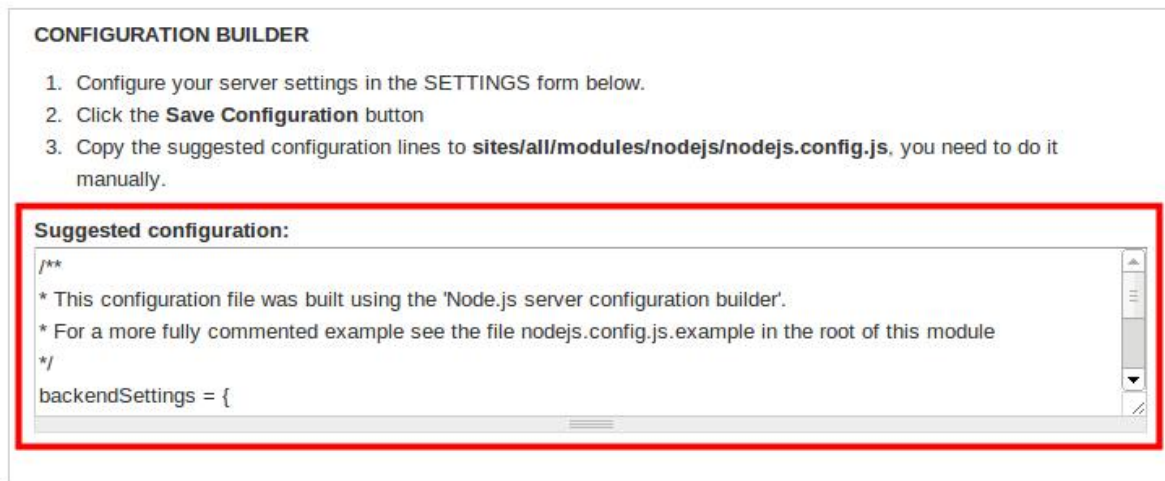


Рис. 2 – Видгляд вікна конфігурації Node.js Integration module у Drupal.

Все, платформу node.js налаштовано. Перевірити її роботу можна, включивши модуль “Nodejs Watchdog” (він входить в пакет модуля Node.js integration).

Для того, щоб постійно не прописувати ‘node server.js’ можна його запустити в фоновому режимі за допомогою пакета Forever. Forever вміє керувати фоновими процесами, а саме: перезапускати їх після збою; перенаправляти стандартний вивід помилок у файли журналів, та виконувати інші корисні функції. Отже, для того, щоб встановити його на сервері потрібно:

```
$ sudo npm install -g forever
```

Тепер замість 'node server.js' в консолі можна виконувати 'forever start server.js'. Для того, щоб зупинити - прописати 'forever stop server.js'. Щоб переглянути, які процеси запущено через forever, можна скористатись 'forever list'.

Далі буде продемонстровано приклад відображення “Hello World!” за допомогою Node.js без участі Drupal CMS.

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(8000);
```

- Переваги Node.js над прост Javascript:
- 1) не вибагливий в ресурсах;
  - 2) немає стандартних особливостей веб сервера;
  - 3) не потрібна аутентифікація;
  - 4) не потребує сесій для користувачів;
  - 5) не потребує cookies;
  - 6) не потребує емейлів;
  - 7) не потребує шаблонів (templates);
  - 8) не містить в собі MVC чи інший інструкцій фреймворка;

## Node.js і Drupal

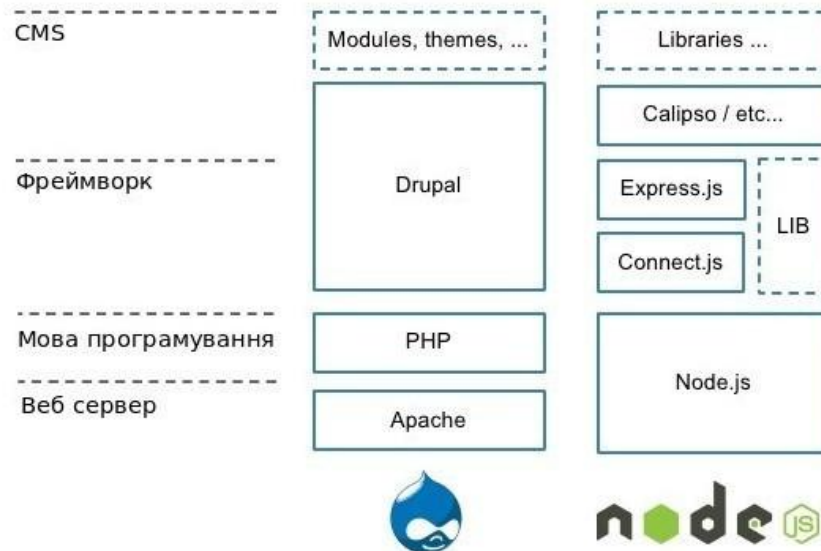


Рис. 3 – Схема інтеграції і взаємодії Drupal з Node.js.

У наступному прикладі буде продемонстровано приклад виведення тексту “Hello World” методами Drupal через взаємодію Node.js API.

```
$message = (object) array(  
  'broadcast' => TRUE,  
  'data' => (object) array(  
    'subject' => 'buu',  
    'body' => 'Hellow World!',  
  ),  
);  
nodejs_enqueue_message($message);
```

Більш складний приклад можна проглянути в модулі “Nodejs Watchdog”, який входить в пакет модуля Node.js Integration.

### Висновки

У роботі досліджено перспективність і аспекти використання асинхронних та синхронних технологій на прикладі серверних скриптів, побудованих за допомогою технології Node.js та показано приклад інтегрування Node.js API з CMS Drupal.

1. Кантелон М.. Node.js in Action. –М.:«Manning Publications Co», 2014.-417 с.
2. Херрон Д. Node.js Разработка серверных веб-приложений на JavaScript. – М.:Издательство ДМК Пресс, 2012.-946 с.
3. Пауерс Ш. Изучаем Node.js С. – Питер: Орияна-Нова, 2014. -400 с.

УДК 004.415.3

Пех П.А.к.т.н. доц., Христинець Н.А., Прокопчук О.П.  
Луцький національний технічний університет

## **C++BUILDER ПРОЕКТ АНАЛІЗУ ЕКСПЕРИМЕНТАЛЬНИХ ДАНИХ ЗА СИМЕТРИЧНИМ КОМПОЗИЦІЙНИМ РОТАТАБЕЛЬНИМ УНІФОРМ ПЛАНОМ ДРУГОГО ПОРЯДКУ**

**Пех П.А., Христинець Н.А., Прокопчук О.П.** C++Builder проект аналізу експериментальних даних, отриманих за симетричним композиційним ротатабельним уніформ планом другого порядку. В статті запропоновано C++Builder проект для побудови регресійної моделі другого порядку шляхом оброблення експериментальних даних, отриманих за симетричним композиційним ротатабельним уніформ планом.

Ключові слова: C++Builder проект, теорія планування експерименту, симетричний композиційний ротабельний уніформ план.

**Пех П.А., Христинець Н.А., Прокопчук А.П.** C++Builder проект аналізу експериментальних даних, отриманих по симетричному композиційному ротатабельному уніформ плану другого порядку. В статті запропоновано C++Builder проект для побудови регресійної моделі другого порядку шляхом оброблення експериментальних даних, отриманих по симетричному композиційному ротатабельному уніформ плану.

Ключевые слова: C++Builder проект, теория планирования эксперимента, симметричный композиционный ротатабельный уніформ план.

**Pekh Petro, Christinee Natalija, Prokopchuk Aleksandr.** C++Builder project for analysis of experimental data obtained by symmetrical compositional rotatable uniform plan of the second order. In this article is designed the C++Builder project to build a second order regression model by processing experimental data obtained by symmetrical compositional rotatable uniform plan.

Keywords: C + + Builder project, the theory of experiment planning, symmetrical compositional rotatable uniform plan.

**Постановка задачі.** Нехай в процесі проведення експерименту досліджується вплив множини факторів  $x_1, x_2, x_3, L, x_k$  на невідому функцію  $y = f(x_1, x_2, x_3, L, x_k)$ . Метою такого дослідження є отримання математичної моделі у вигляді рівняння регресії певного, наприклад, другого порядку:

$$y = b_0 + \sum_{1 \leq i \leq k} b_i x_i + \sum_{1 \leq i < j \leq k} b_{ij} x_i x_j + \sum_{1 \leq i \leq k} b_{ii} x_i^2 \quad (1)$$

де  $x_1, x_2, x_3, L, x_k$  — кодовані значення факторів, а  $b_0, b_i, b_{ij}, b_{ii}$  - коефіцієнти, значення яких розраховуються за результатами експериментів. Число членів такої моделі можна визначити з формули:

$$C_{k+2}^k = \frac{(k+1)(k+2)}{2}, \quad (2)$$

де  $k$  - число факторів. Для трьохфакторного експерименту, наприклад, число членів моделі дорівнює десяти. Крім того, кожен фактор має варіюватися мінімум на трьох рівнях - найнижчому, найвищому та в центрі експерименту.

Шукана модель повинна адекватно відображати досліджуваний процес або явище, тобто при одних і тих же значеннях факторів експериментальні значення функції не повинні істотно відрізнятися від розрахованих за математичною моделлю. Задача аналізу експериментальних даних передбачає визначення коефіцієнтів моделі, їх дисперсій, коваріацій та довірчих інтервалів, а також перевірку значимості розрахованих коефіцієнтів.

Для побудови математичної моделі вченими запропоновано багато різних планів. Об'єктом даного дослідження є симетричний композиційний ротатабельний уніформ план другого порядку. Симетричним називається такий план, у якому всі дослідні точки проводяться у точках, що розташовані симетрично від центру експерименту. Композиційним називається такий план, який можна отримати шляхом побудови відповідного лінійного плану. Це досягається за рахунок введення в схему плану так званих зіркових точок. Нарешті, ротатабельність плану забезпечує однакову дисперсію функції відгуку у різних напрямках плану. Забезпечується вона реалізацією відповідної кількості дослідів у центрі експерименту. Якщо ж ця дисперсія практично незмінна на інтервалі  $[-1; +1]$ , то план буде одночасно і уніформним. Для подальшого дослідження ми скористалися результатами експерименту [1] за схемою симетричного композиційного ротатабельного

уніформплану другого порядку для випадку  $k = 3$  факторів. Рівні варіювання факторів наведені у таблиці 1

Таблиця 1 – Рівні варіювання факторів

Фактори	Амплітуда, $A(\text{мм})$	Співвідношення мас порошків $C_1/C_2$ $Sp$	час сегрегації, $t (\text{хв})$
Основний рівень ( $X_{i0}$ )	3,76	0,15	4
Інтервали варіювання ( $\Delta X_i$ )	1,18	0,05	2
Верхній рівень ( $x_i = +1$ )	4,56	0,20	6
Нижній рівень ( $x_i = -1$ )	2,2	0,10	2
Зіркова точка $+\alpha (x_i = +1.682)$	6,56	0,24	7,4
Зіркова точка $-\alpha (x_i = -1.682)$	0,2	0,06	0,6

Експеримент реалізовано в області матеріалознавства. Метою проведення експерименту було дослідження залежності величини  $y$  - інтенсивності концентрації порошку  $C_1$  в двокомпонентній порошковій суміші від амплітуди коливання  $x_1$ , співвідношення мас порошків  $C_1/C_2$  -  $x_2$  та часу сегрегації  $x_3$ .

Зв'язок між натуральними та кодованими значеннями факторів задається формулами:

$$x_1 = \frac{X_1 - 3,76}{1,18}; \quad x_2 = \frac{X_2 - 0,15}{0,05}; \quad x_3 = \frac{X_3 - 4}{2}. \quad (3)$$

Метою даного дослідження є розроблення програмного комплексу, який дозволяє автоматизувати процес побудови та аналізу математичної моделі на базі експериментальних даних.

**Основна частина.** Нехай результати експерименту за схемою симетричного композиційного ротатбельного уніформ плану другого порядку наведені у табл. 2. Виходячи з цих результатів, послідовно розраховуємо коефіцієнти моделі, їх дисперсії, коваріації та довірчі інтервали, а також перевіряємо значимість розрахованих коефіцієнтів.

Коефіцієнти моделі розраховуємо за такими формулами:

$$y = b_0 + \sum_{i|s|sk} b_i x_i + \sum_{i|s|<j|sk} b_{ij} x_i x_j + \sum_{i|s|sk} b_{ii} x_i^2 \quad (4)$$

$$b_0 = c_1 \sum_{u=1}^N y_u - c_2 \sum_{i=1}^k \sum_{u=1}^N x_{iu}^2 y_u; \quad b_i = c_3 \sum_{u=1}^N x_{iu} y_u; \quad b_{ij} = c_4 \sum_{u=1}^N (x_i x_j)_u y_u; \quad (5)$$

$$b_{ii} = c_5 \sum_{u=1}^N x_{iu}^2 y_u + c_6 \sum_{i=1}^k \sum_{u=1}^N x_{iu}^2 y_u - c_2 \sum_{u=1}^N y_u \quad (6)$$

Для вирішення всіх вище зазначених завдань стосовно такого роду емпіричних даних нами розроблено C++Builder проект. Він складається з головної та восьми підлеглих форм (табл. 3), кожна з яких вирішує ту чи іншу частину задачі. Кількість форм за потреби може бути збільшена.

Таблиця 2 – Результати експерименту за схемою симетричного композиційного ротатабельного уніформ плану другого порядку

№ до сліду	$x_0$	$x_1$	$x_2$	$x_3$	$x_1x_2$	$x_1x_3$	$x_2x_3$	$x_1x_1$	$x_2x_2$	$x_3x_3$	$y$
1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	30.2
2	+1	-1	+1	+1	-1	-1	+1	+1	+1	+1	20.2
3	+1	+1	-1	+1	-1	+1	-1	+1	+1	+1	7.0
4	+1	-1	-1	+1	+1	-1	-1	+1	+1	+1	14.1
5	+1	+1	+1	-1	+1	-1	-1	+1	+1	+1	43.0
6	+1	-1	+1	-1	-1	+1	-1	+1	+1	+1	5.8
7	+1	+1	-1	-1	-1	-1	+1	+1	+1	+1	27.9
8	+1	-1	-1	-1	+1	+1	+1	+1	+1	+1	8.0
9	+1	+1.682	0	0	0	0	0	+2.83	0	0	36.0
10	+1	-1.682	0	0	0	0	0	-2.83	0	0	12.1
11	+1	0	+1.682	0	0	0	0	0	+2.83	0	25.3
12	+1	0	-1.682	0	0	0	0	0	-2.83	0	10.4
13	+1	0	0	+1.682	0	0	0	0	0	+2.83	18.0
14	+1	0	0	-1.682	0	0	0	0	0	-2.83	20.0
15	+1	0	0	0	0	0	0	0	0	0	29.0
16	+1	0	0	0	0	0	0	0	0	0	28.4
17	+1	0	0	0	0	0	0	0	0	0	28.6
18	+1	0	0	0	0	0	0	0	0	0	28.8
19	+1	0	0	0	0	0	0	0	0	0	28.7
20	+1	0	0	0	0	0	0	0	0	0	30.5

Вигляд головної форми показаний на рис.2. У ній використано компонент MainMenu, кожна команда якого (пункт меню) забезпечує перехід на відповідну підлеглу форму, а з кожної підлеглої форми можна повернутись лише на головну.

Зв'язування головної та всіх підлеглих форм досягається тим, що у файлі головної форми MainFormUnit1.cpp включені директиви з іменами файлів усіх підлеглих форм:

```
#include "MainFormUnit1.h"
#include "TabCoefModUnit2.h"
#include "CoefModUnit3.h"
#include "DispCoefModUnit4.h"
#include "CovCoefModUnit5.h"
#include "DovIntCoefModUnit6.h"
#include "TabTeorZnachUnit7.h"
#include "ZnachCoefUnit8.h"
#include "GraphFuncUnit9.h"
#include "AdecvatModUnit10.h"
#include "ProectContUnit11.h",
```

а у всіх файлах підлеглих формах включена директива з іменем MainFormUnit1.h головної форми. Наприклад, для форми Form2 це виглядає так:

```
#include "TabCoefModUnit2.h"
#include "MainFormUnit1.h"
```

На рис.2 наведено форму Form2, з результатами обчислень матриці планування, а на рис.3 наведено форму Form3, з результатами обчислень коефіцієнтів регресійної моделі.

Таблиця 3. Найменування та призначення форм і відповідних їм файлів.

№ з/п	Форма	Найменування файла	Призначення форми
1	2	3	4
1	Form1	MainFormUnit1	Головна форма проекту
2	Form2	TabCoefModUnit2	Формування матриці планування
3	Form3	CoefModUnit3	Розрахунок коефіцієнтів моделі
4	Form4	DispCoefModUnit4	Розрахунок дисперсій коефіцієнтів моделі
5	Form5	CovCoefModUnit5	Розрахунок коваріацій коефіцієнтів моделі
6	Form6	DovIntCoefModUnit6	Визначення довірчих інтервалів коефіцієнтів моделі
7	Form7	TabTeorZnachUnit7	Розрахунок теоретичних значень функції відгуку
8	Form8	ZnachCoefUnit88	Визначення значимості коефіцієнтів моделі
9	Form9	GraphFuncUnit9	Побудова графіків функції відгуку
10	Form10	AdecvatModUnit100	Визначення адекватності моделі
11	Form11	ProectContUnit11	Зміст задач проекту



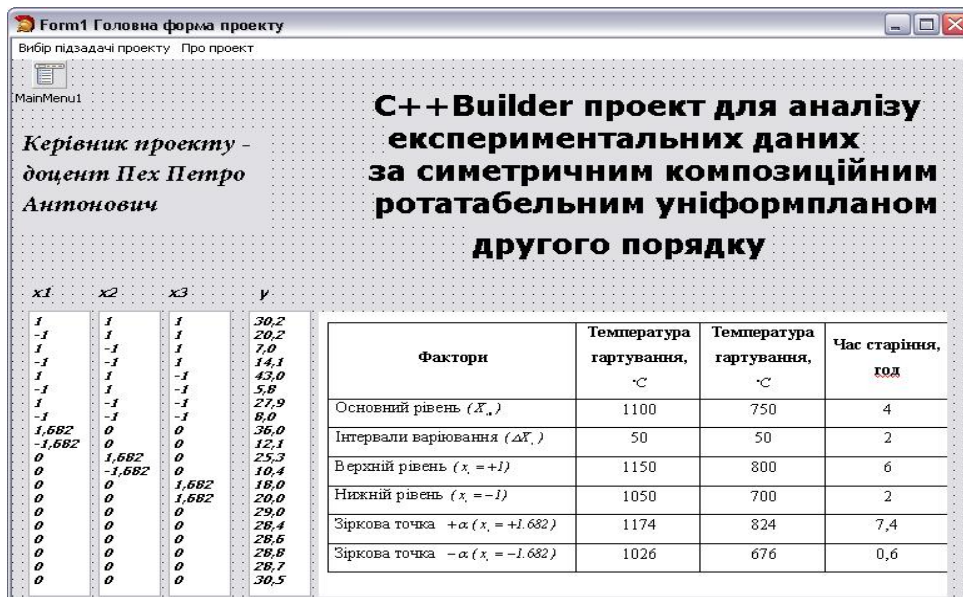


Рис. 1 – Вікно Form1 - головної форми проекту

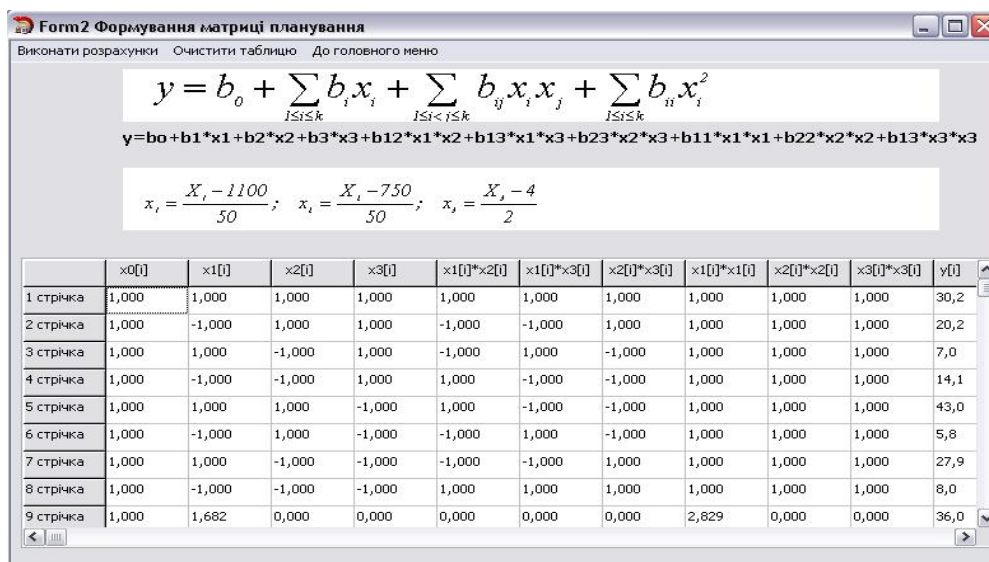


Рис. 2 – Форма Form2 з результатами обчислень матриці планування

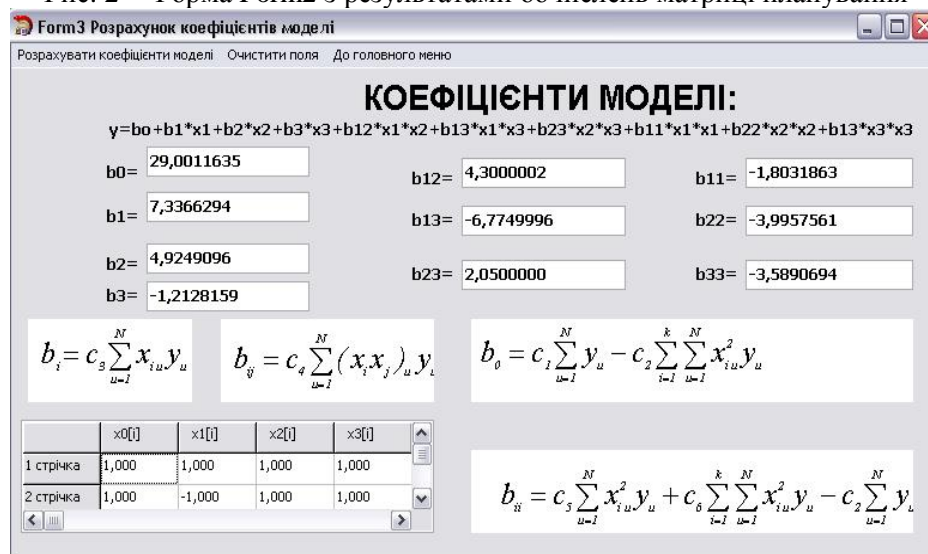


Рис. 3 – Форма Form3 з результатами обчислень коефіцієнтів регресійної моделі

Дисперсії коефіцієнтів моделі розраховуємо за формулами:

$$S_{b_0} = c_7 S_y; \quad S_{b_i} = c_8 S_y; \quad S_{b_{ij}} = c_9 S_y; \quad S_{b_{ii}} = c_{10} S_y, \quad (7)$$

де  $S_y$  - дисперсія досліджуваного. Вона визначається у такій послідовності. Спочатку розраховується середнє значення функції відгуку в центрі експерименту:

$$\bar{y}_0 = \frac{\sum_{g=1}^{n_0} y_{0g}}{n_0} \quad (8)$$

де  $n_0$  - число дослідів у центрі експерименту;  $y_{0g}$  - значення функції відгуку у  $g$  - му досліді в центрі експерименту.

Далі визначаємо величину дисперсії досліджуваного:

$$S_y^2 = \frac{\sum_{g=1}^{n_0} (y_{0g} - \bar{y}_0)^2}{n_0 - 1} \quad (9)$$

Коваріації коефіцієнтів моделі розраховуємо за формулами:

$$\text{cov} b_0 b_{ii} = -c_2 S_y^2 \quad \text{cov} b_{ii} b_{ij} = c_6 S_y^2. \quad (10)$$

Довірчі інтервали коефіцієнтів моделі розраховуємо за формулами:

$$\Delta b_0 = t_{St} S b_0; \quad \Delta b_i = t_{St} S b_i; \quad \Delta b_{ij} = t_{St} S b_{ij}; \quad \Delta b_{ii} = t_{St} S b_{ii} \quad (11)$$

де  $t_{St}$  - табличне значення критерію Стюдента. Табличне значення критерію Стюдента вибирають за таблицею залежно від величини прийнятого рівня значимості, який в інженерних розрахунках приймається рівним 0.05, та числа степенів свободи. У нашому випадку  $t_{St}$ . Цю константу задаємо у розділі опису глобальних змінних форми.

Визначивши довірчі інтервали коефіцієнтів моделі, перевіряємо їх значимість. Якщо модуль коефіцієнта моделі є меншим від модуля відповідного довірчого інтервалу, то нове значення цього коефіцієнта приймається рівним нулю, в протилежному випадку його значення не змінюємо, тобто, коефіцієнт вважається значимим.

Отриману модель слід перевірити на адекватність. Ця перевірка здійснюється за критерієм Фішера. Вважають, що математична модель адекватно відображає досліджуване явище чи процес, якщо розраховане значення критерію Фішера не перевищує табличного значення цього критерію, яке у нашому випадку дорівнює . Значення критерію Фішера розраховуємо за формулою:

$$F_{f_2; f_1}^{\delta f_2 \delta f_1} = \frac{S_{\text{іааа}}^2}{S_y^2}; \quad S_{\text{іааа}}^2 = \frac{\sum_{u=1}^N (y_{ut} - y_u)^2}{f_2}, \quad (12)$$

де  $y_{ut}, y_u$  - теоретичне та експериментальне значення функції в  $u$  -му досліді.

Передбачено також побудову графіків функції відгуку за різних значень вхідних факторів (рис.4).



Рис. 4 – Графіки функції відгуку за різних значень вхідних факторів

### Висновки

У роботі розроблено C++Builder проект, за допомогою якого засобами сучасних інформаційних технологій розв'язується актуальна задача автоматизації процесу побудови та аналізу математичної моделі на базі експериментальних даних, отриманих за схемою симетричного композиційного рота табельного уніформ плану другого порядку.

1. Архангельский А.Я. Программирование в C++Builder. –М.:ООО «Бином-Пресс», 2010.-1298 с.
2. Новик Ф.С., Арсов Я.Б. Оптимизация процессов технологии металлов методами планирования экспериментов. –М.: Машиностроение, София: Техника, 1980.-304 с.

УДК 539.3

Каганюк О.К. к.т.н. доц., Бортник К.Я.к.т.н. доц., Свиридчук В.В.  
Луцький національний технічний університет

## АНАЛІЗ АНОМАЛЬНИХ СТАНІВ ТРАФІКА КОМП'ЮТЕРНОЇ МЕРЕЖІ НА БАЗІ НЕЙРОМЕРЕЖ

**Каганюк О.К., Бортник К.Я., Свиридчук В.В.** Аналіз аномальних станів трафіка комп'ютерної мережі на базі нейромереж. У статті запропонований метод обробки даних перед навчанням нейронної мережі для ідентифікації аномальних станів КС. Задача дослідження аномалій трафіку є однією з найуспішніших застосувань нейронних мереж для рішення задач класифікації.

**Ключові слова:** метод, аномальні стани, нейромережа, трафік, ідентифікація.

**Каганюк О.К., Бортник К.Я., Свиридчук В.В.** Анализ аномальных состояний трафика компьютерной сети на базе нейронных сетей. В статье предложен метод обработки данных перед обучением нейронной сети для идентификации аномальных состояний КС. Задача исследования аномалий трафика является одной из самых успешных применений нейронных сетей для решения задач классификации.

**Ключевые слова:** метод, аномальные состояния, нейросеть, трафик, идентификация.

**Kahanyuk O., Bortnyk K., Sviridyuk V.** Analysis of abnormal states of computer network traffic based on neural networks. The paper proposed a method of processing data before training a neural network to identify abnormal states KS. Zadacha traffic anomalies research is one of the most successful applications of neural networks for solving classification problems.

**Keywords:** method, abnormal conditions, neural network, traffic identification.

**Постановка наукової проблеми.** Виявлення мережевих атак є в даний момент є однією з найбільш гострих проблем мережевих технологій. Однією з актуальних наукових завдань в даний час є аналіз (і подальше прогнозування) самоподібної структури трафіку в сучасних мультисервісних мережах. Для вирішення цього завдання необхідний збір і подальший аналіз різноманітної статистики в діючих мережах. [2,4].

**Метою та завданням** даної роботи є ідентифікація аномальних станів комп'ютерних систем на основі паралельної обробки трафіка КС, використовуючи колектив нейромереж.

**Об'єктом дослідження** є інтелектуальні технології, базовані на нейромережевих принципах, які орієнтовані на розв'язок прикладних задач.

**Предметом дослідження** є методи та моделі ідентифікації аномальних станів трафіка КМ комітетом нейромереж.

**Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.** Більшість сучасних методів виявлення атак використовують деяку форму аналізу контрольованого простору на основі правил або статистичного підходу. В якості контрольованого простору можуть виступати журнали реєстрації або мережевий трафік. Цей аналіз спирається на набір заздалегідь визначених правил, які створюються адміністратором або самою системою виявлення атак.

Будь-який поділ атаки або в часі, або серед кількох злоумисників є важким для виявлення за допомогою експертних систем. За великої різноманітності атак і хакерів навіть спеціальні постійні оновлення бази даних експертної системи ніколи не дадуть гарантій точної ідентифікації всього діапазону атак.

Використання нейронних мереж є одним із способів подолання зазначених проблем експертних систем. На відміну від експертних систем, які можуть дати користувачеві певну відповідь, відповідають чи ні аналізовані характеристики закладеним в базу даних правилам, нейронна мережа проводить аналіз інформації та надає можливість оцінити, чи узгоджуються дані з характеристиками, які вона навчена розпізнавати. У той час як ступінь відповідності нейромережевого подання може досягати 100 %, достовірність вибору повністю залежить від якості системи в аналізі прикладів наданої задачі.

Спочатку нейромережу навчають правильної ідентифікації та попередньо підібраною вибіркою прикладів предметної області. Реакція нейромережі аналізується, і система налаштовується таким чином, щоб досягти задовільних результатів. На додаток до початкового періоду навчання нейромережа набирається також досвіду з протягом часу, у міру того, як вона проводить аналіз даних, пов'язаних з предметною областю.

Важливою перевагою нейронних мереж при виявленні зловживань є їх здатність «вивчати» характеристики умисних атак та ідентифікувати елементи, які не схожі на ті, що спостерігалися в мережі раніше.

Мережі з прямим зв'язком є універсальним засобом апроксимації функцій, що дозволяє їх використовувати у вирішенні задач класифікації. Як правило, нейронні мережі виявляються найбільш ефективним способом класифікації, тому що генерують фактично велике число регресійних моделей (які використовуються у вирішенні задач класифікації статистичними методами).

Багаті можливості відображення особливо важливі в тих випадках, коли на основі кількох оцінок будується високорівнева процедура прийняття рішень. Відомо багато додатків нейронних мереж з прямим зв'язком до завдань класифікації. Як правило, вони виявляються ефективніше інших методів, тому що нейронна мережа генерує нескінченне число нелінійних регресійних моделей.

На жаль, у застосуванні нейронних мереж у практичних завданнях виникає ряд проблем. По-перше, заздалегідь не відомо, якої складності (розміру) може знадобитися мережа для досить точної реалізації відображення. Ця складність може виявитися надмірно високою, що потребує складної архітектури мереж. Так доведено, що найпростіші одношарові нейронні мережі здатні вирішувати тільки лінійно роздільні завдання. Це обмеження можна подолати при використанні багатошарових нейронних мереж. У загальному вигляді можна сказати, що в мережі з одним прихованим шаром, вектор, відповідний вхідному зразку, перетворюється в прихований шар в деякий новий простір, який може мати іншу розмірність, а потім гіперплощини, відповідні нейронам вихідного шару, поділяють його на класи. Таким чином мережа розпізнає не тільки характеристики вихідних даних, але і характеристики характеристик, сформовані прихованим шаром.

Все це підкреслює важливість етапу попередньої обробки даних. Чим більш компактно представлені характеристики зразків, тим менше залежність від настроюваних параметрів мережі (0 або 1).

Підвищення якості навчання нейронної мережі можливе при використанні ефективних методів попередньої обробки даних. Для обробки даних перед навчанням нейронної мережі пропонується використовувати метод, заснований на понятті профілю компактності і комбінаторних формулах для ефективного обчислення функціонала ковзкого контролю. Метод застосовується для підготовки даних в задачах класифікації [3]. Щоб звести своє завдання попередньої обробки даних у моделюванні до задачі попередньої обробки даних при класифікації даних (яка має рішення, що використовує профіль компактності), пропонується провести кластерний аналіз на вихідних параметрах нейронної мережі. Таким чином, будемо мати задачу класифікації, для якої відоме рішення попередньої обробки даних.

Метод, починаючи з повної вибірки, послідовно виключає об'єкти. На кожному кроці вибирається той об'єкт, виключення якого мінімізує функціонал. Виявляється, що процес відсіву об'єктів розбивається на дві стадії. Спочатку виключаються шумові, потім виключаються неінформативні периферійні об'єкти.

Процес зупиняється, коли залишаються об'єкти, виключення яких помітно збільшує функціонал, тоді в масиві даних залишаються опорні об'єкти.

Основним результатом застосування комбінаторної формули для оцінки функціоналу повного ковзкого контролю є те, що вона однаково добре підходить як для виключення шумових об'єктів, так і для скорочення множини прецедентів, будучи при цьому ефективно обчислюваним, точним значенням функціоналу.

Метод спирається на припущення, яке називається гіпотезою компактності: схожі об'єкти набагато частіше лежать в одному класі, ніж в різних. У цьому випадку межа між класами має досить просту форму, а класи утворюють компактно локалізовані області в просторі об'єктів (у математичному аналізі компактними називаються обмежені замкнуті множини, гіпотеза компактності не має нічого спільного з цим поняттям).

Як правило, об'єкти навчання не є рівноцінними. Серед них можуть знаходитися типові представники класів - еталони. Якщо класифікується об'єкт близький до ідеалу, то, швидше за все, він належить тому ж класу. Ще одна категорія об'єктів - неінформативні, або периферійні. Вони щільно оточені іншими об'єктами того ж класу. Якщо їх видалити з вибірки, це практично не позначиться на якості навчання. Нарешті, у вибірку може потрапити деяка кількість шумових викидів - об'єктів, що знаходяться в чужому класі. Зазвичай їх видалення тільки покращує якість класифікації.

Виключення з вибірки шумових і неінформативних об'єктів дає кілька переваг одночасно: підвищується якість класифікації, скорочується обсяг збережених даних і зменшується час класифікації, що витрачається на пошук найближчих еталонів [5].  
 Перейдемо до розгляду функціоналу вибірки, що мінімізується. Нехай  $X$  є множина об'єктів і  $Y$  множина імен класів. Задана навчальна вибірка пар «об'єкт-відповідь»:

$$x^{im} = \{(x_1, y_1), \dots, (x_m, y_m)\} \in x \times y$$

Нехай на множині об'єктів задана функція відстані  $p(x, x')$ . Ця функція повинна бути досить адекватною моделлю подібності об'єктів. Чим менше значення цієї функції, тим більше схожі об'єкти  $x, x'$ .

Для довільного об'єкта  $u$  розташуємо об'єкти навчальної вибірки  $x_i$  в порядку зростання відстаней до  $u$ :

$$p(u, x_{1u}) \leq p(u, x_{2u}) \leq \dots \leq p(u, x_{mu}),$$

де через  $x_{iu}$  позначається елемент навчальної вибірки, який є  $i$ -м сусідом об'єкта  $u$ . Аналогічне позначення введемо і для відповіді на  $i$ -му сусіді –  $y_{iu}$ .

Кожен об'єкт  $u \in X$  породжує свою перенумерацію вибірки.

Розглядається метод найближчого сусіда, який відносить об'єкт  $u$ , що класифікується, до того класу, якому належить найближчий до  $u$  об'єкт навчальної вибірки:  $a(u, x^m) = y_{1u}$ .

Профіль компактності вибірки  $X^m$  є функція:

$$R(j, x^m) = \frac{1}{m} \sum_{i=1}^m [y_i \neq y_{ix}].$$

Іншими словами, профіль компактності  $R(j)$  - це частка об'єктів вибірки, для яких  $j$ -й сусід лежить в іншому класі.

Профіль компактності є формальним виразом гіпотези компактності - припущення про те, що схожі об'єкти набагато частіше лежать в одному класі, ніж в різних. Вибірка  $X^l$  розбивається всілякими способами на дві непересічні підвибірки:

$$x^l = x_n^m \cup x_n^k,$$

де  $x_n^m$  - навчальна підвибірка довжини  $m$ ;

$x_n^k$  - контрольна підвибірка довжини  $k$ ;

$k = L - m, n = 1, \dots, N$  - номер розбиття.

Для кожного розбиття  $n$  будується алгоритм  $a_n(u, x)^m$ . Функціонал повного ковзкого контролю (*complete cross-validation, CCV*) визначається як середня (по всіх розбиттях) помилка на контролі:

$$CCV(x^L) = \frac{1}{N} \sum_{n=1}^N \frac{1}{k} \sum_{x_z \in x_n^k} [a_n(x_i, x_n^m) \neq y_i]$$

Функціонал повного ковзкого контролю характеризує узагальнюючу здатність методу найближчого сусіда.

Справедлива формула для ефективного обчислення *CCV* через профіль компактності:

$$CCV(x^L) = \sum_{j=1}^k R(j, x^L) \Gamma(j),$$

де  $\Gamma(j) = \frac{C_{L-1-j}^{m-1}}{C_{L-1}^m}$

Комбінаторний множник  $\Gamma(j)$  швидко убуває із зростанням  $j$ . Для мінімізації функціоналу  $CCV$  достатньо, щоб при малих  $j$  профіль  $R(j, X^L)$  брав значення, близькі до нуля. Це означає, що близькі об'єкти повинні лежати переважно в одному класі. Таким чином, профіль дійсно є формальним виразом гіпотези компактності [6].

Пропонується використовувати кластерний аналіз для розділення значень виходів мережі на групи, щоб звести задачу попередньої обробки даних при моделюванні за допомогою нейронної мережі до задачі попередньої обробки даних при класифікації, в якій використовується теорія профілю компактності.

В якості характеристики близькості вихідних значень нейронної мережі взято евклідову відстань між точками. Для довільного вектора  $v$  з числом елементів  $n$  евклідова норма знаходиться наступним чином:

$$\|v\| = \sqrt{\sum_{L=1}^m |v_L|^2}.$$

Евклідова відстань є найпопулярнішою метрикою в кластерному аналізі: вона відповідає інтуїтивним уявленням про близькість і, крім того, дуже вдало вписується своєю квадратичною формою у традиційно статистичні конструкції. Геометрично вона найкраще об'єднує об'єкти в кулястих скупченнях, які дуже типові для слабо корельованих сукупностей.

На першому кроці кластерного аналізу кожен об'єкт вважається окремим кластером. На наступному кроці об'єднуються два найближчих об'єкта, які утворюють новий клас, визначаються відстані від цього класу до всіх інших об'єктів, і розмірність матриці відстаней скорочується на одиницю. Процедура повторюється на поточній матриці відстаней, поки не буде досягнуто деяке число кластерів.

**Висновки та перспективи подальшого дослідження.** Таким чином, запропонований метод попередньої обробки даних дає більш якісне навчання нейронної мережі. Попередня обробка полягає у видаленні з масиву суперечливих прикладів. Пошук таких прикладів заснований на теорії профілю компактності в задачі класифікації. Щоб використовувати відомі рішення (теорію профілю компактності) в задачі моделювання, необхідно за допомогою кластерного аналізу виділити групи над значеннями вихідних параметрів нейронної мережі [11]. Чим краще буде зроблена попередня обробка, тим легше буде вирішена задача по виявленню аномальних станів трафіка комп'ютерної мережі.

1. Тимошук П.В., Лобур М.В.. Основи теорії проектування нейронних мереж // Навчальний посібник, Львів, 2007, с. 328
2. Ажмухамедов И.М., Марьенков А.Н. Обеспечение информационной безопасности компьютерных сетей на основе анализа сетевого трафика // № 1 / 2011
3. Астахов А., Актуальные вопросы выявления сетевых атак, <http://www.jetinfo.ru/2002/3/1/article1.3.2002.html>
4. Беляев А., Петренко С. Системы обнаружения аномалий: новые идеи в защите информации // Экспресс-Электроника №2, 2004 г. - С. 86 - 96.
5. Введение в сетевые атаки, [http://www.tshram.com/hacker/net\\_attacks.shtml#1](http://www.tshram.com/hacker/net_attacks.shtml#1)
6. Воронцов К.В. Комбинаторные оценки качества обучения по прецедентам // Докл. РАН. – 2004. – Т. 394, №2. – С. 175–178
7. Потемкин В.Г. Справочник по MATLAB, <http://matlab.exponenta.ru/ml/book2/index.php>
8. Дюк В., Самойленко А. Data Mining: учебный курс, СПб: Питер, 2001.
9. Емельянова Ю. Г., Талалаев А. А., Тищенко И. П., Фраленко В. П. Нейросетевая технология обнаружения сетевых атак на информационные ресурсы // Программные системы: теория и приложения : электрон. научн. журн. 2011. № 3(7), С. 3–15.
10. Жульков Е. Поиск уязвимостей в современных системах IDS// Открытые системы. СУБД -2003. - N 7/8. С. 37 - 42.
11. Качановский Ю.П., Коротков Е.А. ПРЕДОБРАБОТКА ДАННЫХ ДЛЯ ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ // Фундаментальные исследования. – 2011. – № 12 (часть 1). – стр. 117-120.

УДК 681.515.8

Здолбівська Н.В., Здолбівський А.П., Янчук Ю.В.  
Луцький національний технічний університет

## ДОСЛІДЖЕННЯ ПРОПУСКНОЇ ЗДАТНОСТІ АГРЕГОВАНИХ ІНТЕРНЕТ-КАНАЛІВ

**Здолбівська Н.В., Здолбівський А.П., Янчук Ю.В. Дослідження пропускної здатності агрегованих інтернет-каналів.** У статті представлено дослідження пропускної здатності основних способів агрегації інтернет-каналів, їхні переваги та недоліки, методи роботи з об'єднанням мережевих інтерфейсів у різних середовищах (як програмних так і фізичних). Розроблена та введена в дію агрегована мережа на базі комп'ютерної аудиторії ЛНТУ.

**Ключові слова:** агрегація, інтернет-канал, мережа, пропускна здатність.

**Рис. 5. Літ. 15.**

**Здолбівская Н.В., Здолбівский А.П., Янчук Ю.В. Исследование пропускной способности агрегированных интернет-каналов.** В статье представлено исследование пропускной способности основных способов агрегации интернет-каналов, их преимущества и недостатки, методы работы с объединением сетевых интерфейсов в различных средах (программных и физических). Разработана и введена в действие агрегированная сеть на базе компьютерной аудитории ЛНТУ.

**Ключевые слова:** агрегация, интернет-канал, сеть, пропускная способность.

**N. Zdolbitska, A.Zdolbitskyu., Y. Yanchuk. Research aggregate bandwidth of Internet channels.** This paper presents the research capacity main ways aggregation of Internet channels, their advantages and disadvantages, methods of work with the union network interfaces in different environments (both software and physical). Developed and put in place a network based on aggregated audience LNTU computer.

**Keywords:** aggregation, Internet channel, network bandwidth.

**Постановка наукової проблеми.** На сьогодні комп'ютерні технології заповнили чи не кожен сферу людської діяльності. Особливо актуальним є об'єднання суміжних ресурсів в єдине ціле задля отримання ефективних результатів. Мережеві ресурси дають змогу зекономити місце та час при організації колективної роботи. Проте питанням надійності та швидкості роботи мережевих систем часто не надають потрібного значення і в результаті доводиться зустрічатися з дестабілізацією функціонування мережі. На допомогу в такій ситуації приходять агрегація каналів. Оскільки лівова частка отримання інформації надходить із мережі Інтернет, то проведено дослідження агрегованих інтернет-каналів та порівняно їхню продуктивність зі звичайними прямими методами підключень, враховано всі «за» і «проти».

Агрегування каналів – технологія, яка дозволяє об'єднати кілька фізичних каналів в один логічний. Таке об'єднання дозволяє збільшувати пропускну здатність і надійність каналу. Саме на каналному рівні протоколів локальних мереж виявляються особливості використання надлишкових зв'язків, такі, як дублювання і зациклення кадрів в петлеподібних маршрутах [1]. І хоча ці проблеми можуть бути вирішені на більш високому мережевому рівні маршрутизаторами IP / IPX і програмними засобами операційних систем, підтримка надлишкових зв'язків комутаторами другого рівня, в порівнянні з вищесказаним має низку переваг. Перш за все, це зниження вартості мережі, оскільки багато її ділянок будуються без залучення відносно дорогих маршрутизаторів (комутаторів третього рівня). Крім того, швидкість реакції на обрив зв'язку часто підвищується.

Агрегування каналів застосовується досить широко та принципово відрізняється від способу використання надлишкових зв'язків – алгоритму побудови покриваючого дерева (Spanning Tree Algorithm, STA). Він переводить надлишкові зв'язки в «гарячий» резерв, залишаючи в робочому стані тільки мінімальний набір каналів, необхідний для забезпечення зв'язності сегментів мережі [2,3]. У цьому випадку підвищується надійність мережі, оскільки при відмові якого-небудь з'єднання в лад автоматично вводяться резервні зв'язки, і через невеликий проміжок часу працездатність мережі відновлюється.

При агрегування фізичних каналів всі надлишкові зв'язки залишаються в робочому стані, а наявний трафік розподіляється між ними для досягнення балансу завантаження. Головна перевага агрегування каналів в тому, що потенційно підвищується смуга пропускання: в ідеальних умовах смуга пропускання агрегування каналу може дорівнювати сумі смуг пропускання всіх об'єднаних в ньому каналів [4].



В даній статті практично реалізовано агрегування кількох інтернет потоків та досліджено можливості пропускної здатності модернізованої мережі.

**Аналіз досліджень.** Агрегація інтернет-каналів – явище відносно нове. Не кожна система навіть п'ятилітньої давності підтримує агрегацію, але оскільки способів введення в дію є декілька, то це однозначно спрощує її використання. Найбільше застосування даного методу з'єднання з мережею припадає на підприємства та організації, які потребують стабільного та швидкого з'єднання. На даний момент існує кілька видів об'єднання інтернет-каналів. Найпоширенішими з них є: комутатор-комутатор, комутатор-маршрутизатор, комутатор-хост.

У даній статті реалізовано фізичну агрегацію типу «комутатор-маршрутизатор» та досліджено ефективність застосування такого виду вдосконалення мережі. Існує також така агрегація каналів, яка дозволяє безпосередньо в комп'ютері об'єднати кілька мережевих карт в одну логічну систему(в системах Linux – Bonding, Windows Server 2012 – NIC Teaming) [5,6], але, з огляду на персональність, такий вид поєднання є не дуже доцільним для комп'ютерної мережі.

**Виклад основного матеріалу та обґрунтування результатів дослідження.** В даній праці впроваджено в мережу аудиторії (рис. 1) агрегацію інтернет-каналів (рис. 2). Стаття є актуальною також для підприємств, де є необхідним мати впевнене з'єднання з провайдером, а подекуди і кількома, для забезпечення надійності роботи установи.

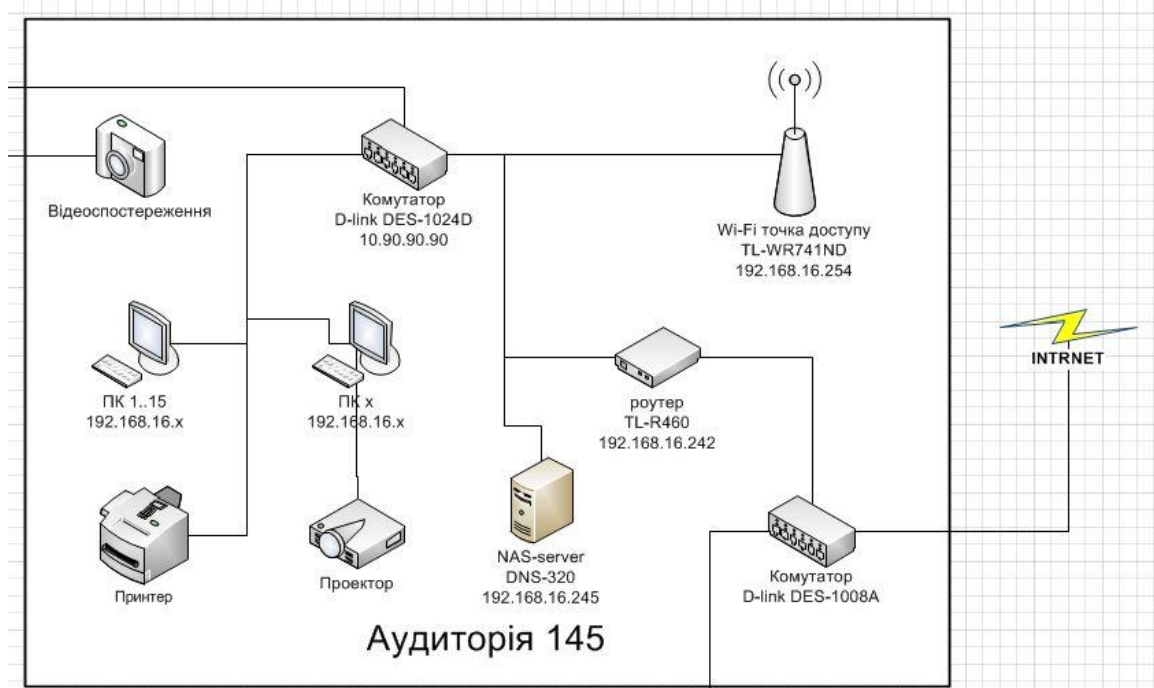


Рис. 1. Схема мережі навчальної аудиторії без застосування агрегації каналів

Фізична реалізація впроваджена за принципом «маршрутизатор-комутатор», де замість одного потоку каналів їх іде декілька(позначка 1<sup>A</sup> на рисунку 2). Об'єднання каналів реалізовано за допомогою комутатора D-Link DES-1008A та роутера з можливістю балансування каналів TP-Link TL-R480+ [7]. В ході практичних досліджень отримано висновок, що агрегація інтернет-каналів значно пришвидшила роботу мережі, завдяки принципу розпаралеленості. Так сусідні комп'ютери у мережі можуть обмінюватися трафіком через різні канали, що дає високий приріс продуктивності.

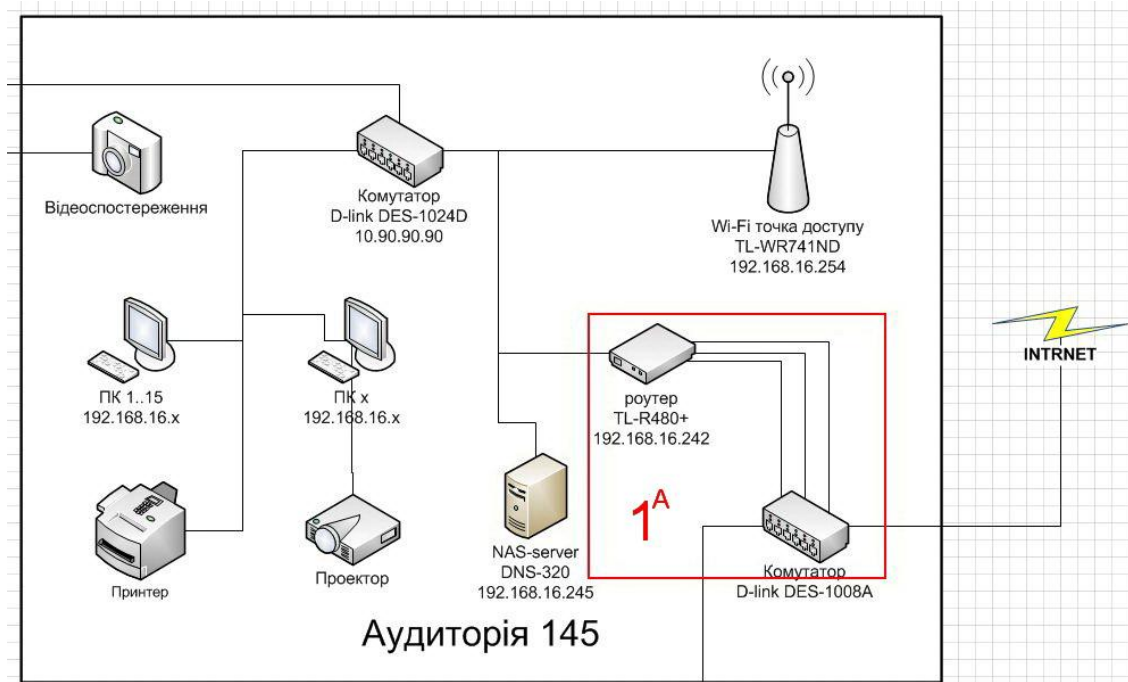


Рис. 2. Схема мережі навчальної аудиторії із застосуванням агрегації каналів

Для аналізу завантаження сайтів взято громіздкий ресурс <http://kurs.com.ua/>. Середній час очікування повного завантаження без агрегації каналів сторінки склав близько 4 секунд. Під час агрегування трьома каналами отримано результат у 2,5-3секунди, що дав приріст у швидкості на 25-37%. Такий результат зумовлений тим, що отримуваний трафік порівняно невеликий для розгону агрегації і тим, що частина запитів від сусідніх комп'ютерів завдавала навантаження на новостворену мережу. Очевидно, що без агрегування канал було навантажено дуже сильно і час очікування зростає, так як кожен запит клієнтів ставав у чергу в одному і тому ж каналі.

При тестуванні інтернет-ресурсів на швидкість отримано висновок, що багато порталів прив'язуються до одного каналу та йдуть по ньому так і не використавши інші наявні в агрегованій системі.

Що стосується мультипоточної передачі інформації, то ситуація тут кардинально інша. Дослідження пропускної здатності агрегованих інтернет-каналів на базі торент програми Utorrent дало доволі хороші результати. За рахунок наявності багатьох каналів обміну трафіком досягнуто збільшення швидкості у кілька разів.

На проміжку 1 рисунка 3 агреговано два канали по два мегабіти. Досягнуто зростання швидкості за хвилеподібною тенденцією.



Рис. 3. Агрегація двох інтернет-каналів

На межі частини 1 та частини 2 графіка, під час того як було здійснено відключення агрегації, швидкість впала практично до нуля, адже здійснювалася настройка та пошук робочого

каналу для передачі трафіку. Повторне ввімкнення агрегації двох каналів дало швидкий темп росту швидкості. При відключенні одного з каналів, після так званої адаптації до всіх умов (в тому числі обмежень, які накладаються на канал адміністратором), права частина другого сегменту графіка швидкості показала майже ідеальні два мегабіти, які надані мережею на один інтернет-канал.

Під час повернення до агрегації двох каналів (рис. 4) на секторі 3 відбувся характерний спад на початку перемикавання та подальше зростання швидкості. Об'єднання трьох двомегабітних каналів (рис.4) дало сталий приріст у швидкості, який зображено на секторі 4. Слід зауважити, що відслідковано тенденцію спаду швидкості навіть при додаванні нових інтернет каналів до вже існуючих чи то одного як на межі 2 та 3 секторів, чи навіть двох – межа 3 та 4 рисунка 4.

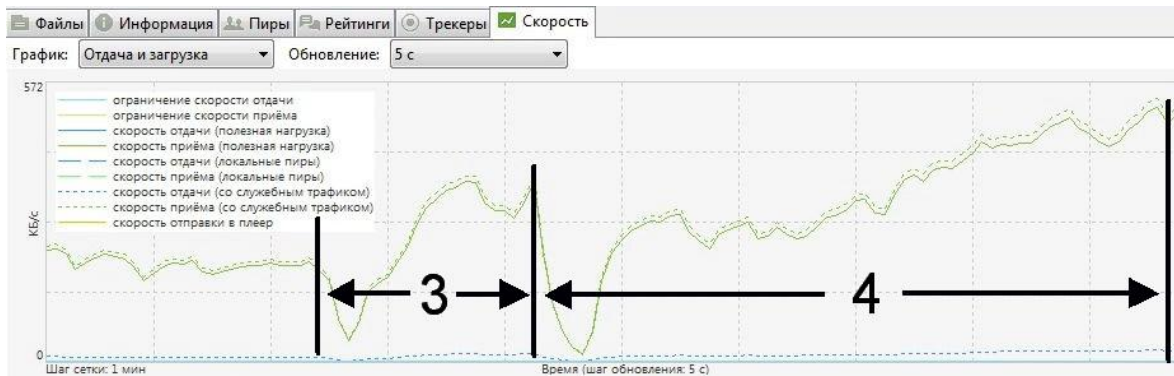


Рис. 4. Агрегація двох та трьох інтернет-каналів

Об'єднання трьох двомегабітних каналів дало близько 80% від ідеально можливого результату, навіть при працюючих суміжних комп'ютерах мережі, які створюють завади для передачі потоків трафіку.

Під час поетапного переходу (сегмент 5 рисунка 5) від трьохканального до двоканального режимів роботи, а після того і до прямого каналу зв'язку без агрегації відбувся типовий спад (межа 5 та 6) і як результат обмін трафіком по одному каналові, майже за ідеальних показників.



Рис. 5. Схема мережі навчальної аудиторії без застосування агрегації каналів

Що стосується загального приросту продуктивності мережі, то відбувається він за рахунок того що одне і те ж навантаження на один звичайний канал значно більше дається в знаки, ніж той самий обсяг трафіку по кільком каналах передачі даних.

**Висновки.** На практиці було доведено, що агрегація інтернет-каналів чудово збільшує пропускну здатність мережі та у значній мірі знімає питання навантаження, а в результаті часу очікування відповіді ресурсу.

З огляду на роботу процесу агрегації зроблено такі висновки у порівнянні зі звичайним не агрегованим інтернет-каналом:

– переваги:

- збільшується швидкість передачі інформації, як на одиничному комп'ютері, так і сусідніх комп'ютерах мережі;
  - значно якісніший рівень стабільності, за рахунок можливості мати резервний канал
  - менша навантаженість роботи мережі;
  - менший час очікування;
  - можливість одночасного підключення до двох провайдерів;
- недоліки:
- тимчасовий спад швидкості під час перемикання режимів комутації, обумовлений пошуком надійного каналу для передачі інформації;
  - додаткові витрати на обладнання, яке необхідне для агрегації;
  - складність в налаштуванні.

На основі викладеного матеріалу можна зробити висновок, що дослідження пропускнуої здатності агрегованих інтернет-каналів дало досить хороший приріст продуктивності у роботі з мережею і є актуальним продовжувати аналогічні дослідження за різних умов агрегації(кількості агрегованих каналів, обладнання, що бере участь у процесі агрегації, операційних системах задіяних у мережі), для знаходження найоптимальнішого для тієї чи іншої потреби.

1. Олифер Н.А. «Журнал сетевых решений/LAN», № 02, 2002. – 32 с.
2. Самойленко Н.С. Загальна інформація по агрегації каналів [Електронний ресурс]. – Режим доступу: [http://xgu.ru/wiki/Агрегирование\\_каналов](http://xgu.ru/wiki/Агрегирование_каналов)
3. Агрегация каналов [Електронний ресурс]. – Режим доступу: [https://ru.wikipedia.org/wiki/Агрегирование\\_каналов](https://ru.wikipedia.org/wiki/Агрегирование_каналов).
4. Олифер В. Г., Олифер Н. А. Коммутируемые сети Ethernet // Компьютерные сети. Принципы, технологии, протоколы. – 4-е изд. – СПб.:Питер, 2010. — с. 459-464
5. На пути повышения надёжности и скорости: Linux bonding [Електронний ресурс]. – Режим доступу: <http://samag.ru/archive/article/355>
6. NIC Teaming в Windows Server 2012 [Електронний ресурс]. – Режим доступу: <http://habrahabr.ru/company/microsoft/blog/162509/>
7. Маршрутизатор с балансировкой нагрузки TL-R480T+ [Електронний ресурс]. – Режим доступу: <http://www.tp-link.ua/products/details/?categoryid=&model=TL-R480T%2b#over/>
8. Microsoft Corporation. Компьютерные сети. Сертификация Network+. – М.: Русская редакция, 2008. – 704с.
9. Microsoft TCP/IP. – : Русская редакция, 1999. – 214с.
10. Борисенко Олексій Андрійович Керуючі системи. – К.: Центр навчальної літератури, 2004. – 216с.
11. Лук'янова Валентина Вячеславівна Комп'ютерний аналіз даних. – К.: Вид. центр "Академія", 2003. – 344с.
12. Остропицький В.М., Мозговая И.В. Internet-технологии. – Днепропетровск: Инновация, 2004.- 192с.
13. Локальные сети. Модернизация и поиск неисправностей: Александр Поляк-Брагинский – Москва, БХВ-Петербург, 2007 г. – 640 с.
14. Убайдуллаев Р.Р. «Волокно-оптические сети» – М.: Эко-Трендз, 1998. – 268с.
15. В.А. Дуленко, Ю.С. Кабальнов «Основы современных компьютерных технологий» – Уфа: ОН и УЮИ ВМД РФ, 2003. – 122 с.

УДК 61:004.651(075.8)

Андрущак І.Є., Шендрик Ю.В.

Луцький національний технічний університет

## РЕАЛІЗАЦІЯ МЕТОДУ ПРИЙНЯТТЯ РІШЕННЯ ІЗ ЗАСТОСУВАННЯМ ТЕОРІЇ ВТРАТ

**Андрущак І.Є., Шендрик Ю.В. Реалізація методу прийняття рішення із застосуванням теорії втрат.** В роботі реалізовано системний підхід до прийняття рішень, що враховує ставлення ОПР до різних стратегій лікування. Встановлені оцінки підтверджено експериментально як із зміною кількості атрибутів, так і обсягу наборів навчальних даних.

**Ключові слова:** теорія втрат, прийняття рішень, теорія очікуваної корисності.

**Андрущак И.Е., Шендрик Ю.В. Реализация метода принятия решения с применением теории потерь.** В работе реализован системный подход к принятию решений, учитывает отношение ОПР в различных стратегий лечения. Установленные оценки подтверждено экспериментально как с изменением количества атрибутов, так и объема наборов учебных данных.

**Ключевые слова:** теория потерь, принятие решений, теория ожидаемой полезности

**Andrushchak I., Shendrik Y. Realization of decision-making method is with application of theory of losses.** This paper implemented a systematic approach to decision making that takes into account the ratio of ODA to different treatment strategies. Installed evaluation confirmed experimentally as a change in the number of attributes and sets the amount of training data.

**Keywords:** theory of losses, making decision, theory of utility.

### Вступ

Прийняття рішення дуже часто визначається невизначеністю, яка впливає на увесь процес вироблення рішення. В намаганні змодельовати невизначеність фахівці в галузі прийняття рішень пропонують методики, що ґрунтуються на теорії ймовірності і статистиці [Edwards,2007], теорії інформації та ентропії [Shannon,1962], нечіткій логіці [Zimmer, 2000].

В клінічних медичних дослідженнях багато зусиль прикладено розробці систем підтримки прийняття рішень для діагностики та лікування ряду захворювань [Schurink, 2005]. Більшість з цих систем ґрунтуються на ймовірнісних прогностичних моделях. Незважаючи на те, що прогностичні моделі зарекомендували себе як потенційно придатні до клінічних застосувань, все ж вони не відповідають очікуванням осіб, що приймають рішення (ОПР) з метою вдосконалення прийняття рішень. Однією з причин є та, що більшість медичних систем підтримки прийняття рішень (СППР) ґрунтуються на теорії очікуваної корисності, яка дуже часто не відповідає реальній дійсності. До того ж більшість моделей в медицині не враховують побажань ОПР, які є ключовими в раціональному процесі прийняття рішень.

Метою даної роботи є реалізувати підхід до прийняття рішень, що враховує ставлення ОПР до різних стратегій лікування.

Підхід полягає в наступному. По-перше, ми відходимо від традиційної теорії очікуваної корисності в намаганні задовольнити як формальні критерії раціональності, так і людську інтуїцію щодо правильних рішень. По-друге, застосовується теорія втрат (regret theory) [Tsalatsanis, 2010], оскільки втрата – це когнітивна емоція, що поєднує як раціональність, так і інтуїцію, які є ключовими елементами в прийнятті рішень, яке б використовувало особисті переваги ОПР. Отже, ми переформулюємо метод DCA з точки зору теорії втрат з метою оцінки альтернативних стратегій лікування з урахуванням ставлення до них та переваг ОПР.

Для реалізації такого підходу спочатку розраховується порогова ймовірність, при якій ОПР байдуже, які альтернативні стратегії прийняті. Це робиться визначаючи рівень втрати, який відчуває ОПР при прийнятті помилкового рішення. Далі застосовується метод DCA на основі втрат для визначення оптимальної стратегії. Оптимальною стратегією є стратегія, яка приносить найменше втрат.

### Прийняття рішень на основі теорії втрат

На рис. 1 зображено типове дерево рішень для призначення лікування на основі прогностичної моделі. Тут представлено дві конкуруючі стратегії (лікувати та не лікувати) і чотири можливі наслідки.

Далі використовуватимуться такі позначення для випадкових подій:  $D+$  - наявність захворювання,  $D-$  - захворювання відсутнє,  $Rx+$  - результати обстеження вказують на наявність захворювання та необхідність призначення лікування (для вибраної порогової ймовірності  $p_t$ ), тобто «тест позитивний»,  $Rx-$  - тест негативний. Тоді  $p = P(D+)$  - ймовірність наявності захворювання,  $p = P(D-)$  - ймовірність відсутності захворювання,  $U_i, i = \overline{1,4}$  - корисності, що відповідають кожному наслідку. А саме,  $U_i = U(Rx_i, D_i)$ , де  $U(\bullet, \bullet)$  - деяка функція корисності,  $Rx_i \in \{Rx+, Rx-\}$ ,  $D_i \in \{D+, D-\}$ . Так  $U_1 = U(Rx+, D+)$  - корисність призначеного лікування для пацієнта, що має захворювання,  $U_2 = U(Rx+, D-)$  - корисність призначеного лікування для пацієнта, що не має захворювання.

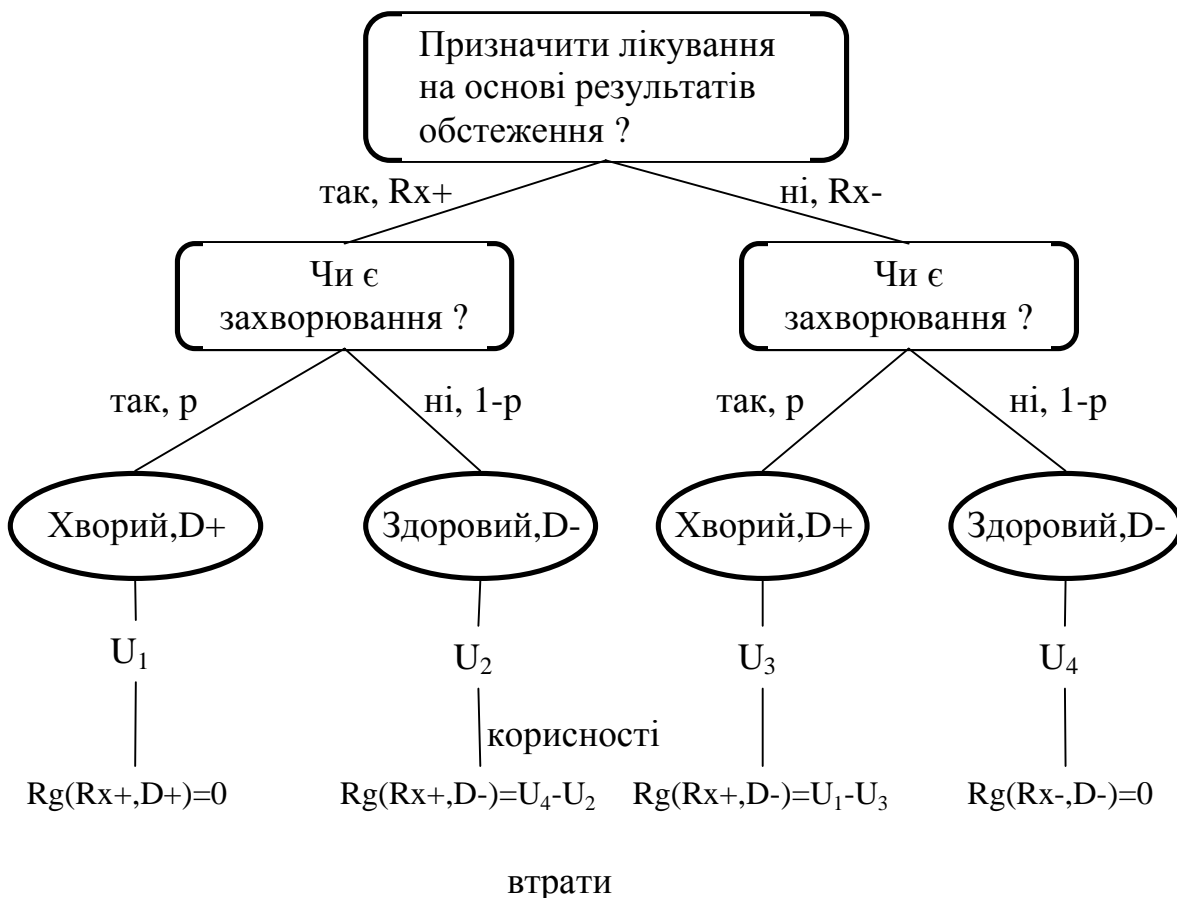


Рис. 1. Дерево рішень на основі теорії очікуваної корисності.

Зауважимо, що термін «лікування» вживається в даній роботі в загальному розумінні і він включає будь-яке медичне втручання, наприклад, терапевтичне лікування, хірургічну операцію або діагностичну процедуру.

Ймовірнісна природа прогностичних моделей значно ускладнює процес прийняття рішень. Наприклад, якщо прогностична модель оцінює ймовірність захворювання для пацієнта в 40%, то не зрозуміло, чи призначати пацієнту лікування чи ні. Відповідь з точки зору класичної теорії прийняття рішень полягає у застосуванні поняття порогової ймовірності  $p_t$ , яка визначається як ймовірність, при якій особі, що приймає рішення байдуже, яку стратегію обрати (призначити



лікування чи ні). Використовуючи поняття порогової ймовірності пацієнту слід призначити лікування, якщо  $p \geq p_t$ , і навпаки в протилежному випадку.

Однак оскільки в більшості випадків рішення приймаються в умовах невизначеності, то після прийняття рішення може виявитися, що більш прийнятною є інша альтернатива. Така інформація вводить для ОПР поняття втрат - англійською «regret», що дослівно перекладається як «жаль», «смуток» [Tsalatsanis, 2010].

Формально втрати можуть бути виражені як різниця між корисностями від вчиненої дії і від дії, яку зважаючи на ретроспективу слід було здійснити [Tsalatsanis, 2010]. Розрахунок втрат представлено на дереві рішень (рис.1).

Втрати можуть відчуватися будь-якою стороною, залученою в процес прийняття рішень. Ми для визначеності вважатимемо, що ОПР є лікуючий лікар.

Для початку застосуємо теорію втрат для оцінки порогової ймовірності  $p_t$ , при якій альтернативні стратегії (призначити лікування або не призначити) є рівносильними.

Виходячи з дерева рішень (рис.1) бачимо, що очікувані втрати від призначеного лікування становлять  $(1 - p)(U_4 - U_2)$ , втрати від ненаданого лікування -  $p(U_1 - U_3)$ . Виходячи з означення порогової ймовірності приходимо до співвідношення для визначення  $p_t$ :

$$(1 - p_t)(U_4 - U_2) = p_t(U_1 - U_3).$$

Звідси:

$$p_t = \frac{1}{1 + \frac{U_1 - U_3}{U_4 - U_2}} \quad (1)$$

- співвідношення для визначення порогової ймовірності на основі втрат.

Зауважимо, що співвідношення (1) невизначене для  $U_4 - U_2 = 0$ , тобто у випадку, коли немає жодних втрат від призначення непотрібного лікування. В такому разі  $p_t = 0$  - це практично нереальна ситуація.

#### Метод DCA на основі втрат

Нагадаємо, що метод DCA може бути застосований для визначення стратегії, якої слід дотримуватися виходячи із значення порогової ймовірності (наприклад, використати модель і призначити лікування тільки у випадку  $p \geq p_t$ , лікувати всіх, не лікувати нікого).

У класичному методі DCA розраховується сукупна функція корисності [Vickers, 2006]. При застосуванні теорії втрат обчислюється очікувана втрата при використанні певної стратегії [Tsalatsanis, 2010]. Так на основі дерева рішень маємо:

$$ERg[\text{модель}] = P(p < p_t / D+)(U_1 - U_3) + P(p \geq p_t / D-)(U_4 - U_2), \quad (2)$$

де  $Rg[\bullet]$  - втрати, пов'язані з використанням певної стратегії. Таким же чином з рис.2 маємо, що:

$$ERg[\text{лікувати всіх}] = (1 - p)(U_4 - U_2), \quad (3)$$

$$ERg[\text{не лікувати нікого}] = p(U_1 - U_3). \quad (4)$$

При аналізі стратегій на основі втрат використовується величина різниці сукупних очікуваних втрат (Net Expected Regret Difference, NERD), яка може бути розрахована [Tsalatsanis, 2010] для NERD[нікого не лікувати, модель], NERD[нікого не лікувати, лікувати всіх], NERD[нікого не лікувати, нікого не лікувати] за формулою:

$$NERD = \frac{\#TP}{n} - \frac{\#FP}{n} \frac{p_t}{1 - p_t}, \quad (5)$$

а для NERD[лікувати всіх, модель] за формулою:

$$NERD = \frac{\#TN}{n} \frac{p_t}{1-p_t} - \frac{\#FN}{n}. \quad (6)$$

Тут  $\#TP, \#FP, \#TN, \#FN$  - кількості вірно та помилково позитивних, вірно та помилково негативних пацієнтів, розраховані для певної порогової ймовірності. Далі наведемо формулювання алгоритму методу DCA на основі втрат.

**Вхідні дані:** інформація про пацієнтів: кількість  $n$ , вектор діагнозів  $d = (d_1, \dots, d_n)^T$  та матриця результатів обстежень  $I = (I_1, \dots, I_n) \in R^{m \times n}$ . Тут

$$d_i = \begin{cases} 1, & \text{захворювання є} \\ 0, & \text{захворювання немає} \end{cases}, \quad I_i \in R^m - \text{діагностичні дані для кожного пацієнта, що}$$

можуть бути перетворені у ймовірнісне значення  $\hat{p}_i$ , наприклад на основі логістичної регресії.

**Вихідні дані:** різниці сукупних очікуваних втрат для усіх моделей та стратегій: лікувати всіх, лікувати нікого як функції від порогової ймовірності.

**Метод:**

1. Вибираємо порогову ймовірність  $p_t$ .
2. Вважаємо, що пацієнт є позитивним, якщо  $\hat{p}_i \geq p_t$ .
3. Обчислюємо кількості вірно та помилково позитивних.
4. Обчислюємо значення NERD(нікого не лікувати, модель) на основі (5).
5. Обчислюємо NERD(лікувати всіх, модель) на основі (6).
6. Обчислюємо значення NERD(нікого не лікувати, лікувати всіх) на основі (5).
7. Повторюємо кроки 1-6 на інтервалі значень  $p_t$ .
8. Креслимо графіки усіх NERD, обчислених на кроках 4-6 залежно від значення порогової ймовірності.

Використовуючи метод DCA на основі втрат оптимальне рішення для кожної порогової ймовірності вибирається на основі принципу транзитивності, а саме, якщо  $NERD(\text{стратегія 1, стратегія 2}) > NERD(\text{стратегія 2, стратегія 3}) > 0$ , то стратегія 2 краща від стратегії 1, а стратегія 3 краща від стратегії 2. Отже, стратегія 3 – оптимальна для даної порогової ймовірності.

Зауважимо, що метод може бути застосований не лише до чисельних вхідних даних про пацієнта, але й мультимедійних, наприклад, ЕКГ, ЕЕГ, УЗД, КТ, МРТ і ін., які повинні попередньо бути виражені в ймовірнісних значеннях.

**Висновки**

В роботі представлено програмну реалізацію методу DCA на основі втрат з метою підтримки прийняття рішень щодо вибору оптимальної стратегії лікування. Порівняно із традиційним методом DCA підхід на основі втрат враховує переваги ОПР щодо стратегій на основі попереднього досвіду та використовує інтуїтивно більш зрозуміле поняття втрати.

Метод реалізовано у вигляді Netbeans-додатку з вхідними даними у вигляді xml-файлу, що робить можливим його використання в різноманітних Інтернет-проектах, пов'язаних з прийняттям рішень.

1. A.J. Vickers, A.M.Cronin, E.B.Elkin, M.Gonen Extensions to decision curve analysis, a novel method for evaluating diagnostic tests, prediction models and molecular markers, BMC Medical Informatics and Decision Making 2008, 8:53 doi: 10.1186/1472-6947-8-53
2. Vickers AJ, Elkin EB: Decision curve analysis: a novel method for evaluating prediction models. Med Decis Making 2006, 26(6):565-574.
3. Tsalatsanis et al.: A regret theory approach to decision curve analysis: A novel method for eliciting decision makers' preferences and decision making. BMC Medical Informatics and Decision Making 2010 10:51.



4. Марценюк В.П. Математичні моделі в системі підтримки прийняття рішень страхового забезпечення лікування онкологічних захворювань: підхід на основі динаміки Гомперца / В.П. Марценюк, І.Є. Андрушак, І.С. Гвоздецька, Н.Я. Климук // Доповіді Національної академії наук України. –2012. – №10. – С. 34-39.
5. Марценюк В.П. Підхід на основі актуарних математичних моделей до задач страхової медицини / В. П. Марценюк, І.Є. Андрушак, Н.Я. Климук // Медична інформатика та інженерія. Науково-практичний журнал. – 2010. – №4. – С. 85-87.
6. Марценюк В.П. О модели онкологического заболевания со временем пребывания на стадии в соответствии с распределением Гомперца / В.П. Марценюк, Н.Я. Климук // Проблемы управления и информатики. Международный научно-технический журнал . – 2012. – № 6. – С. 137-143.
7. Марценюк В.П., Семенець А.В. Медична інформатика. Інструментальні та експертні системи. – Тернопіль: Укрмедкнига, 2004. – 222 с.
8. Edwards W, Miles RFJ, von Winterfeldt D: Advances in decision analysis. From foundations to applications. New York: Cambridge University Press 2007.
9. Shannon C, Weaver W: The mathematical theory of communication. Urbana: The University of Illinois Press 1962.
10. Zimmer man H: An application-oriented view of modeling uncertainty. European Journal of Operational Research 2000, 122: 190-198.
11. Schurink CAM, Lucas PJF, Hoepelman IM, Bonten MJM: Computer-assisted decision support for the diagnosis and treatment of infectious diseases in intensive care units. The Lancet Infectious Diseases 2005, 5(5):305-312.

УДК 519.6: 531

Валяшек В.Б., к.ф.-м.н., Каплун А. В. д.ф.-м.н.

Тернопільський національний технічний університет імені Івана Пулюя, Тернопіль

Юзевич В.М., д.п.н.

Фізико-механічний інституту ім. Г. В. Карпенка НАН України, Львів

## МАТЕМАТИЧНЕ ТА КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ФІЗИЧНИХ ХАРАКТЕРИСТИК МАТЕРІАЛУ У ВЕРШИНІ ТРІЩИНИ З УРАХУВАННЯМ ЕФЕКТУ ЗМІЦНЕННЯ

**Валяшек В.Б., Каплун А.В., Юзевич В.М. Математичне та комп'ютерне моделювання фізичних характеристик матеріалу у вершині тріщини з урахуванням ефекту зміцнення.** На основі співвідношень механіки деформованого твердого тіла, фізики поверхневих явищ та електрохімії побудовано математичну модель для визначення енергетичних та електрохімічних характеристик механічно навантаженого елемента конструкції на межі з водним електролітом. Розглядається трубопровід із сталі 20 у 3 % розчині хлориду натрію (NaCl). Розроблено алгоритм та реалізовано обчислювальну процедуру щодо розрахунку поверхневої енергії пластичного деформування і перенапруження реакції розчинення металу для ювенільної поверхні у вершині тріщини при механічному навантаженні об'єкту (труби) в межах від границі текучості матеріалу  $\sigma_T$  до границі  $1,246 \cdot \sigma_T$ , встановленої нормативним документом. Проведено оцінювання ресурсу навантаженого внутрішнім тиском трубопроводу з пітингом на зовнішній поверхні, у вершині якого знаходиться корозійна тріщина.

**Ключові слова:** тріщина, ювенільна поверхня, коефіцієнт інтенсивності напружень, перенапруження, поверхнева енергія пластичного деформування, корозійний струм.

**Валяшек В.Б., Каплун А.В., Юзевич В.М. Математическое та компьютерное моделирование физических характеристик материала в вершине трещины с учетом эффекта упрочнения.** На основе соотношений механики деформируемого твёрдого тела, физики поверхностных явлений и электрохимии построена математическая модель для определения энергетических и электрохимических характеристик механически нагруженного элемента конструкции на границе с водным электролитом. Рассматривается трубопровод из стали 20 в 3 % растворе хлорида натрия (NaCl). Разработан алгоритм и реализована вычислительная процедура расчета поверхностной энергии пластического деформирования и перенапряжения реакции растворения металла для ювенильной поверхности в вершине трещины при механическом нагружении объекта (трубы) в пределах от границы текучести материала  $\sigma_T$  до границы  $1,246 \cdot \sigma_T$ , установленной нормативным документом. Проведена оценка ресурса нагруженного внутренним давлением трубопровода с питингом на внешней поверхности, в вершине которого находится коррозийная трещина.

**Ключевые слова:** трещина, ювенильная поверхность, коэффициент интенсивности напряжений, перенапряжение, поверхностная энергия пластического деформирования, коррозионный ток.

**V. Valiashek, A. Kaplun, V. Yuzevych Mathematical and computer modeling of physical characteristics of material in top of crack taking into account effect of strengthening.** On the basis of correlations of mechanics of deformable solid, physics of the surface phenomena and electrochemistry a mathematical model is created for determination of energetic and electrochemical characteristics of the mechanically loaded element of construction on a interface with a water electrolyte. An algorithm is worked out and calculable procedure is realized in relation to the calculation of surface energy of flowage and overstrain of dissolution reaction of metal for a juvenile surface in the top of crack at the mechanical loading of object (pipes) in limits from the limit of fluidity of material  $\sigma_T$  to the limit  $1,246 \cdot \sigma_T$ , set by a normative document. The estimation of resource of the pipeline loaded with intrinsic pressure is conducted with pitting on an external surface. It is taken into account that a corrosive crack is in the top of pitting.

**Keywords:** Cracks, juvenile surface, coefficient of intensity of tensions, overstrain, surface energy of flowage, corrosive current.

**Постановка наукової проблеми.** Розглядаємо метал в розчині електроліту з поверхневим дефектом у вигляді пітинга, у вершині якого знаходиться тріщина (рис. 1, зображено проєкцію на площину  $xOy$ ). Зовнішнє корозійне середовище (в пітингу та тріщині) – водний розчин електроліту. Під дією механічного навантаження на метал (відбувається розтяг в напрямку осі  $Oy$ , якому відповідають напруження  $\sigma_{yy}$ ) у вершині тріщини відбувається руйнування пасивних плівок і утворюється ювенільна поверхня (ЮП) шириною  $\delta$  (довжиною  $L$ ), проєкція якої є півколо  $\pi r$ , а також область пластичних деформацій при  $\sigma_{yy} = \sigma_T$  ( $\sigma_T$  – границя текучості матеріалу) [1,2]. Відстань  $\delta$  у першому наближенні можна трактувати як розкриття вершини тріщини  $\delta_{1c}$ . Під впливом механічного навантаження і корозійного середовища вершина тріщини  $i$ , відповідно, ЮП поширюється в глибину тіла практично перпендикулярно до межі в напрямку осі  $x$  [2]. В околі вершини тріщини проходять катодні та анодні електрохімічні реакції, які характеризують корозійний процес розчинення металу. Вершину тріщини (ЮП) – трактуємо як анод (А), за її межами на бокових поверхнях – катодна область (К) [1,2]. Система «А – К» являє собою електрохімічну пару (рис. 1).

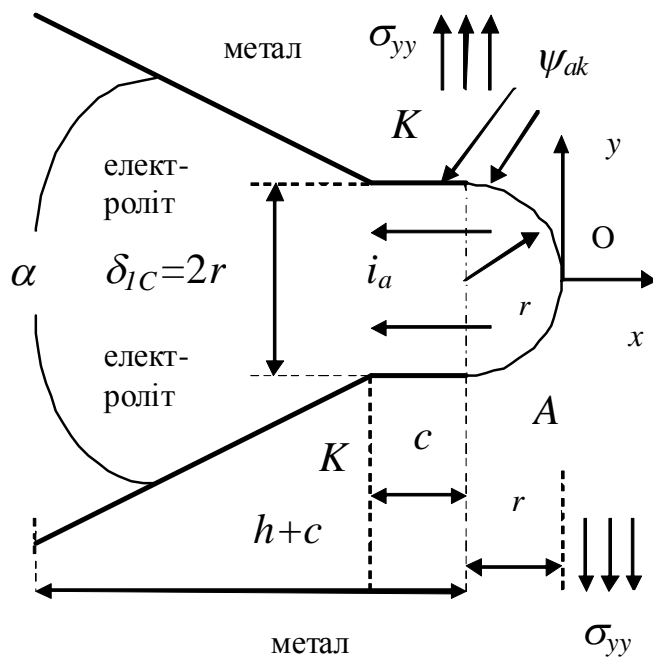


Рис. 1. Пітинг (h) з тріщиною (c) у трубі з відзначенням катодної (K) та анодної (A) ділянок. Загальна глибина дефекту  $h+c+r$ ;  $\alpha$ ,  $\delta_{IC}$  – кут і розкриття у вершині тріщини;  $\Delta\psi_{ak}$  – різниця потенціалів між анодною і катодною частинами;  $I_a$  – густина анодного корозійного струму.

Fig. 1. A pitting (h) with a crack (c) in a pipe with marking of cathode (K) and anodic (A) areas. Total depth of defect  $h+c+r$ ;  $\alpha$ ,  $\delta_{IC}$  – the angle and the opening in the top of crack;  $\Delta\psi_{ak}$  – a difference of potentials between anodic and cathode parts;  $I_a$  – density of anodic corrosive current.

В даній праці потрібно аналітично описати і дослідити сумісний вплив зовнішнього корозійного середовища та механічного навантаження на поверхневу енергію пластичного деформування, перенапруження реакції розчинення металу і анодний корозійний струм для ювенільної поверхні у вершині тріщини при механічному навантаженні об'єкту (трубопроводу) в межах від межі текучості матеріалу  $\sigma_T$  до межі міцності  $\sigma_B$  ( $\sigma_{yy}/\sigma_T = \sigma/\sigma_T = 1 \div \sigma_B/\sigma_T$ ). Вказаний діапазон навантажень вибрано тому, що при  $\sigma_T$  починається пластичне деформування матеріалу, а при  $\sigma_B$  деформування завершується утворенням магістральної тріщини. Дана задача розглядається для аналізу умов деформування металу в корозійному середовищі з подальшим прогнозуванням ресурсу елементів конструкцій, які використовуються у трубопроводах і парогенераторах атомних електростанцій (АЕС) [3].

Ювенільна поверхня – поверхня металу у вершині тріщини, вільна від оксидів та інших забруднень [1]. В даній задачі (на рис. 1) її площа фактично  $\pi rL$ , але потік корозійних частинок проходить через площадку площею  $2rL = \delta L$ .

Предметом даного дослідження є макроскопічні співвідношення механіки деформівного твердого тіла, фізики поверхневих явищ тіла та електрохімії для визначення взаємозв'язків енергії пластичного деформування поверхні та перенапруження анодної реакції для ЮП шириною  $\delta$  у вершині тріщини на границі між металом і зовнішнім корозійним середовищем.

**Аналіз останніх досліджень.** Для оцінювання змін енергетичних характеристик міжфазних шарів доцільно використати експериментальні дані щодо зв'язку між коефіцієнтом інтенсивності напружень (КІН)  $K_{IC}$  (Па· $\sqrt{м}$ ) та розкриттям  $\delta_{IC}$  у вершині тріщини для сталі 20 у 3 % розчині NaCl (табл. 1) [2] при розтягуючих напруженнях ( $\sigma/\sigma_T = 1 \div \sigma_B/\sigma_T$ ).

**Таблиця 1**

**Числові значення розкриття  $\delta_{1C}$  і густини корозійного струму  $I_a$  у вершині тріщини для сталі 20 в корозійному середовищі при механічних розтягуючих напруженнях  $\sigma/\sigma_T$**

Розкриття $\delta_{1C}$ , мкм	15,09	22,76	34,33	51,79
Густина корозійного струму $I_a$ , $\frac{мМ}{рік}$	0,275	0,327	0,379	0,437
Відносне механічне напруження $\sigma/\sigma_T$	1,0	1,082	1,164	1,246

$K_1$  зв'язаний з розкриттям тріщини  $\delta_{1C}$  та перенапруженням  $\zeta$  реакції розчинення металу на основі відомих співвідношень [1,4,5]

$$K_1 = \sqrt{\frac{E}{1-\vartheta^2} \cdot \left( WPL - Z_{si} F \rho \delta \frac{\zeta}{M} \right)}, \quad K_1 = K_{1SCC} = \sqrt{E \cdot \sigma_T \cdot \delta_{1C}}. \quad (1)$$

Тут  $Z_{si}$  – формальний заряд сольватованих (гідратованих) іонів;  $F = 96500$  Кл/моль – постійна Фарадея;  $\delta = 2r$  – ширина фронту мікротріщини, що насувається, м (довжина фронту  $L$  перпендикулярна проекції на рис. 1);  $M$  – молекулярна маса металу, кг/моль ( $M = 0,0558$  кг/моль – для сталі);  $K_{1SCC}$  – порогове значення КІН, тобто мінімальне значення, що відповідає початку поширення тріщини під впливом сумісної дії механічного навантаження і корозійного середовища;  $WPL$  – поверхнева енергія пластичного деформування;  $E, \vartheta$  – модуль поздовжньої пружності матеріалу та коефіцієнт Пуассона відповідно;  $\zeta$  – електрохімічне перенапруження (В), тобто відхилення електродного потенціалу від його рівноважного (по відношенню до приелектродного складу розчину) термодинамічного значення при поляризації електроду під струмом [6]. Необхідно відзначити, що друге співвідношення (1) впливає з КРТ критерію міцності (критичного розкриття тріщини), який визначає гранично рівноважний стан пружнопластичного тіла з тріщиною на момент досягнення в її вершині розкриття  $\delta_{1C}$  і виконується для напруження  $\sigma/\sigma_T=1$  [1]. Методика розрахунку  $K_1$  для діапазону  $1 < \sigma/\sigma_T \leq \sigma_B/\sigma_T$  наведена у праці [7] і ґрунтується на використанні підходів механіки деформівного твердого тіла. Фізична величина  $WPL$  входить у відому формулу (критерій міцності) Гріфітса-Орована [1].

За результатами контактного деформування різних марок сталі (17ГС, 17ГС-У, 06Г2НАБ, 10Г2ФБ, 10Г2ФБ-У, 10ЧГНМАЮ) встановлено емпіричне співвідношення, яке зв'язує КІН  $K_1$  з  $WPL$  [8]:

$$K_1 = a_1 \cdot \sqrt{WPL} - a_2; \quad a_1 = 226 \cdot 10^6 \frac{\sqrt{Н}}{м}; \quad a_2 = 6,98 \text{ МПа} \cdot \sqrt{м}. \quad (2)$$

У співвідношенні (2) необхідно підставляти числові значення  $a_1=226, a_2=6,98, K_1$  виражати в  $\text{МПа} \cdot \sqrt{м}$ , тоді  $WPL$  отримаємо в  $\frac{\text{МДж}}{\text{м}^2}$ .

Оскільки сталь 20 за своїми механічними властивостями не значно відрізняється від тих марок сталі, для яких наведено апроксимацію (2) у праці [8], то припускаємо, що лінійна залежність (2) наближено виконується і для сталі 20.

Співвідношення для густини струму  $I_a$  у вершині тріщини [9,10] наступне:

$$I_a = \frac{\alpha \cdot \chi \cdot \Delta\psi_{ak}}{\delta \cdot \ln\left(\frac{h+c+r}{\delta}\right)}, \quad (3)$$

де  $\alpha$  – кут у вершині тріщини;  $\chi$  – електропровідність електроліту;  $\Delta\psi_{ak}$  – омична зміна електричного потенціалу між анодною і катодною частинами;  $h, c$  – глибина пітинга та тріщини,  $r$

– радіус кривизни проекції ювенільної поверхні (рис. 1). Співвідношення (3) записано для вершини тріщини – аноду А, тобто для ділянки, яка відповідає півколу  $\pi r$  на рисунку. Бокові відрізки тріщини ( $c$ ) і пітинга ( $h/\cos(\alpha/2)$ ) є проекціями катодної К області дефекту, яка покрита слабо провідними окислами (пасивними плівками). Аналогічний рисунок наведено у праці [11] та відзначено, що вказаний дефект (пітинг з тріщиною) заповнений корозійним середовищем і знаходиться на поверхні труби діаметра  $D$  з товщиною стінки  $d$  ( $h+c+r<d$ ).

В реальних умовах експлуатації елементів конструкцій, зокрема, в трубопроводах чи парогенераторах, необхідно враховувати явище стрес-корозії [2,3,12]. Тому співвідношення (3) доцільно узагальнити, тобто доповнити інформацією про зміни механічних параметрів та фізичних характеристик матеріалу. Метою роботи є дослідження впливу зовнішнього корозійного середовища та механічного навантаження на поверхневу енергію пластичного деформування і перенапруження реакції розчинення металу для ювенільної поверхні у вершині тріщини при механічному навантаженні об'єкту в межах від межі текучості матеріалу  $\sigma_T$  до межі міцності  $\sigma_B$  ( $\sigma/\sigma_T=1\div\sigma_B/\sigma_T$ ). Поставлена задача розглядається для аналізу умов деформування металу в корозійному середовищі з подальшим прогнозуванням ресурсу елементів конструкцій, які використовуються у магістральних трубопроводах та парогенераторах атомних електростанцій (АЕС).

**Основні допущення та рівняння.** У першому наближенні приймаємо до уваги емпіричне співвідношення між розкриттям тріщини  $\delta_{1C}$  і геометричним параметром  $\delta$ , який характеризує ширину ювенільної поверхні, а також відповідне значення емпіричної константи  $\xi_\delta$  [13]:

$$\delta_{1C} = \xi_\delta \cdot \delta; \quad \xi_\delta = 1. \quad (4)$$

для сталі 20 (гомогенного металу у твердому стані при температурі  $T=20^0C$  і атмосферному тиску  $p = 100$  кПа). Таким чином, у запропонованій для розгляду задачі  $\delta=15,1\div 51,8$  мкм (табл. 1). Довжина (фронт) тріщини  $L$  перпендикулярна до рис. 1 і в даному підході не розглядається та не враховується. В реальних ситуаціях може бути  $L \gg \delta$ .

Деякі фізико-механічні характеристики сталі 20 наступні [14,15]:

$$Z_{\text{ст}} = 2; \quad E = 213 \text{ ГПа}; \quad \sigma_T = 245 \text{ МПа}; \quad \sigma_B = 470 \text{ МПа}; \quad \nu = 0,3; \quad \rho = 7860 \text{ кг/м}^3 \quad (5)$$

З урахуванням другого співвідношення (1), числових даних (5) і  $\delta_{1C} = 15,1$  мкм (табл. 1) отримано:

$$K_{\text{ISCC}} = 28,1 \text{ МПа}\sqrt{\text{м}} \quad (6)$$

Для аналізу граничного (пластичного) стану матеріалу у вершині тріщини, який відповідає умові деформування циліндричної труби, використовуємо критерій пластичності Губера-Мізеса і відповідні вирази для механічних напружень  $\sigma_y, \sigma_z$  [11]:

$$\sigma_y^2 + \sigma_z^2 + \sigma_y \cdot \sigma_z = \frac{1}{3} \sigma_T^2, \quad \sigma_y = \frac{pD}{2d}; \quad \sigma_z = 9 \frac{pD}{2d}, \quad (7)$$

де  $\sigma_T$  – границя текучості матеріалу труби ( $x, y, z$  – прямокутні декартові координати);  $p$  – внутрішній тиск усередині труби. У співвідношеннях (7) для  $\sigma_y, \sigma_z$  враховано, що товщина стінки труби  $d$  значно менша діаметра  $D$ .

Для тіла (циліндричної труби), ослабленого дефектом типу пітинга з тріщиною у вершині, запишемо співвідношення для коефіцієнта концентрації напружень  $K_t$  і внутрішнього критичного тиску  $p = p_{cr}$ , аналогічні до тих, які подано у [11]:

$$K_t = \left( 2,021 - 1,301 \cdot \beta + 0,727 \cdot \beta^2 - 0,147 \cdot \beta^3 \right) \cdot \frac{d}{d_1 - c};$$

$$p_{cr} = \frac{2\sqrt{2} \cdot d \cdot \sigma_T}{3K_t \cdot D} \cdot \frac{(1,5 + K_z) \cdot (r_0 + c)^4}{(r_0 + c)^4 + 0,5 \cdot r_0^2 (r_0 + c)^2 + r_0^4}, \quad (8)$$

$$\text{де } K_z = \left( \frac{d_1 - c}{2} \cdot \frac{2(d_1 - c) + 3r_0}{d_1 - c + r_0} - \frac{3d}{2K_t} \right) / \left( \frac{d}{K_t} + \frac{r_0}{3} \cdot \left( \frac{r_0^3}{(d_1 - c + r_0)^3} - 1 \right) \right); \beta = \frac{r_0}{d_1 - c};$$

критичний тиск  $p_{cr}$  (8) відповідає умові досягнення у вершині тріщини граничного (пластичного) стану згідно з критерієм Губера-Мізеса (7). Вираз для  $p_{cr}$  отримано аналітично із розв'язку задачі механіки деформівного твердого тіла [11].

Щоб оцінити фізичні величини у (8), використано наступні числові дані для циліндричної труби:

$$\begin{aligned} d &= 16 \text{ мм}; D = 0,71 \text{ м}; d_1 = 15,8 \text{ мм}; \\ h &= d - d_1 = 0,2 \text{ мм}; c = c_0 = 0,5 \text{ мм}; r_0 = 7,5 \text{ мкм}. \end{aligned} \quad (9)$$

Тут  $c_0, h$  відповідають відносному механічному напруженню  $\sigma/\sigma_T = 1$  (табл. 1).

Необхідно відзначити, що концентратор напружень (пітинг) з короткою корозійною тріщиною ( $c_0 = 0,5$  мм) – це система, де активна зона (окіл вершини тріщини) знаходиться на малій відстані від поверхні пітинга [2]. Тому проходить достатньо легка доставка продуктів розчинення металу (іонів заліза) з активної зони в загальний об'єм розчину електроліту [2]. При збільшенні довжини  $c$  тріщини доставка розчинених внаслідок корозії іонів заліза з активної зони в загальний об'єм розчину електроліту сповільнюється [2,12].

Критеріальним співвідношенням щодо оцінювання ресурсу трубопроводу буде задане значення внутрішнього тиску  $p_{cr}$ , яке знаходиться в межах  $5,0 \div 5,5$  МПа і повинно відповідати нормативним документам [15]. Для прикладу вибираємо:

$$p_{cr} = 5,0 \text{ МПа}. \quad (10)$$

Використовуючи (8)–(10) і дані в табл. 1, числовим методом послідовно визначаємо глибину тріщини  $c$  для кожної пари значень  $\delta_{1C}, \sigma/\sigma_T$  (табл. 2).

**Таблиця 2. Глибина тріщини  $c$  для сталі 20 в корозійному середовищі при механічних розтягуючих напруженнях  $\sigma/\sigma_T$**

Глибина тріщини $c$ , мм	0,5	1,6	2,66	3,55
Відносне механічне напруження $\sigma/\sigma_T$	1,0	1,082	1,164	1,246

При цьому вважаємо, що розміри каверни змінюються незначно, тобто змінами параметра  $h$  нехтуємо.

Відносна зміна густини струму  $I_a^*/I_{a0}$  для сталі 20 у корозійному середовищі (в глинистих ґрунтах з помірною вологістю) отримана експериментально для трубопроводу в залежності від терміну експлуатації [16], відповідає рівномірній корозії на зовнішній поверхні, перерахована на глибину тріщини  $c$ , оскільки  $c$  змінюється з часом, і подана у таблиці 3:

**Таблиця 3. Відносна зміна густини струму  $I_a^*/I_{a0}$  для сталі 20 у корозійному середовищі при різних значеннях глибини  $c$  тріщини**

Відносна зміна густини корозійного струму $I_a^*/I_{a0}$	1,0	0,429	0,229	0,171
Глибина тріщини $c$ , мм	0,5	1,6	2,66	3,55

З (1), (2) для кожного набору ( $\sigma/\sigma_T, \delta_{1C}, c$ ) з урахуванням (4)–(10) і даних в табл. 1-3 числовим методом визначено поверхневу енергію пластичного деформування WPL і електрохімічне перенапруження  $\zeta$ . (рис. 2).

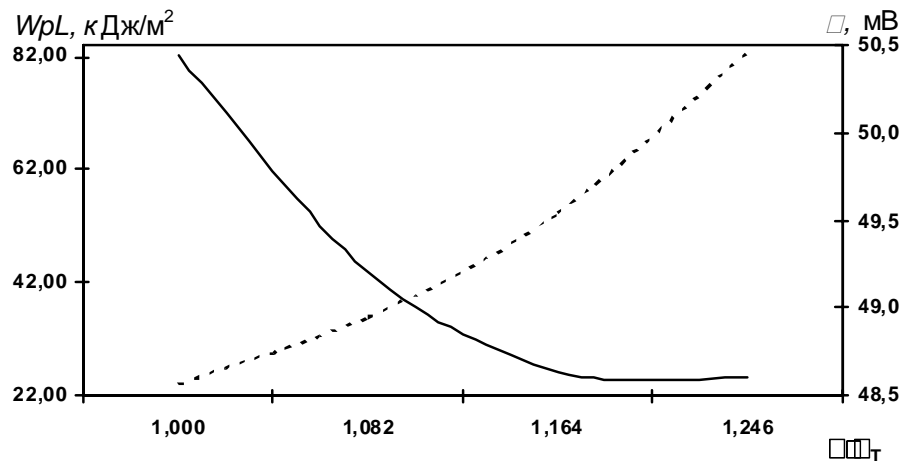


Рис. 2. Залежності поверхневої енергії пластичного деформування  $W_{pL}(\sigma/\sigma_T)$  (кДж/м<sup>2</sup>) (пунктирна лінія) і перенапруження  $\zeta(\sigma/\sigma_T)$  (мВ) (суцільна лінія) у діапазоні механічних розтягуючих напружень  $\sigma/\sigma_T=1\div 1,246$  для сталі 20, що знаходиться в 3 % розчині NaCl.

Густину струму  $I_a$  в діапазоні  $\sigma/\sigma_T=1\div 1,246$  апроксимуємо аналітичною залежністю:

$$I_a = I_{as} \cdot \left( 1 + \beta_w \cdot \left( \frac{W_{pL} - W_{pL0}}{W_{pL0}} \right)^S \right), \quad (11)$$

де  $I_{as} = 0,275 \frac{мм}{рік} = 0,237 \frac{А}{м^2}$ ;  $\beta_w = 0,3669$ ;  $S = 0,4406$ ;  $W_{pL0} = 24,1 \frac{кДж}{м^2}$  ( $\zeta = 50,4\div 48,6$  мВ;

$W_{pL} = 24,1\div 82,75 \frac{кДж}{м^2}$ ). В результаті розрахунків встановлено, що непевність (похибка) отриманих числових значень  $\beta_w, S$  не перевищує 5 %.

З урахуванням апроксимації (11), яка містить скалярну енергетичну характеристику поверхневого шару  $W_{pL}$ , узагальнимо відоме співвідношення (3) для густини струму у вершині тріщини:

$$I_a = \frac{\alpha \cdot \chi \cdot \Delta \psi_{ak}}{\delta \cdot \ln \left( \frac{h+c+r}{\delta} \right)} \cdot \left( 1 + \beta_w \cdot \left( \frac{W_{pL} - W_{pL0}}{W_{pL0}} \right)^S \right). \quad (12)$$

З табл. 2 і (9) отримано  $W_{cd} = \frac{c}{d} = \frac{3,55}{16} = 0,222$ . Відповідна відносна глибина тріщини згідно з нормативними документами [15] вважається критичною ( $(W_{cd})_{cr}=20\div 22$  %). Тут враховується запас міцності.

Для оцінювання ресурсу трубопроводу потрібно визначити час проходження тріщиною максимальної допустимої віддалі  $s=3,55$  мм (табл. 2). Якщо б початкова густина корозійного струму  $I_{as} = 0,275 \frac{мм}{рік}$  не змінювалась, то ресурс трубопроводу становив би 12,9 років. Але при цьому слід врахувати два реальні механізми зміни густини корозійного струму. Перший механізм характеризується зростанням густини струму корозії і пов'язаний з поверхневою енергією пластичного деформування (12) (табл. 1, рис. 2). Другий механізм супроводжується зменшенням густини струму корозії і його характер встановлено експериментально (табл. 3) [16]. Побудовано відповідний алгоритм, який враховує узагальнене співвідношення (12), два механізми змін густини струму анодного розчинення (корозії) і числовим методом отримано час росту корозійної тріщини, тобто ресурс трубопроводу:

T=18,7 роки,

який відповідає внутрішньому тиску  $p_{cr}=5$  МПа і відносній глибині тріщини  $W_{cd}=0,222$ . При цьому крок обчислювальної процедури вибрано 1 рік ( $\Delta T=1$  рік).

В корозійному процесі для трубопроводу із сталі 20 в 3 % розчині NaCl при  $\sigma/\sigma_T=1\div 1,246$  зміни фізичних величин WPL,  $\zeta$  такі (рис. 2):

$$\frac{(WPL)_{\max}}{WPL0} = \frac{82,75}{24,1} = 3,52; \quad \frac{(\zeta)_{\max} - (\zeta)_{\min}}{(\zeta)_{\max}} = \frac{50,4 - 48,6}{50,4} = 0,036.$$

**Висновки та перспективи подальших досліджень.** На основі підходів механіки деформівного твердого тіла, фізики поверхневих явищ та електрохімії сформульовано співвідношення математичної моделі для оцінювання поверхневої енергії пластичного деформування та перенапруження реакції розчинення металу у вершині тріщини для навантаженого металу, який знаходиться у водному розчині електроліту. Дана модель враховує коефіцієнт інтенсивності напружень (K<sub>1</sub>), коефіцієнт концентрації напружень, поверхневу енергію пластичного деформування і стосується діапазону механічних навантажень металу від межі текучості матеріалу до межі, яка встановлена нормативними документами для елемента конструкції.

Проведено оцінювання значення K<sub>1</sub> для напруження, яке відповідає границі текучості матеріалу (металу). Це значення  $K_{1SCC}=28,07$  МПа· $\sqrt{м}$  можна вважати граничною умовою і тестовим варіантом для визначення K<sub>1</sub> в діапазоні  $\sigma/\sigma_T=1\div\sigma_B/\sigma_T$ .

На основі аналізу залежностей між поверхневою енергією пластичного деформування (WPL) і навантаженням, що діє на трубопровід з пітингом, у вершині якого знаходиться тріщина, встановлено, що в діапазоні зміни розтягуючих механічних напружень від межі текучості  $\sigma_T$  до  $1,246\cdot\sigma_T$  (межі, встановленої нормативними документами) для сталі 20 у 3 % розчині NaCl WPL зростає в 3,53 рази, а перенапруження  $\zeta$  електродної реакції (розчинення заліза) зменшується на 3,6 %.

Узагальнено співвідношення густини струму у вершині тріщини для навантаженого металу (відбувається розтяг у напрямках осей y і z) з урахуванням скалярної енергетичної характеристики – поверхневої енергії пластичного деформування WPL.

Відповідно до запропонованої математичної моделі розроблено та реалізовано алгоритм числового розрахунку параметрів задачі, етапи якого наступні:

- визначення глибини тріщини  $c$  для пари значень розкриття  $\delta_{1C}$  та відносного механічного напруження  $\sigma/\sigma_T$  з урахуванням критерію пластичності Губера-Мізеса і заданого значення тиску всередині труби;

- розрахунок поверхневої енергії пластичного деформування WPL та електрохімічного перенапруження  $\zeta$  для заданої низки значень: глибини тріщини  $c$ ,  $\delta_{1C}$ ,  $\sigma/\sigma_T$ ;

- з урахуванням узагальненого співвідношення (12), яке зв'язує густину корозійного струму  $I_a$  з шириною ювенільної поверхні і поверхневою енергією пластичного деформування WPL, двох механізмів зміни густини струму анодного розчинення (корозії), числовим методом розраховано час росту корозійної тріщини в навантаженій внутрішнім тиском трубі, тобто проведено оцінювання ресурсу трубопроводу.

Необхідно відзначити, що в даній праці вперше встановлено взаємозв'язок між такими фізичними величинами як внутрішній тиск  $p_{cr}$  трубопроводу, поверхнева енергія WPL пластичного деформування, коефіцієнти інтенсивності K<sub>1</sub> та концентрації K<sub>c</sub> напружень, густина струму  $I_a$  та перенапруження  $\zeta$  електродної реакції розчинення металу у вершині тріщини, ресурс трубопроводу і розроблено алгоритм оцінювання їх відносних змін з урахуванням критерію пластичності Губера Мізеса у вершині заповненої розчином електроліту тріщини, яка знаходиться на зовнішній поверхні труби.

Дослідження енергетичних та електрохімічних параметрів напружено-деформованого стану металевих елементів конструкцій і їх впливу на інтенсивність корозійних процесів у дефектах допоможуть оцінити час поширення корозійної тріщини і, відповідно, на основі катодного захисту оптимізувати заходи, які дозволять продовжити ресурс трубопроводу чи паропроводу атомної електростанції.



1. *Панасюк В. В.* Основы механики разрушения / *В. В. Панасюк, А. Е. Андрейкив, В. З. Партон.* – К.: Наук. думка, 1988. – 488 с.
2. *Дмитрах І. М.* Вплив корозійних середовищ на локальне руйнування металів біля концентраторів напружень / *І. М. Дмитрах, В. В. Панасюк.* – Львів: ФМІ, 1999. – 341 с.
3. *Карзов Г. П.* Зарождение и развитие коррозионного растрескивания в металле теплообменных труб парогенераторов типа ПГВ-1000 в стояночных и нестационарных режимах эксплуатации / *Г. П. Карзов, С. А. Суворов, В. А. Федорова* // Материалы 8-го международного семинара по горизонтальным парогенераторам. – Подольск (Россия): ОКБ "Гидропресс", 2010. – С. 1-11.
4. *Джала Р.* Параметри та критерії для дослідження руйнування сталі в сірководневих середовищах / *Р. Джала, В. Юзевич* // Вимірювальна техніка та метрологія. – 2012. – № 73. – Р. 148-151.
5. Новый справочник химика и технолога. Электродные процессы. Химическая кинетика и диффузия. Коллоидная химия // Редактор *С. А. Симанова* – Санкт-Петербург: АНО НПО "Профессионал", 2004. – 838 с.
6. *Антропов Л. И.* Теоретическая электрохимия. Учеб. для хим.-технолог. спец. вузов / *Л. И. Антропов.* 4-е изд., перераб. и доп. – М.: Высш. шк., 1984. – 519 с.
7. *Покровский В. В.* Прогнозирование влияния предварительного термомеханического нагружения на повышение сопротивления хрупкому разрушению конструкционных теплоустойчивых сталей с трещинами. Сообщение 1. Модель и методика расчета эффекта предварительного термомеханического нагружения / *В. В. Покровский, А. Г. Иванченко* // Проблемы прочности. – 2002. – № 6. – С. 96-106
8. *Бахрачева Ю. С.* Оценка вязкости разрушения сталей по результатам контактного деформирования / *Ю. С. Бахрачева* // Вестник Волгоградского государственного университета. Серия 10: Инновационная деятельность. – 2012. – Выпуск № 7. – С. 53-56.
9. *Kaeshe H.* Die Korrosion der Metalle. Physikalisch-chemische Prinzipien und aktuelle Probleme / *H. Kaeshe.* – 1979. – Berlin-Heidenberg-New York: Springer-Verlag. – 400 p.
10. *Панасюк В. В.* Механика квазихрупкого разрушения / *В. В. Панасюк.* – К.: Наукова думка, 1991. – 416 с.
11. *Джала Р.* Оцінювання параметрів напруженого стану металу трубопроводу з корозійною каверною / *Р. Джала, В. Юзевич* // Матеріали 5-ї Міжнародної конференції «Механіка руйнування матеріалів і міцність конструкцій». – Львів: 24-27 червня 2014 р. – С. 675-680.
12. *Крижанівський Є. І.* Особливості стрес-корозії сталі магістральних трубопроводів у кислих електролітах / *Є. І. Крижанівський, Л. Я. Побережний* // Розвідка та розробка нафтових і газових родовищ. – 2004. – № 4 (13). – С. 70-74.
13. *Сакара А.* Розрахункова модель докритичного росту корозійно-механічних тріщин у металевих пластинах / *А. Сакара, Ю. Банахевич* // Фізико-хімічна механіка матеріалів. – Львів: Фізико-механічний інститут НАН України, 2010. – № 3. – С. 23-29.
14. Таблицы физических величин. Справочник / Под ред. *И. К. Кикоина.* – М.: Атомиздат, 1976. – 1008 с.
15. *Анучкин М. П.* Трубы для магистральных трубопроводов / *М. П. Анучкин, В. Н. Горицкий, Б. И. Мирошниченко.* – М.: Недра, 1986. – 231 с.
16. *Ждек А. Я.* Визначення залишкового ресурсу тривало експлуатованих нафтопроводів із врахуванням наявних корозійних дефектів та умов експлуатації / *А. Я. Ждек, В. Я. Грудз* // Науковий вісник ІФНТУНГ. – 2012. – № 2 (32). – С. 134-142.

УДК 621.316.7

Гадай А. В., к.т.н.

Луцький національний технічний університет

## МОДЕЛЮВАННЯ РОБОТИ СИНХРОННОГО ДВИГУНА ДЛЯ ЗМЕНШЕННЯ ВІДХИЛЕНЬ НАПРУГИ В СЕРЕДОВИЩІ MATLAB/SIMULINK

**Гадай А. В. Моделювання роботи синхронного двигуна для зменшення відхилень напруги в середовищі MATLAB/SIMULINK.** В статті розглядається робота синхронного двигуна в складі бурової установки, яка живиться від шин обмеженої потужності. Подано математичну модель двигуна та досліджений вплив регулювання збудження на роботу мережі. Показано, що форсування збудження синхронного двигуна дає можливість зменшити зниження напруги за рахунок динамічної компенсації реактивної потужності.

**Ключові слова:** напруга, синхронний двигун, автоматичне регулювання збудження, форсування збудження, динамічна компенсація реактивної потужності.

Форм. 3. Рис. 6. Літ. 7.

**Гадай А. В. Моделирование работы синхронного двигателя для уменьшения отклонений напряжения в среде MATLAB/SIMULINK.** В статье рассматривается работа синхронного двигателя в составе буровой установки, которая питается от шин ограниченной мощности. Подано математическую модель двигателя и исследовано влияние регулирования возбуждения на работу сети. Показано, что форсирование возбуждения синхронного двигателя позволяет уменьшить снижение напряжения за счет динамической компенсации реактивной мощности.

**Ключевые слова:** напряжение, синхронный двигатель, автоматическое регулирование возбуждения, форсировка возбуждения, динамическая компенсация реактивной мощности.

**Gadai A. Simulation of synchronous motor for reducing voltage deviations in MATLAB / SIMULINK.** The article reviews the work of synchronous motor as part of rig, which is powered by the bus limited power. Submitted a mathematical model motor and investigated the effect of excitation control on the network. Shown that forcing excitation synchronous motor can reduce the voltage drop through the dynamic reactive power compensation.

**Keywords:** voltage, synchronous motor, automatic regulation the excitation, forced excitation, dynamic reactive power compensation.

**Постановка проблеми.** У випадку відхилення рівня напруги від оптимальної величини сумарні втрати, які визначаються технологічними та електромагнітними збитками, збільшуються [1]. Технологічні збитки враховують збитки від порушення технологічного процесу, погіршення якості продукції, зменшення продуктивності праці, Електромагнітні збитки характеризуються збільшенням втрат електроенергії, виходом з ладу електротехнічного обладнання, порушенням роботи автоматики.

**Аналіз останніх досліджень і публікацій.** Вибір критерію оптимальності є важливою і складною задачею, а використання необґрунтованих критеріїв часто призводить до прийняття рішень, які не забезпечують бажаних техніко-економічних показників роботи електроприводів, включаючи і надійність. Необхідно відмітити, що загального, єдиного підходу до вибору критеріїв оптимізації показників якості електроенергії для крупних електроприводів, які працюють в електричних мережах обмеженої потужності немає. В роботах [2, 3] стверджується, що оптимальними критеріями регулювання струму збудження синхронних двигунів (СД) є економічність роботи самої машини і найкраще співвідношення її електромеханічних показників. Однак такий підхід є неповним. Система автоматичного регулювання збудження (АРЗ) СД повинна враховувати технологічні особливості режимів роботи СД і найповніше використання їх компенсувальної здатності, яка дозволяє суттєво поліпшити якість напруги живлення у вузлі електричної мережі. АРЗ СД повинен забезпечувати мінімум втрат електроенергії в електричній мережі.

**Невирішені частини проблеми.** Пристрої автоматичного регулювання збудження (АРЗ) синхронних двигунів (СД) виконуються так, щоб протягом кожного технологічного режиму напруга у вузлі навантаження підтримувалась оптимальною. Існуючі АРЗ не дозволяють прогнозувати або керувати з випередженням режимами напруги і показниками якості електроенергії. При зміні навантаження двигуна неможливо встановити оптимальний струм збудження. Його необхідно автоматично регулювати в залежності від конкретних умов, які характеризують електропостачальну систему і навантаження СД. Вибір закону АРЗ повинен виконуватись з врахуванням можливих змін рівня напруги.

**Метою дослідження** є визначення можливості використання СД для зменшення різких і глибоких знижень напруги за рахунок динамічної компенсації реактивної потужності.

**Основні результати дослідження.** Методи дослідження режимів роботи потужних синхронних електроприводів, які працюють в електричних мережах обмеженої потужності, базуються на використанні аналітичних залежностей і математичних моделей для ЕОМ. Загальноприйнятим при вирішенні таких задач є розділення дослідження на аналіз квазіусталених режимів, аналіз перехідних процесів при пуску двигунів та їх динамічної стійкості.

Системи рівнянь можуть бути подані в рухомих або нерухомих системах координат. Головна складність чисельного аналізу з використанням нерухомих системи координат при розрахунку перехідних процесів СД пов'язана з необхідністю обертання на кожному інтервалі матриці коефіцієнтів високого порядку, який визначається кількістю розрахункових контурів. При цьому вимагається великий обсяг обчислень і відповідний йому машинний час.

При вивченні квазіусталених режимів СД використовують рівняння Парка-Горєва в  $d, q$ -координатах, що записані через діючі значення величин [1]. Систему рівнянь, що характеризує електромагнітні перехідні процеси СД подамо у матричному вигляді

$$\begin{bmatrix} L_d & 0 & M_{sf} & M_{sD} & 0 \\ 0 & L_q & 0 & 0 & M_{sQ} \\ 1,5M_{sf} & 0 & L_f & M_{fD} & 0 \\ 1,5M_{sD} & 0 & M_{fD} & L_D & 0 \\ 0 & 1,5M_{sQ} & 0 & 0 & L_Q \end{bmatrix} \begin{bmatrix} \frac{dI_d}{dt} \\ \frac{dI_q}{dt} \\ \frac{dI_f}{dt} \\ \frac{dI_D}{dt} \\ \frac{dI_Q}{dt} \end{bmatrix} = \begin{bmatrix} -U_d \\ -U_q \\ U_f \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} R_s & \omega L_q & 0 & 0 & \omega M_{sQ} \\ -\omega L_d & R_s & -\omega M_{sf} & -\omega M_{sD} & 0 \\ 0 & 0 & R_f & 0 & 0 \\ 0 & 0 & 0 & R_D & 0 \\ 0 & 0 & 0 & 0 & R_Q \end{bmatrix} \begin{bmatrix} I_d \\ I_q \\ I_f \\ I_D \\ I_Q \end{bmatrix}, \quad (1)$$

де  $R_s, R_f, R_D, R_Q$  – активні опори відповідно обмотки статора, обмотки збудження та демпферних обмоток по осях  $D, Q$ ;

$L_d, L_q$  – індуктивності статорної обмотки відповідно по повздовжній і поперечній осях;

$M_{sf}, M_{sD}, M_{sQ}, M_{fD}$  – взаємні індуктивності між відповідними обмотками СД;

$U_d, U_q, U_f$  – напруги відповідно обмотки статора по осях  $d, q$  та обмотки збудження.

Систему рівнянь (1) можна записати:

$$\begin{aligned} \frac{dI_d}{dt} &= \alpha_{11}(-U \sin \theta + p\omega_r L_q I_q - R_s I_d) + \alpha_{12}(U_f - R_f I_f); \\ \frac{dI_q}{dt} &= \alpha_q(E - U \cos \theta - p\omega_r L_d I_d - R_s I_q); \\ \frac{dI_f}{dt} &= \alpha_{12}(-U \sin \theta + p\omega_r L_q I_q - R_s I_d) + \alpha_{22}(U_f - R_f I_f), \end{aligned} \quad (2)$$

де  $\alpha_{11} = L_f / (L_f L_d - M_{sf}^2)$ ;  $\alpha_{12} = M_{sf} / (L_f L_d - M_{sf}^2)$ ;

$\alpha_{22} = L_d / (L_f L_d - M_{sf}^2)$  – обернені індуктивності, що характеризують зв'язок між струмами та ЕРС обмотки збудження та обмотки якоря по повздовжній осі;

$\alpha_q = 1 / L_q$  – обернена індуктивність статора по поперечній осі.

Рівняння (1) або (2) доповнюються рівняннями, що характеризують електромеханічні процеси

$$\frac{d\omega_r}{dt} = \frac{-\frac{P(t)}{\omega_r} - M_n(t)}{J}; \quad \frac{d\theta}{dt} = \omega_r - \frac{\omega}{p_0}. \quad (3)$$

Живлення бурильної установки здійснюється за схемою, наведеною на рис.1,а. Схема досліджуваної мережі складається з джерела живлення, лінії та вузла навантаження, до складу якого входить синхронний двигун приводу лебідки. На рис.1,б наведена заступна схема мережі, на основі якої складена схема для моделювання рис.2.

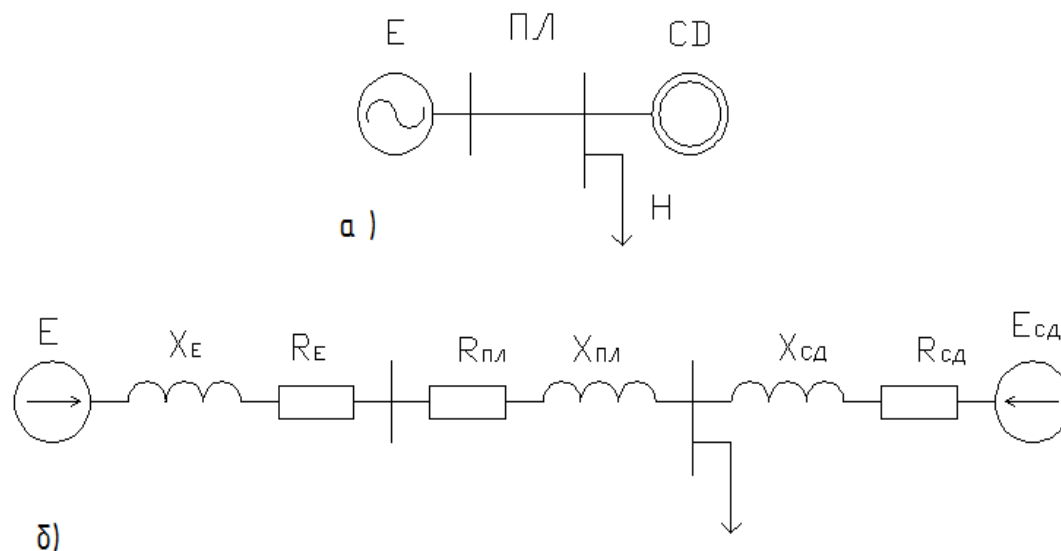


Рис. 1. Схема живлення бурової установки з СД

Електропривод лебідки повинен піднімати і опускати бурильну колону із заданою швидкістю при змінній масі колони. Для оптимального використання електроприводу [5] необхідно підтримувати постійну потужність в усьому діапазоні зміни маси бурильної колони під час піднімання і опускання. У випадку живлення бурової установки від системи електропостачання обмеженої потужності, активна та реактивна потужності СД сильно змінюються [6] і навантаження СД лебідки має ударний характер. Це призводить до відхилення і коливання напруги на ввіді бурової установки і коливання СД. В наслідок цього виникають аварії і передчасний вихід з ладу двигуна [7].

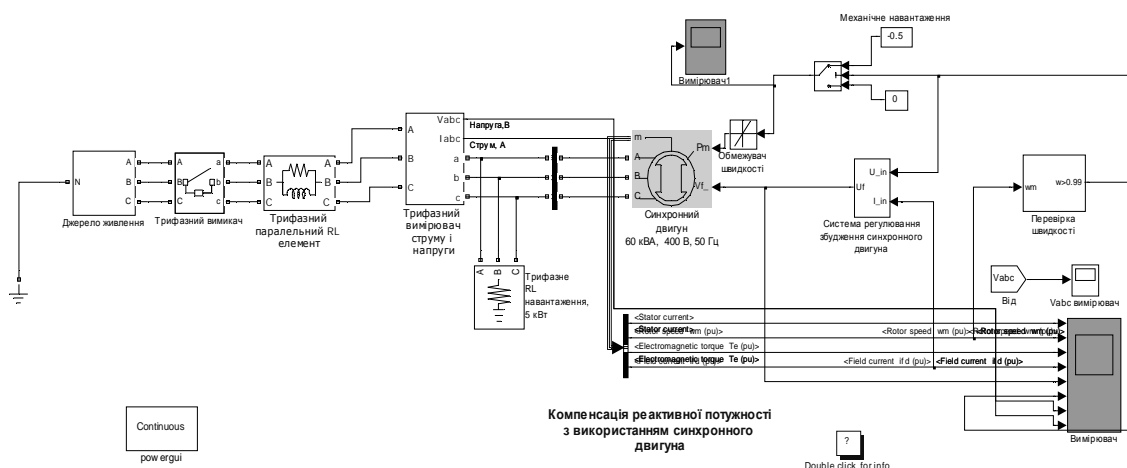


Рис. 2. Модель системи живлення бурової установки з СД

На бурових установках для приводу лебідки використовуються СД типу СДЗБ-13 потужністю 450–630 кВт. Система збудження таких двигунів забезпечує форсування збудження, при цьому напруга й струм збудження будуть максимальними, а час досягнення максимального струму збудження – мінімальним.

У разі форсування струму збудження (рис. 3) СД мають здатність різкої зміни реактивної потужності (рис. 4). У даному випадку струм збудження форсується з 267 А до 330 А. При цьому реактивна потужність номінально завантаженого СД зменшується з 260 квар до 15 квар.

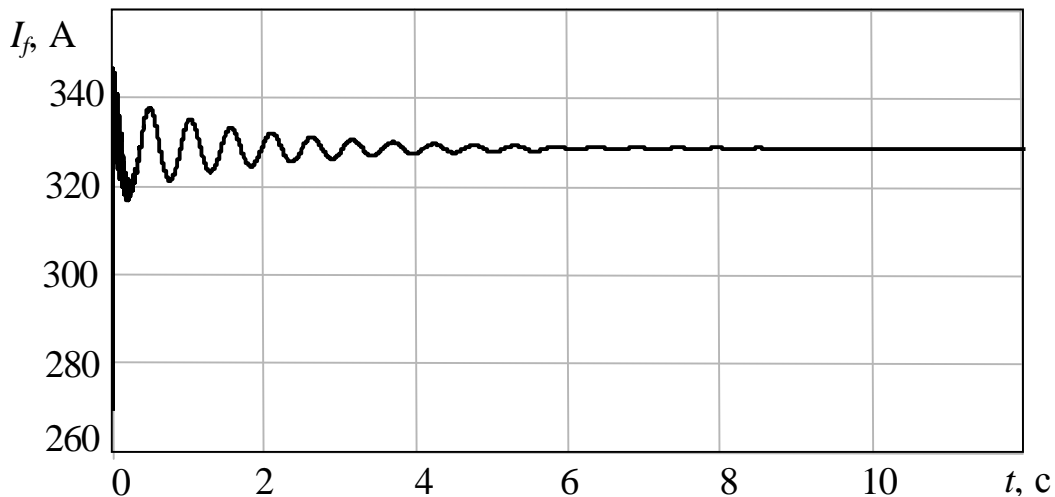


Рис. 3. Форсування струму збудження СД

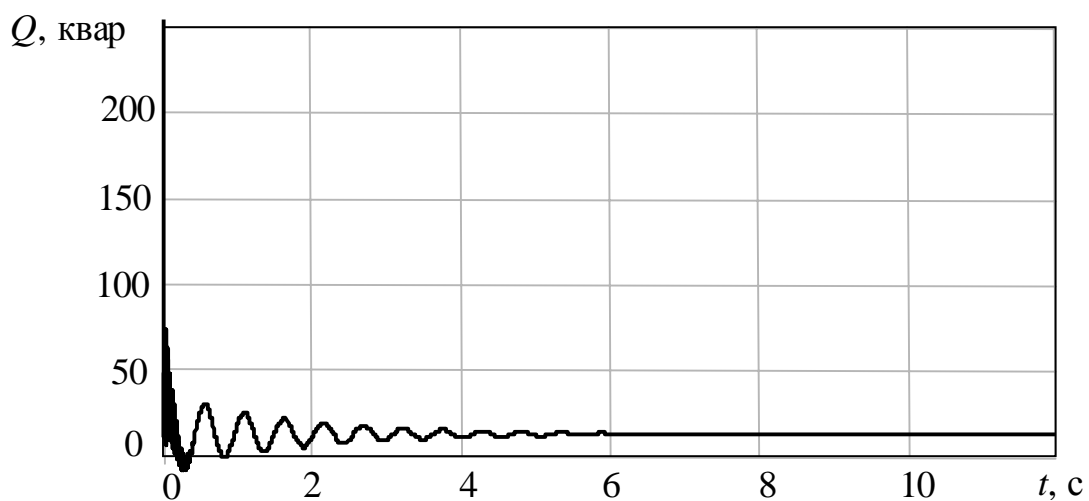


Рис. 4. Зменшення споживання реактивної потужності СД у разі форсування збудження

Під час форсування струму збудження різко змінюється активна потужність (рис. 5), проте зниження напруг зменшується (рис. 6). залежності амплітуди фазної напруги побудовані якщо активний та реактивний опори лінії живлення становлять відповідно 2 Ом і 8 Ом.

**Висновки.** Аналізуючи вище викладене, можна констатувати, що форсування збудження СД забезпечує можливість зменшити зниження напруги за рахунок динамічної компенсації реактивної потужності. Однак при цьому виникають коливання активної потужності, що вимагає їх демпфірування.

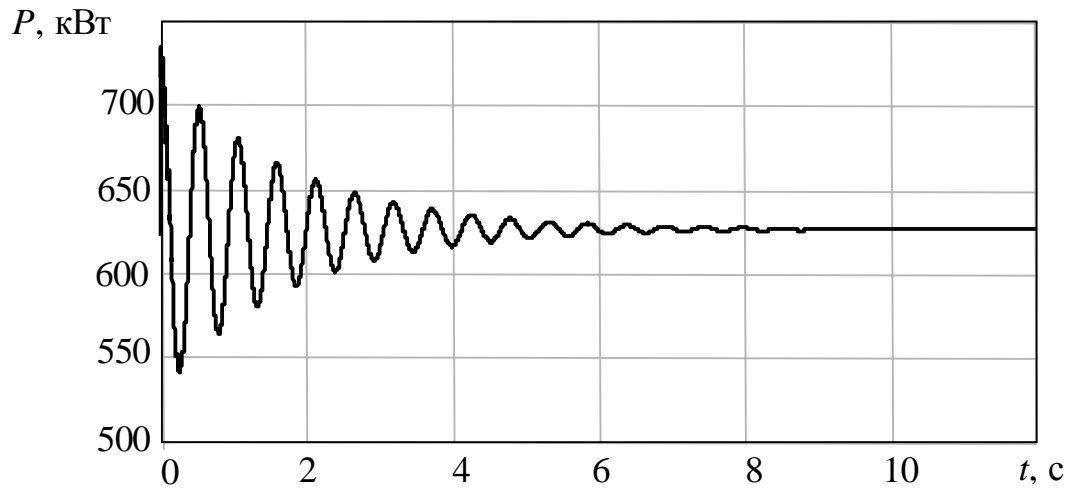


Рис. 5. Залежність активної потужності СД у разі форсування збудження

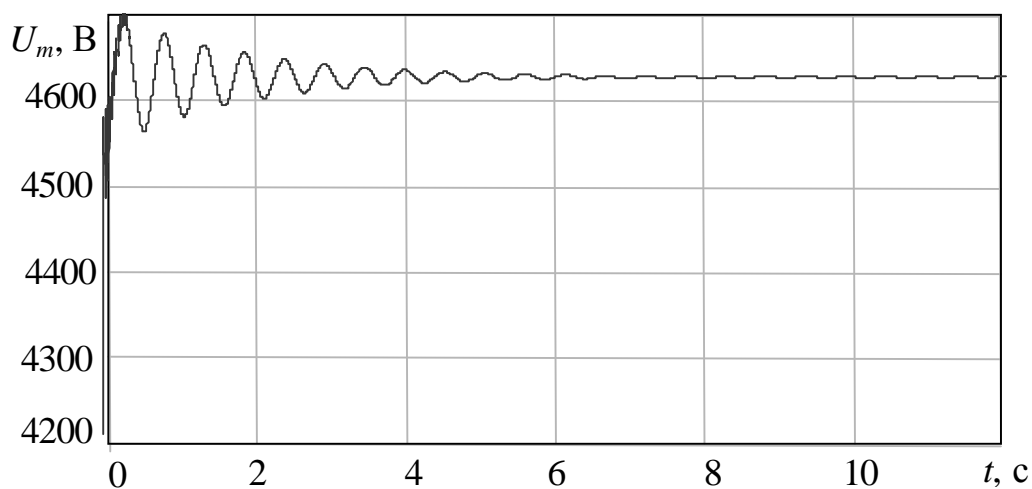


Рис. 6. Залежність амплітуди фазної напруги у разі форсування збудження СД

1. Суднова В.В. Качество электрической энергии / В.В. Суднова. – М.: ЗАО Энергосервис, 2000. – 80 с.
2. Красник В. В. Автоматические устройства по компенсации реактивных нагрузок в электрических сетях предприятий / В. В. Красник. – М.: Энергия, 1975. – 113 с.
3. Айрапетян Г. А. Расчет на ЭЦВМ режима работы синхронного двигателя при изменении различных параметров режима / Г. А. Айрапетян, Г. А. Налчаджян. – В кн.: Научн. тр. – Ереван.политехн.ин-т, 1972. – №28. – С.152–159.
4. Костюк О. М. Колебания и устойчивость синхронных машин / О. М. Костюк, М. И. Соломаха. – К.: Наукова думка, 1991. – 200 с.
5. Моцохейн Б.И., Парфенов Б.М. Электропривод буровых лебедок. / Б.И. Моцохейн, Б.М. Парфенов. – М.: Недра, 1978. – 304 с.
6. Исследование автоматического регулятора возбуждения приводного синхронного двигателя буровой лебедки. Машины и нефтяное оборудование. / И.И.Саяк, Р.А.Селепина, Г.С.Корниевич, В.Н.Холин. – М.: ВНИИОЭНГ, 1975, № 8, с.30-35.
7. Абрамович Б.Н., Купцов А.Б., Лозовой И.Ф. Бесщеточныя синхронные двигатели с повышенными эксплуатационными характеристиками для главных приводов буровых установок. Машины и нефтяное оборудование. / Б.Н. Абрамович, А.Б. Купцов, И.Ф. Лозовой – М.: ЕНИИОЭНГ, 1982, № I, с.26-28.

УДК 517.977

Семенюк В.Я., Міскевич О.І., Горошко Д.О., Машевський М.В.  
Луцький національний технічний університет, Луцьк

## ОПТИМІЗАЦІЯ ПЕРЕМІЩЕНЬ ТОВАРІВ ПРИ ПРОЕКТУВАННІ ВИРОБНИЧИХ СИСТЕМ

**Семенюк В.Я., Міскевич О.І., Горошко Д.О., Машевський М.В. Оптимізація переміщень товарів при проектуванні виробничих систем.** На основі методу віток і меж побудовано систему відшукування оптимального плану розміщення виробничих ліній, яка мінімізує транспортні витрати на переміщення матеріалів для виробництва та виготовленої продукції. Розглядається склад і 6 виробничих ліній. Розроблено алгоритм та реалізовано обчислювальну процедуру щодо розрахунку оптимального плану розміщення виробничих ліній і складу. Проведено оцінювання витрат часу на транспортування між складом і виробничими лініями.

**Ключові слова:** метод віток і меж, оптимізація транспортно-складської системи, алгоритм знаходження мінімальних і максимальних оцінок часу.

**Семенюк В.Я., Міскевич О.І., Горошко Д.О., Машевський М.В. Оптимізація перевезення товарів при проектуванні виробничих систем.** Используя метод ветвей построено систему поиска оптимизированого плана расстановки производственных линий, которая минимизирует транспортные расходы на перевозку материалов для производства и товаров. Рассматриваем склад и 6 производственных линий. Разработан алгоритм и вычислительная процедура поиска оптимального плана размещения производственных линий и склада. Сделана оценка использования времени на транспортировку между складом и производственными линиями.

**Ключові слова:** метод ветвей, оптимізація транспортно-складської системи, алгоритм пошуку мінімальних і максимальних оцінок часу.

**Semenyuk V., Miskevych O., Horoshko D., Mashevskij M. Optimization of products displacement on industrial systems designing.** Based on branch and bound algorithm design there was built a system of finding the optimal plan of production lines placement, that minimized transportation costs of moving manufacture materials and produced products. There were examined warehouse and 6 production lines. There also have been designed an algorithm and implemented a computing procedure for calculation of optimal plan for production lines and warehouse placement. In conclusion, there was made the time-consuming evaluation for transportation between warehouse and production lines.

**Keywords:** branch and bound algorithm design, optimization of transportation and warehousing system, algorithm of finding the minimal and maximal time estimations.

### Постановка наукової проблеми.

Розглядаємо гнучку виробничу систему, при проектуванні якої необхідно розташувати виробничі обладнання так, щоб сума транспортних перевезень була мінімальною. Це дозволить скоротити кількість транспортних засобів необхідних для перевезення, а також мінімізувати транспортні витрати.

На графічному представленні (рис. 1) розміщено склад та  $2n$  однакових приміщень під виробничі лінії. За відстань  $l(X_i, X_j)$  приймем найкоротший шлях між  $X_i$  та  $X_j$ . Матриця відстаней між вершинами і складом буде симетричною. Кругами на рис. 1 позначені зони розвертання. Щоб не ускладнювати задачу конкретними вимірами будемо вважати, що відстані між будь-якими двома зонами розвертання однакові і рівні 1.

Будемо вважати, що технологія обробки виробів задана. Це дозволяє скласти матрицю інтенсивностей передачі виробів між різними одиницями

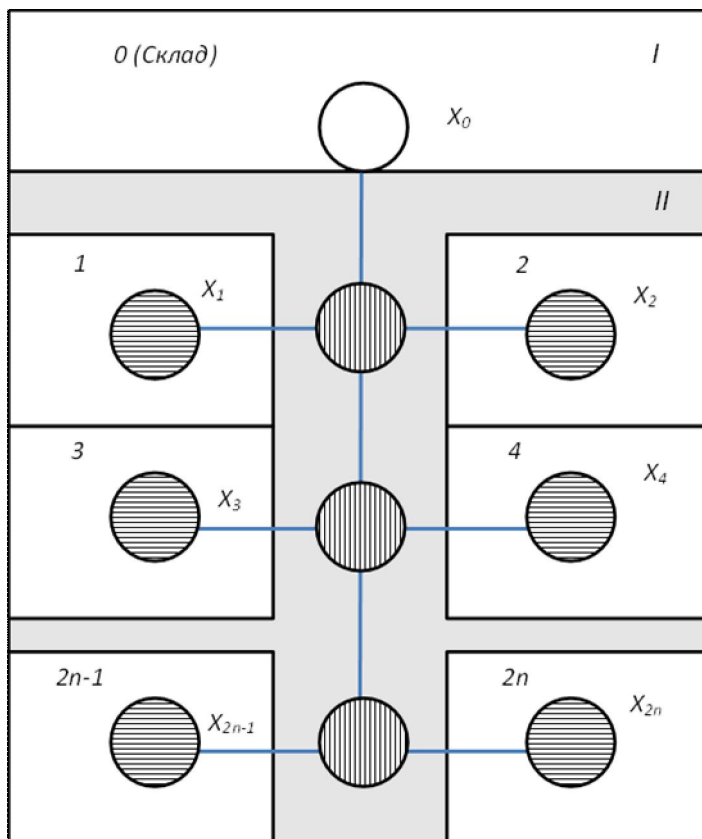


Рис. 1. Графічне представлення виробничої системи.

обладнання  $A = \|\alpha_{i,j}\|$ , елементами якої є інтенсивності передачі виробів від  $i$ -ої одиниці технологічного обладнання до  $j$ -ої.

$$L = \begin{pmatrix} 0 & 2 & 2 & 3 & 3 & K & n+1 & n+1 \\ 2 & 0 & 2 & 3 & 3 & K & n+1 & n+1 \\ 2 & 2 & 0 & 3 & 3 & K & n+1 & n+1 \\ 3 & 3 & 3 & 0 & 2 & K & n & n \\ 3 & 3 & 3 & 2 & 0 & K & n & n \\ K & K & K & K & K & K & K & K \\ n+1 & n+1 & n+1 & n & n & K & 0 & 2 \\ n+1 & n+1 & n+1 & n & n & K & 2 & 0 \end{pmatrix}$$

Рис. 2. Матриця відстаней між одиницями обладнання

В даній праці потрібно дослідити знаходження оптимального плану розміщення виробничих ліній, що забезпечить мінімізацію витрат на перевезення. Так як в якості критерію оптимальності в задачі розміщення технологічного обладнання прийнятий мінімум сумарної довжини шляхів, пройдених виробами в процесі обробки, вона зводиться до наступної: знайти мінімум цільової функції

$$\sum_0^{2n} \sum_0^{2n} l(X_i, X_j) a_{ij} \quad (1)$$

Предметом даного дослідження є метод віток і меж знаходження такого плану розміщення обладнання, що забезпечить мінімальні затрати на закупівлю як самої техніки для перевезень так і зменшить транспортні витрати при виробництві, що зрештою вплине на собівартість виробництва.

**Аналіз останніх досліджень.** Метод віток і меж використовується для вирішення деяких NP-повних задач. Швидкість алгоритму залежить від вигляду функції та способу визначення оцінок, але гарантовано не більше за повний перебір. Дана задача є однією з під задач планування гнучких виробничих ліній. Ці задачі широко вивчаються, але формалізованих алгоритмів знаходження оптимальних планів не існує. В більшості випадків ці задачі розв'язуються вручну. В даній статті ми спробуємо дати опис алгоритму розв'язання класу однотипних задач де є один склад і один навантажувач, що може рухатись по заданим маршрутам. Розроблене програмне забезпечення дозволить проектувати витрати на перевезення в бізнес-планах виробництв.

**Основні допущення та рівняння.** Результатом роботи алгоритму є знаходження мінімуму функції (1) на допустимій множині. В ході роботи алгоритму виконується дві операції: розбиття вихідної множини на підмножини(гілки), та знаходження оцінок(меж). Існує оцінка множини згори та оцінка знизу. Оцінка згори - точка що гарантовано не менша за максимум на заданій підмножині. Оцінка знизу - точка що гарантовано не більша за максимум на заданій підмножині.

Будемо вважати, що інтенсивності перевезень між складом та виробничим обладнанням  $X_i$  спадає із зростанням номера. Цього завжди можна досягти простим сортуванням. Програмно виконується досить просто будь-яким класичним способом.

На першому кроці гілкування отримаємо  $n$  різних варіантів.

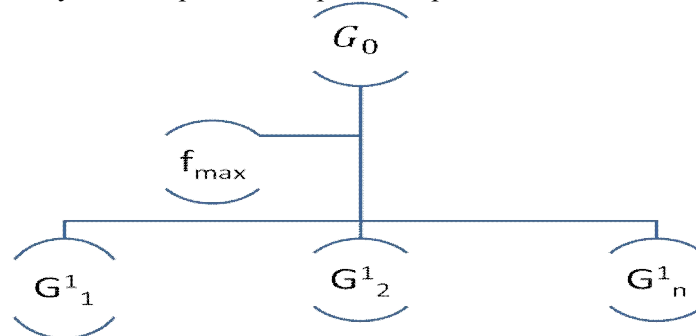


Рис. 3. Варіанти гілкування



Тобто перший тип технологічного обладнання можна встановити в одній з  $n$  пар позицій. Так як розміщення обладнання симетричне ми можемо скоротити кількість варіантів гілкування вдвоє. Тепер постає задача знайти максимальні та мінімальні оцінки для кожної з віток (так звані межі). Верхня межа може бути для всіх віток одна – план розміщення обладнання, що має мінімальну кількість перевезень. Окрім максимального значення запам'ятаємо також план розміщення за якого він досягається. На кожному з наступних кроків ця межа буде зменшуватись доки не зрівняється з мінімальною межею найменшої вітки, що і дасть оптимальний план.

Отже ми визначили, що оцінка зверху – це деякий можливий варіант розміщення технологічного обладнання. Тепер відшукаємо оцінки знизу кожної з віток. Для кожної з віток першого етапу гілкування нам відомо лише 2 позиції (це розміщення складу і розміщення першого виду технологічного обладнання). Усі інші позиції можуть бути заповнені обладнанням у довільному порядку. Найменша довжина перевезень матриці  $L$  рис. 3 складає 2, тому можна вважати, що сумарна довжина перевезень не більша ніж добуток усіх інтенсивностей перевезень на найменшу можливу відстань, тобто 2. Проте така оцінка є занадто грубою і наша задача не буде відрізнятися від простого перебору.

Спробуємо оцінити мінімальну межу суми перевезень якомога максимально. Розглянемо першу вітку. Тут перший тип обладнання знаходиться в першій парі позицій, нехай просто на позиції 1. Відстань, що дорівнює 2 буде лише до складу і позиції 2. Решта позицій знаходяться на відстані 3 і більше. Знайдемо суму матриці  $A$  і транспонованої  $A$ .  $C = A + A^T$ . Тут  $c_{i,j} = a_{i,j} + a_{j,i}$  елементи матриці  $C$  показують суму інтенсивностей перевезень між обладнанням  $i$  та  $j$  та в зворотному напрямку. Додамо рядки 0 і 1 матриці  $C$  з вагами 2, номер стовпця з найбільшою сумою відповідає номеру технологічного обладнання, яке дасть найменший доданок в загальну суму. Тобто для кожного з видів технологічного обладнання можна застосувати формулу, чим більша сума інтенсивностей перевезень між обладнаннями  $X_i$  та  $X_j$  тим на меншу відстань між собою їх потрібно розташовувати. Так як матриці  $C$  і  $L$  симетричні зробимо наступні дії. Матрицю  $C$  замінимо матрицею  $C_0$  в якій елементи під головною діагоналлю замінили нулями, а діагональні на 1. Відсортуємо елементи в рядках новоутвореної матриці над головною діагоналлю в порядку спадання і запишемо в  $C_{\text{сорт}}$ . В матриці  $L$  діагональні елементи замінимо на 1. Добуток цих матриць дасть мінімальну оцінку плану розміщення обладнання. Для кожної з віток різниця буде лиш у тому, що один тип обладнання фіксується у різних позиціях. Для другого етапу гілкування фіксовано уже буде два типи обладнання і так далі. Очевидно, що для кожного етапу мінімальна оцінка не зменшуватиметься. Для кожної вітки наступних етапів гілкувань буде визначатись і максимальний набір, який буде не збільшуватись.

Виконуючи етапи гілкування ми перевіряємо чи наша мінімальна оцінка гілки менша за максимальну загальну оцінку. Якщо менша, то процес гілкування продовжуємо, якщо більша – робимо висновок про не існування оптимального плану на цій гілці, якщо рівні – робимо висновок про знаходження оптимального плану для цієї гілки. При неможливості продовжувати гілкування наш максимальний план буде оптимальною схемою розміщення технологічного обладнання.

Результатом роботи буде не більше ніж  $2n$  етапів гілкування в кожному з яких не більше ніж  $n$  гілок. Обчислювальна складність алгоритму  $O(n^3)$ , тоді як при використанні повного перебору вона буде  $O(c^n)$ .

Розглянемо приклад задачі в якій необхідно 6 різнотипних верстатів розмістити на 6 позиціях так, щоб сумарні витрати на перевезення були мінімальні.

	0	1	2	3	4	5	6
0	0	2	2	3	3	4	4
1	2	0	2	3	3	4	4
2	2	2	0	3	3	4	4
3	3	3	3	0	2	3	3
4	3	3	3	2	0	3	3
5	4	4	4	3	3	0	2
6	4	4	4	3	3	2	0

	0	1	2	3	4	5	6
0	0	2	0	5	0	4	8
1	0	0	2	1	3	8	2
2	0	0	0	0	5	3	0
3	0	0	0	0	2	5	6
4	0	0	0	0	0	2	3
5	0	0	0	0	0	0	3
6	0	0	0	0	0	0	0

Рис. 4. Матриця відстаней між одиницями обладнання (ліворуч) та інтенсивностей передачі виробів (праворуч)

Розроблене програмне забезпечення дало відповідь



Рис. 5. Оптимальний план та затрати на перевезення

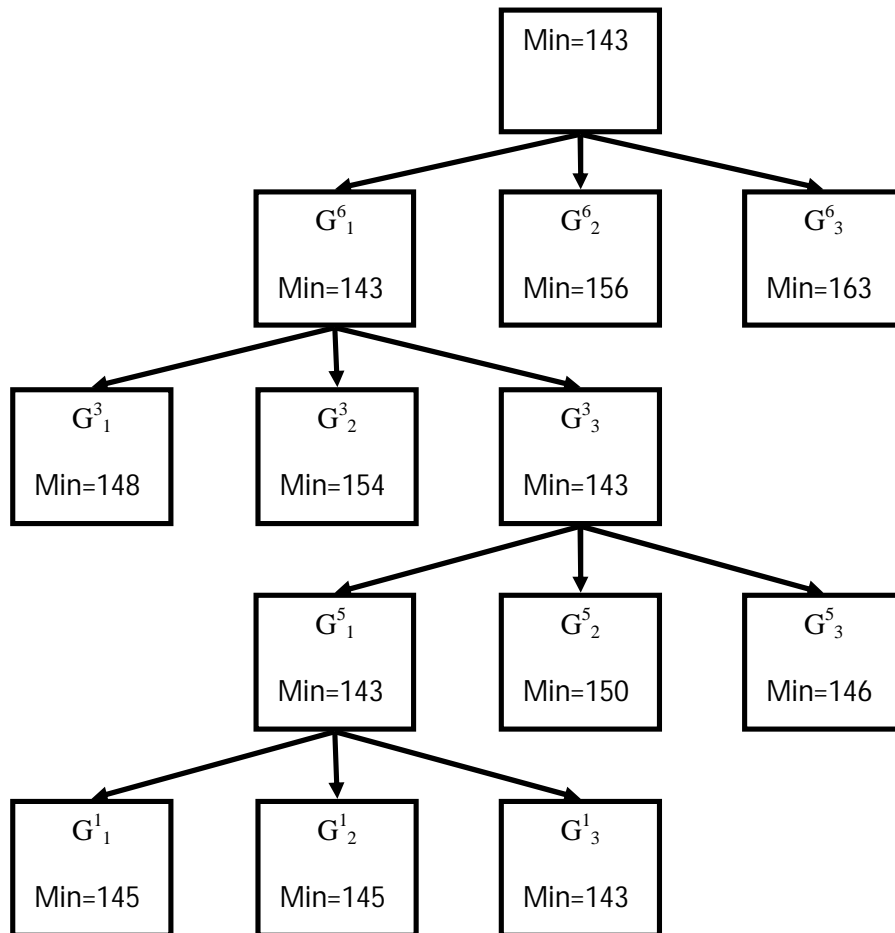


Рис. 6. Граф гілкування

Так як вершини що залишились потрапили на другу пару симетричних позицій то процес гілкування можна завершити. Розроблене програмне забезпечення знайшло оптимальний план перевезень.

**Висновки та перспективи подальших досліджень.** На основі підходів теорії розміщення гнучких виробничих ліній було побудовано алгоритм відшукування оптимальних розміщень досить широкого класу задач та побудовано програмну модель розрахунків. Дана модель враховує знижену кількість гілкувань системи внаслідок симетричності позицій, максимальні і мінімальні оцінки функціонування системи, які встановлюються конкретними технологічними показниками.

Проведено оцінювання оптимального плану розміщень для конкретного технологічного процесу.

Відповідно до запропонованої математичної моделі розроблено та реалізовано алгоритм числового розрахунку мінімальної оцінки транспортних витрат:

- знайти суму матриці інтенсивностей перевезень та транспонованої до неї матриці;
- перетворити отриману матрицю шляхом сортування та заміни елементів;
- перетворити матрицю довжин перевезень та знайти добуток перетворених матриць.

Необхідно відзначити, що в даній праці деталізовано описано знаходження мінімальних

оцінок гілок поділу на кожному етапі гілкування, що дало можливість формалізувати опис. Побудований алгоритм дозволив вирішувати широкий клас задач теорії розміщення. Практична цінність алгоритму у зменшенні обчислювальних потужностей для цього класу задач, що дасть можливість вирішувати ширші задачі спираючись на розв'язану.

Дослідження теорії розміщення виробничого обладнання показує, що при вирішенні широких класів задач застосування методів прямого перебору є дуже неефективне із збільшенням числа вершин. Методи віток і меж дозволяють значно скоротити кількість обчислень, але для більш широких класів задач постає проблема визначення оцінок для віток. Але розроблений алгоритм дозволяє поєднувати його як складову до більш широких алгоритмів імітаційного моделювання або генетичних алгоритмів.

1. Бурдаков С.Ф., Дьяченко В.А., Тимофеев А.Н. Проектирование манипуляторов промышленных роботов и роботизированных комплексов. – М.: Высш. шк., 1986. – 264 с.
2. Довбня Н.М., Кондратьев А.Н., Юревич Е.И. Роботизированные технологические комплексы вГПС. – Л.: Машиностроение. Ленингр. отд-ние, 1990. – 303 с.
3. Кирилович В.А., Підтиченко О.В. Склад математичної моделі гнучких виробничих комірок для задачі автоматизованого планування обладнання // Оптимизация производственных процессов. – № 9. – 2006. – С. 46–53.
4. Козловский В.А. Организационные и экономические вопросы построения производственных систем. – Л.: Изд-во Ленингр. ун-та, 1981. – 216 с.
5. Романов В.А. Математическое и программное обеспечение задач выбора и размещения оборудования на участке машиностроительного производства: Дис... канд. физ.-мат. наук: 05.13.11. – Институт прикладной математики им. М.В. Келдыша. – М., 1986. – 133 с.
6. Keramas James G. Robot technology fundamentals. – New York, Delmar Publishers, 1999. – 408 p.
7. Кирилович В.А. Автоматизоване формування маршрутів обслуговування робочих позицій промисловими роботами / В.А. Кирилович, О.В. Підтиченко // Вісник ТДТУ. – 2008. – Том. 13. – №4. – С. 152 – 157.
8. Шишмарёв В.Ю. Автоматизация производственных процессов в машиностроении. – М.:Издательский центр «Академия», 2007.
9. Полетаев В.А. Разработка компоновки и планировки гибких производственных систем: Методические указания. – Иваново: ИГЭУ, 1999.
10. Калинин О.М., Ямпольский С.Л., Песков Л.В. Моделирование гибких производственных систем. – К.: Техника, 1991
11. Лаздынь С.В., Секирин А.И., Коробкова Т.А. Оптимизация компоновки технологического оборудования гибких производственных систем с использованием генетических алгоритмов. //Международный сборник научных трудов “Прогрессивные технологии и системы машиностроения”, вып. 34. - Донецк: ДонНТУ, 2007. - С. 114-120
12. Курейчик В.В. Эволюционные методы решения оптимизационных задач. - Таганрог: Издво ТРТУ, 1999.

УДК 681.5.015.8

Смолянкін О.О., Маркіна Л.М., Захарчук О.Б.  
Луцький НТУ

## ДОСЛІДЖЕННЯ ТА РЕГУЛЮВАННЯ ТЕПЛОВИХ ПРОЦЕСІВ В ЗАМКНУТОМУ ОБ'ЄКТІ

**Смолянкін О.О., Маркіна Л.М., Захарчук О.Б.** Дослідження та регулювання теплових процесів в замкнутому об'єкті. В статті розглядається ідентифікація теплового процесу методом Сімою. Досліджено перехідну характеристику даного процесу та побудовано модель за допомогою пакету Matlab. Для отриманої моделі розраховано регулятор.

**Ключові слова:** тепловий процес, метод Сімою, перехідна характеристика, ПІД-регулятор, регулювання в Matlab.

**Смолянкин О.А., Маркина Л.М., Захарчук О.Б.** Исследования и регулирование тепловых процессов в замкнутом объекте. В статье рассматривается идентификация теплового процесса методом Симой. Исследована переходная характеристика данного процесса и построена модель с помощью пакета Matlab. Для полученной модели рассчитаны регулятор.

**Ключевые слова:** тепловой процесс, метод Симой, переходная характеристика, ПИД-регулятор, регулирование в Matlab.

**Smolyankin O., Markina L., Zaharchuk O.B.** The research and regulation of thermal processes in a closed facility. The paper considers identification of thermal process by Simoyu method. The transient response of the process was analyzed and the model was built in Matlab package. The regulator for the resulting model was calculated.

**Keywords:** thermal process, Simoyu method, transient response, PID controller, regulating in Matlab

Теплові процеси зустрічаються в усіх галузях виробництва. Найбільше їх використання припадає на харчову [10], хімічну та нафтопереробну галузі [3], де точність швидкість та якість регулювання теплових процесів відіграють ключову роль в якості готової продукції та її цінності категорії. Оскільки проводити як активні так і пасивні експерименти з технологічним обладнанням в таких галузях є затратним, тому доцільним є створення установок, котрі б імітували той чи інший тепловий процес, його дослідження та підбір необхідних параметрів регуляторів.

Необхідно зазначити, що дослідженням та створенням стендів [6] та установок [8] по вивченню даної проблеми займаються Голінко І. М., Мисак В.Ф., Проць Я. І. Та інші вчені.

Українськими вченими розглядаються різні аспекти даної проблеми та невирішеними залишаються можливості варіації параметрів контролерів для регулювання теплових процесів, використання інших контролерів та інших принципів регулювання. Також постає питання про ідентифікацією та дослідження теплових процесів, не охоплених наявними дослідженнями, а отже і їх регулювання.

Метою дослідження є ідентифікація теплового процесу в замкнутому контурі, створення його моделі та визначення оптимальних параметрів регулятора.

Розроблена установка (рис. 1) [1], [5] для зняття перехідних характеристик та підбору регулятора на основі контролю за технологічним процесом нагріву повітря.

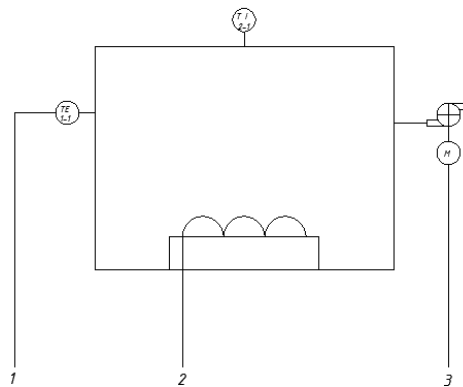


Рис. 1. Функціональна схема процесу нагріву повітря  
авторська розробка

Спроековано прозорий корпус з акрилу, всередині якого знаходиться датчик температури для зняття показів 1-1. Для тарування датчика було застосовано додатковий еталонний термометр. Сигнал від датчика надходить на плату перетворення сигналів, а звідти на контролер. Нагрівання відбувається за допомогою тону з ніхромю 1-3. За необхідності створення збурення вмикається вентилятор 1-4.

Для ідентифікації об'єкту та визначення передаточної функції використано метод Сімою М. П. для об'єкту без самовирівнювання [9]. Всі розрахунки та розробка моделей проводила в програмі Matlab.

Першим етапом ідентифікації стала побудова кривої розгону [2]. Оскільки крива розгону є характеристикою інтегрального типу, то може бути описана моделлю наступного виду:

$$W_M(s) = \frac{K}{s} \cdot \overline{W}_M(s) \cdot e^{-s\tau} = \frac{K}{s} \cdot \frac{1+b_1s+b_2s^2+\dots+b_ms^m}{1+a_1s+a_2s^2+\dots+b_ns^n} \cdot e^{-s\tau} \quad (1)$$

де:

$K$  – коефіцієнт підсилення,

$\tau$  – час запізнення (запізнення),

$a_i, b_i$  – коефіцієнти передаточної функції.

Основною задачею є визначення коефіцієнтів  $a_i, b_i$  передаточної функції.

Після проведення ряду перетворень формулу (1) можна звести до вигляду:

$$W_M(s) = \left( \frac{k_1}{s} - K_2 \cdot \frac{1+b_1s+b_2s^2+\dots+b_ms^m}{1+a_1s+a_2s^2+\dots+b_ns^n} \right) e^{-\tau s} \quad (2)$$

Наступним етапом став розрахунок моментів і площ, які дали змогу побудувати основні найпоширеніші моделі об'єкту регулювання. Результат роботи яких показано на Рис. 2.

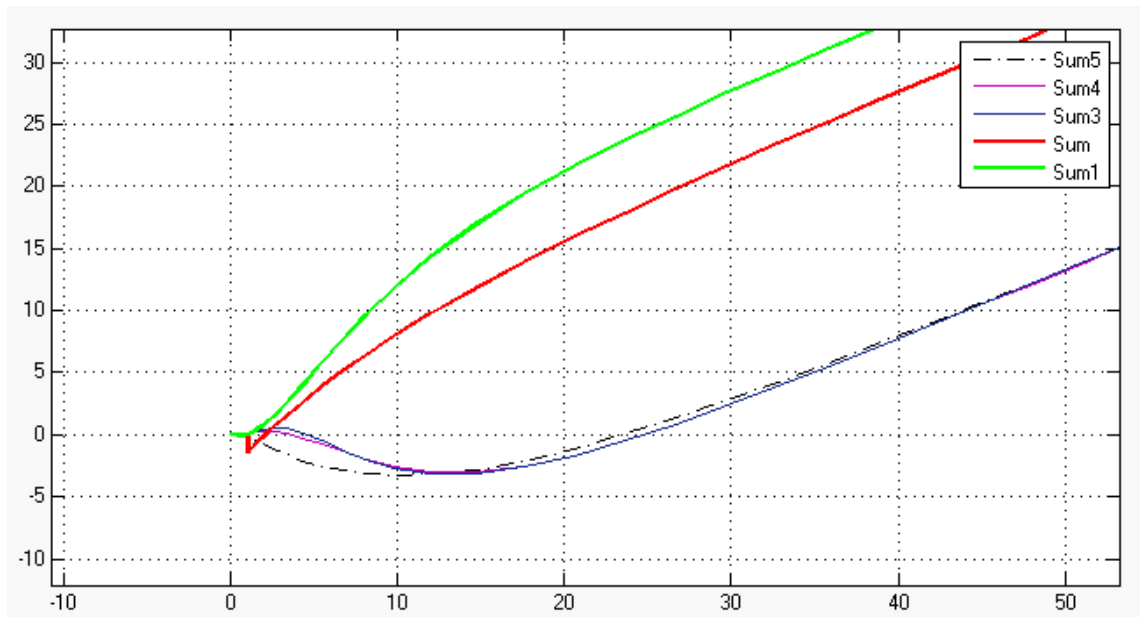


Рис. 2. Результат роботи моделей авторська розробка

Отже, з використаних моделей лише одна точно відображає перехідний процес. Ця модель представлена на рис. 3.

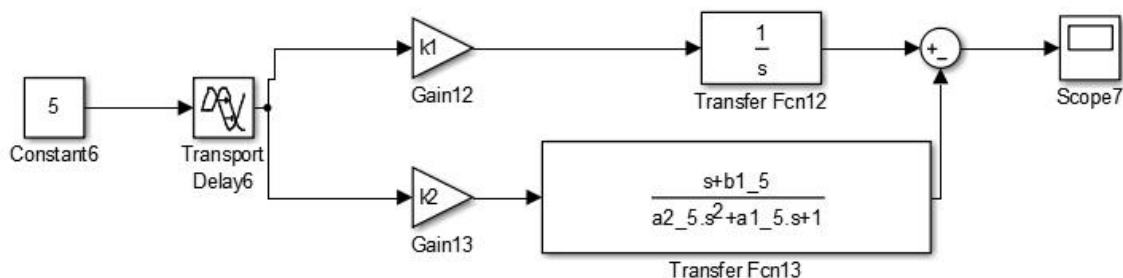


Рис. 3. Модель процесу нагріву повітря авторська розробка

Після отримання нової математичної моделі об'єкта регулювання постало питання підбору коефіцієнтів для ПІД-регулятора [4].

Пропорційно-інтегрально-диференціальний (ПІД) регулятор[7] - найбільш ефективний і поширений вид регулятора, що забезпечує досить високу точність при управлінні різними процесами.

ПІД-регулятор виробляє вихідний сигнал, який розраховується за такою формулою:

$$u = K_1 \left( \varepsilon + \frac{1}{T_i} \int \varepsilon dt + T_p \frac{d\varepsilon}{dt} \right) \quad (3)$$

Пропорційна складова залежить від неузгодженості  $\varepsilon$  і відповідає за реакцію на миттєву помилку регулювання.

Інтегральна складова містить в собі накопичену помилку регулювання, яка є додатковим джерелом вихідної потужності і дозволяє домогтися максимальної швидкості досягнення уставки при відсутності перерегулювання.

Диференціальна складова залежить від швидкості зміни регульованого параметра, що викликає реакцію регулятора на різку зміну вимірюваного параметра, що виникло, наприклад, в результаті зовнішнього збурюючого впливу.

Для ефективної роботи ПІД-регулятора необхідно підібрати значення коефіцієнтів. Це можна зробити вручну або скористатися автонастроюванням.

Було створено Simulink-модель процесу рис. 4.

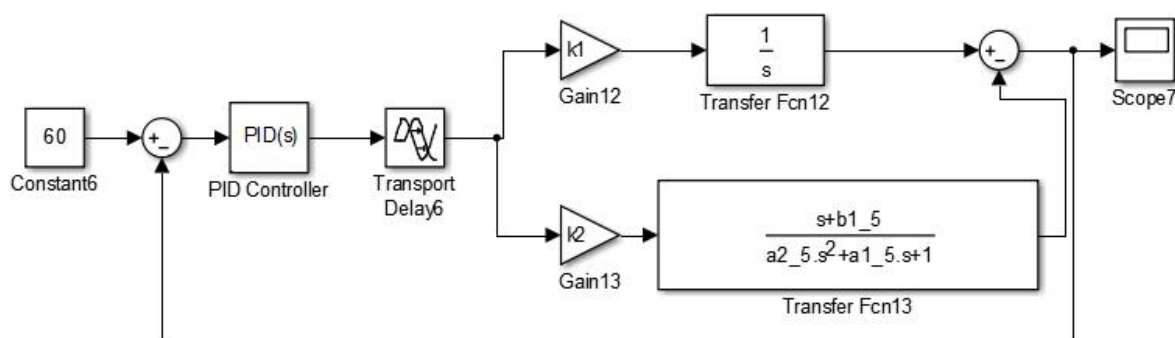


Рис. 4. Simulink-модель об'єкта регулювання з ПІД-регулятором авторська розробка

Для розрахунку коефіцієнтів було використано вбудовану функцію Tune.

Під час розрахунку були визначені такі оптимальні параметри регулятора рис. 5:

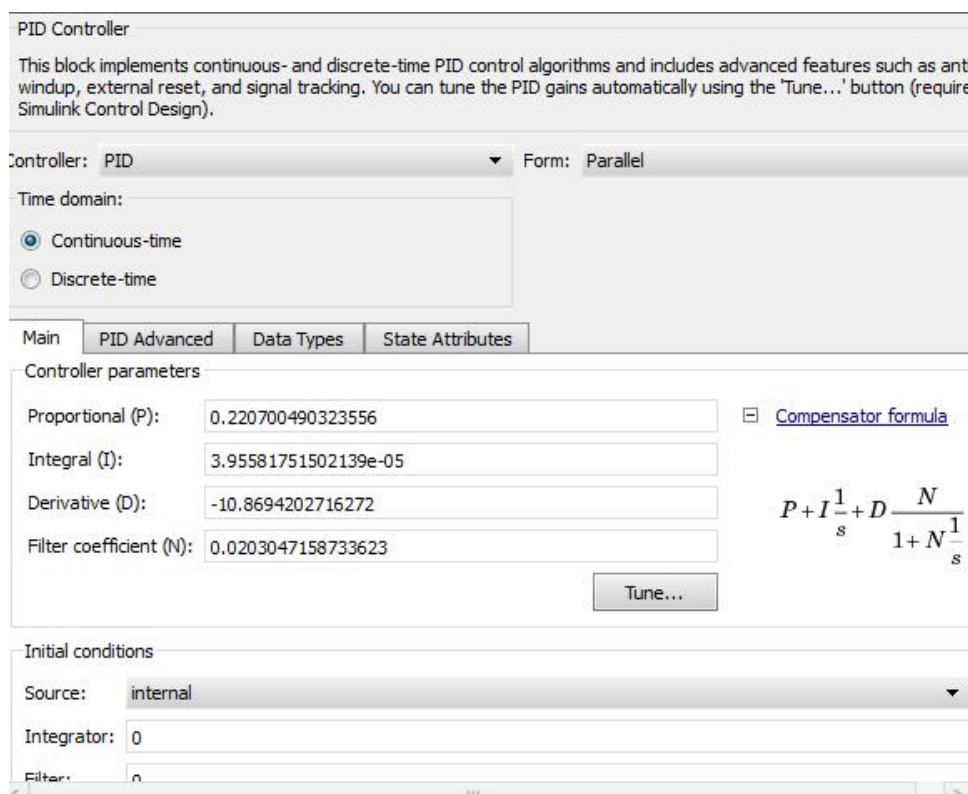


Рис. 5. Коефіцієнти ПД-регулятора авторська розробка

Результат роботи моделі з ПД-регулятором представлено на рис. 6.

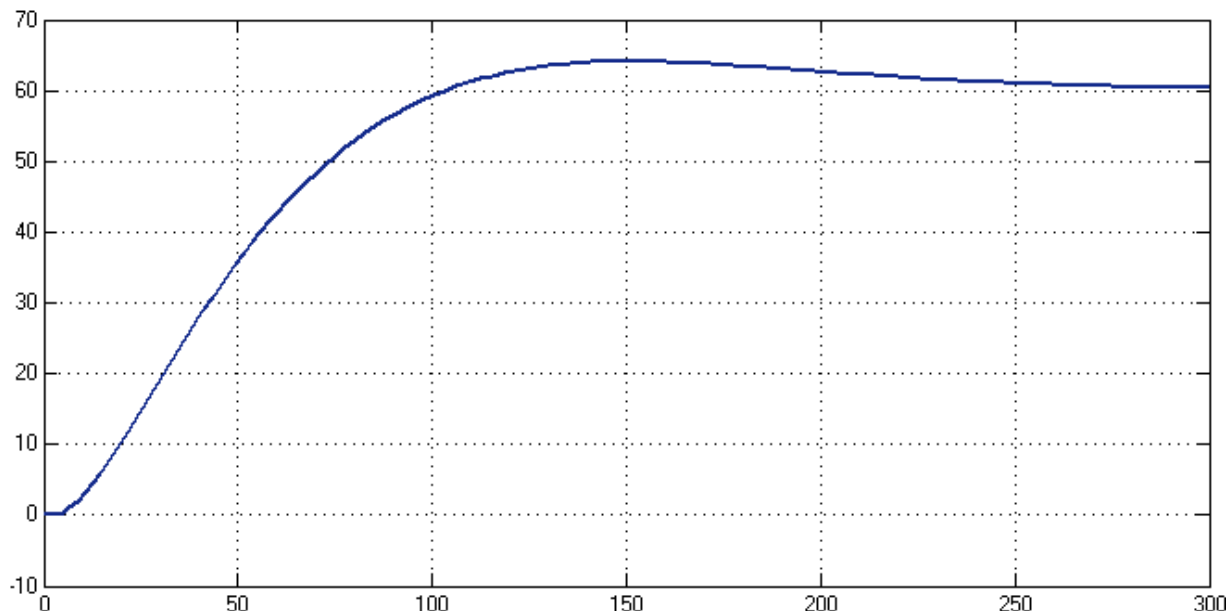


Рис. 6. Робота моделі з ПД-регулятором авторська розробка

Після розробки моді та підбору параметрів регуляторів постало питання в їх реалізації до установки. В якості регулятора було вибрано плату Arduino Uno, яка має не тільки свою платформу для програмування, але й прекрасно працює на базі Matlab і Simulink. Також для даної плати розроблено бібліотеку для ПД-регулювання, яка дає можливість застосувати знайдені параметри ПД-регулятра. Оскільки скетч-програма для плат Arduino досить проста і програми в ній займають мало пам'яті тому доцільно скористатись саме нею.

Спочатку підключається необхідна бібліотека для роботи з ПД регуляторами:

```
#include <PID_v1.h>
```

Далі описуються константи:

```
unsigned long time;
```

```
const int ledPin = 13; // лапка підключення діода
```

```
const int tenPin = 3; // лапка підключення тена
```

```
const int datchuk=0; // лапка підключення датчика температури
```

```
const int ventPin = 11; // лапка підключення вентилятора
```

```
const int knopPin = 2; // лапка підключення кнопки
```

Опис змінних і функцій:

```
double Setpoint, Input, Output;
```

```
PID myPID(&Input, &Output, &Setpoint,0.22,3.96,-10.87, DIRECT); // з перерегулюванням
```

Процедура запуску включає в себе ініціалізацію константи tenPin як «вихід», ledPin як «вихід» та константи datchuk як «вхід», ventPin як «вихід», knopPin як «вхід». Присвоєння змінній Setpoint значення 60 (значення необхідної температури), та прописується, що ПІД-регулятор буде ввімкнений під час створення.

```
void setup (void) {
```

```
pinMode(ledPin, OUTPUT);
```

```
pinMode(ventPin, OUTPUT);
```

```
pinMode(tenPin, OUTPUT);
```

```
pinMode(datchuk, INPUT);
```

```
pinMode(knopPin, INPUT);
```

```
Setpoint = 60;
```

```
myPID.SetMode(AUTOMATIC);
```

```
Serial.begin (9600);
```

```
}
```

В основному циклі програми спочатку прописуються змінні, потім за допомогою полінома 3 степеня прописується значення температури.

```
void loop (void) {
```

```
byte i;
```

```
int k;
```

```
float celsius;
```

```
float x,y;
```

```
y=(5.0/1023)*analogRead(datchuk);
```

```
x=analogRead(datchuk);
```

```
celsius=0.0000004786501*x*x*x-0.0007839772560*x*x+0.5698116814640*x-  
116.7036710064136;
```

```
k=digitalRead(knopPin);
```

Ініціалізується значення температури celsius як вхідне для регулятора, а значення tenPin як вихідне і розраховується вихідний сигнал.

```
Input=celsius;
```

```
Output=tenPin;
```

```
myPID.Compute();
```

Тепер значення вихідного сигналу подається на тен:

```
analogWrite(tenPin,Output);
```

А на термінал виводяться значення часу, напруги на датчику, температури.

```
Serial.print (celsius);
```

```
Serial.print ("\t");
```

```
time = millis()/1000;
```

```
Serial.println(time);
```

```
delay(2000);
```

У разі виникнення ситуації, коли температура перевищує 100 градусів автоматично вимикається тен і засвідчується 13діод на платі.

```
if (celsius < 100){
```

```
digitalWrite(ledPin, HIGH);
```

```
analogWrite(tenPin, 0); }
```

Якщо кнопка ввімкнена, то вмикається вентилятор.



```
if (k!=0)  
digitalWrite(ventPin, 30);  
else digitalWrite(ventPin, 0);}
```

Результат роботи програми можна побачити в терміналі в режимі реального часу з вказаним інтервалом.

Після проведення дослідів отримано наступні графіки регулювання рис. 7:

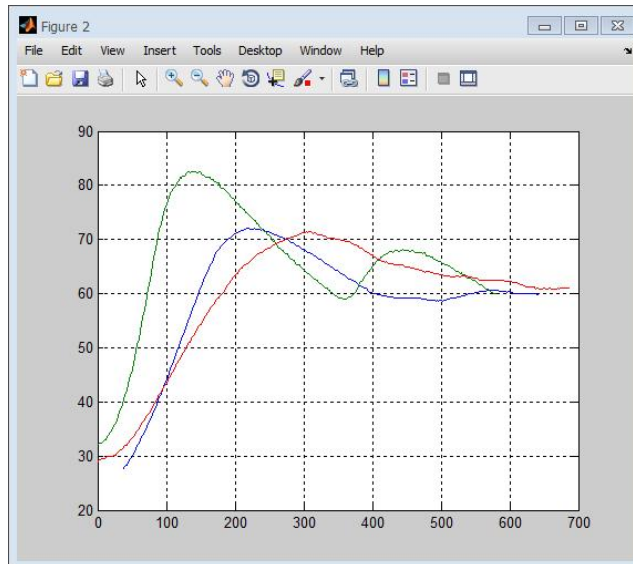


Рис. 7. Перехідні характеристики з Т-регулятором, ПД-регулятором і зі збуренням авторська розробка

Аналізуючи вище викладене можна зробити висновок, що ПД-регулювання можна застосовувати в теплових процесах, де необхідна точність, плавність і якість регулювання. В той же час побудова і настройка таких регуляторів не вимагає особливих затрат.

Проте на даний час існують більш точні та якісно нові регулятори, побудовані на основі нечіткої логіки та нейромодельовання. Можливість їх застосування до теплових процесів на даному етапі розвитку науки мало вивчена, тому перспектива подальших досліджень лежить саме в цьому напрямку.

1. НПАОП 40.1-1.32-01 Правила будови електроустановок. Електрообладнання спеціальних установок.
2. Волгин В.В. Методы расчета систем автоматического регулирования/ В.В. Волгин// Учебное пособие. – М.: Издательство МЭИ, 1972. – 192 с.
3. Врагов А.П. Теплообмінні процеси та обладнання хімічних і газонафтопереробних виробництв/ А.П. Врагов // Навч. посібник. – Суми: Вид-во СумДУ, 2006. – 262с.
4. Говоров П. П. Конспект лекцій з курсу «Теорія автоматичного керування» (для студентів денної і заочної форм навчання напряму підготовки 6.050701 «Електротехніка та електротехнології») / П. П. Говоров, В. П. Говоров, В. О. Перепечений, О. В. Король //Харк. нац. акад. міськ. госп-ва; авт. - Х.: ХНАМГ, 2012. - 221 с.
5. Клюев А.С. Проектирование систем автоматизации и ТП/ А.С. Клюев, Б.В. Глазов. - М: Рад. шк., 1990.-464с.
6. Мисак В.Ф. Методичні вказівки до виконання лабораторних робіт з курсу «Автоматизація і енергозбереження» для студентів спеціальності «Автоматизоване управління виробничими процесами»/Укл. В.Ф. Мисак, І.М. Голінко – Київ:НТУУ «КПІ», 2006.-31с.
7. Попович М.Г. Теорія автоматичного керування/ М.Г. Попович, О.В. Ковальчук. –К.:Либідь, 1997.-544с.
8. Проць Я.І. Автоматизація неперервних технологічних процесів/Я.І. Проць, О.А. Данилюк, Т.Б. Лобур // Навчальний посібник для технічних спеціальностей вищих навчальних закладів – Тернопіль: ТДТУ ім. І.Пулюя, 2008. –239с.
9. Симою М.П. Определение коэффициентов передаточных функций линеаризованных звеньев систем регулирования/ М.П. Симою.// Автоматика и телемеханика. – 1957. - № 6. – С 514-527.
10. Товажнянський Л.Л.. Харчові технології у прикладах і задачах/ Л.Л. Товажнянський, С.І. Бухкало, П.О. Капустенко, О.П. Арсенєва, С.І. Орлова //Підручник. – К.: Центр учбової літератури, 2008. – 576 с.

УДК 621.3.078.4

Каганюк О.К. к.т.н. доц.

Луцький національний технічний університет

## РІВНЯННЯ НЕКЕРОВАНОГО РУХУ ВУГЛЕВИДОБУВНИХ МАШИН

**Каганюк О.К. Рівняння некерованого руху вуглевидобувних машин.** Одним з основних шляхів підвищення продуктивності праці і інтенсифікації виробництва є автоматизація технологічних процесів, впровадження промислових роботів і маніпуляторів. Інтенсифікація видобутку вугілля без постійної присутності людей в забої висувають нові вимоги до рішення сучасних систем автоматичного управління забійним устаткуванням.

Однією з важливих підсистем управління є автоматичне управління вуглевидобувним комбайном у вугільному пласту.

**Ключові слова:** Система автоматичного керування, профіль пласта, безлюдна виїмка вугілля

**Kaganiuk A.K. The equation of motion uncontrolled mining machinery.** One of the main ways to increase productivity and intensification of production is automation of technological processes, implementation of industrial robots and manipulators. Intensification of coal without the constant presence of people in the mine put forward new requirements for modern solution Automatic Control Systems slaughter equipment.

One of the major subsystems of management control is automatically Mining equipment in coal seams.

**Keywords:** Automatic control layer profile, lonely dredging coal

**Каганюк А.К. Уравнения нерегулируемого движения угледобывающими машинами.** Одним из основных путей повышения производительности труда и интенсификации производства есть автоматизация технологических процессов, внедрение промышленных роботов и манипуляторов. Интенсификация добычи полезного ископаемого без постоянного присутствия людей в забое, выдвигает новые требования по совершенствованию современных систем автоматического управления угледобывающими комбайнами. Одной из наиболее важной подсистем есть подсистема управления угледобывающими комбайнами в профиле пласта

**Ключові слова:** Система автоматичного керування, профіль пласта, безлюдна виїмка вугілля

Одним з основних шляхів підвищення продуктивності праці і інтенсифікації виробництва є автоматизація технологічних процесів, впровадження промислових роботів і маніпуляторів.

**Актуальність** цієї статті полягає в тому, що з пошуком альтернативних джерел енергії, стає необхідним займатися розробкою вугільних сховищ, побудовою систем автоматичного управління вугільним комбайнами з використанням нових технологій і можливостей швидкого впровадження систем автоматичного управління вугільними комбайнами.

**Проблеми** інтенсивного і високопродуктивного видобутку вугілля без постійної присутності людей в забої висувають нові вимоги до рішення сучасних систем автоматичного управління забійним устаткуванням.

Однією з важливих підсистем управління є автоматичне управління вуглевидобувним комбайном у вугільному пласту. Досі це завдання не вирішене, що ставить неможливим виведення машиніста із забою. Необхідність постійно стежити за процесом різання і положенням різальних елементів обумовлює перебування машиніста комбайна в зоні підвищеної небезпеки.

На рішення цієї задачі були спрямовані зусилля великої кількості вчених і наукових колективів, які в радянські часи працювали над створенням систем автоматичного управління вугільними комбайнами не лише в Україні, але і в інших країнах світу.

Розглянемо управління об'ємної моделі об'єкту при переміщенні цієї моделі в повздовжньому напрямі. Для складання рівняння руху моделі використовуємо траєкторію руху утворюючої точки С робочого елемента цієї моделі некерованої машини в повздовжньому напрямі.

З урахуванням прийнятих допущень, об'ємна модель об'єкту представлена на рис. 1.

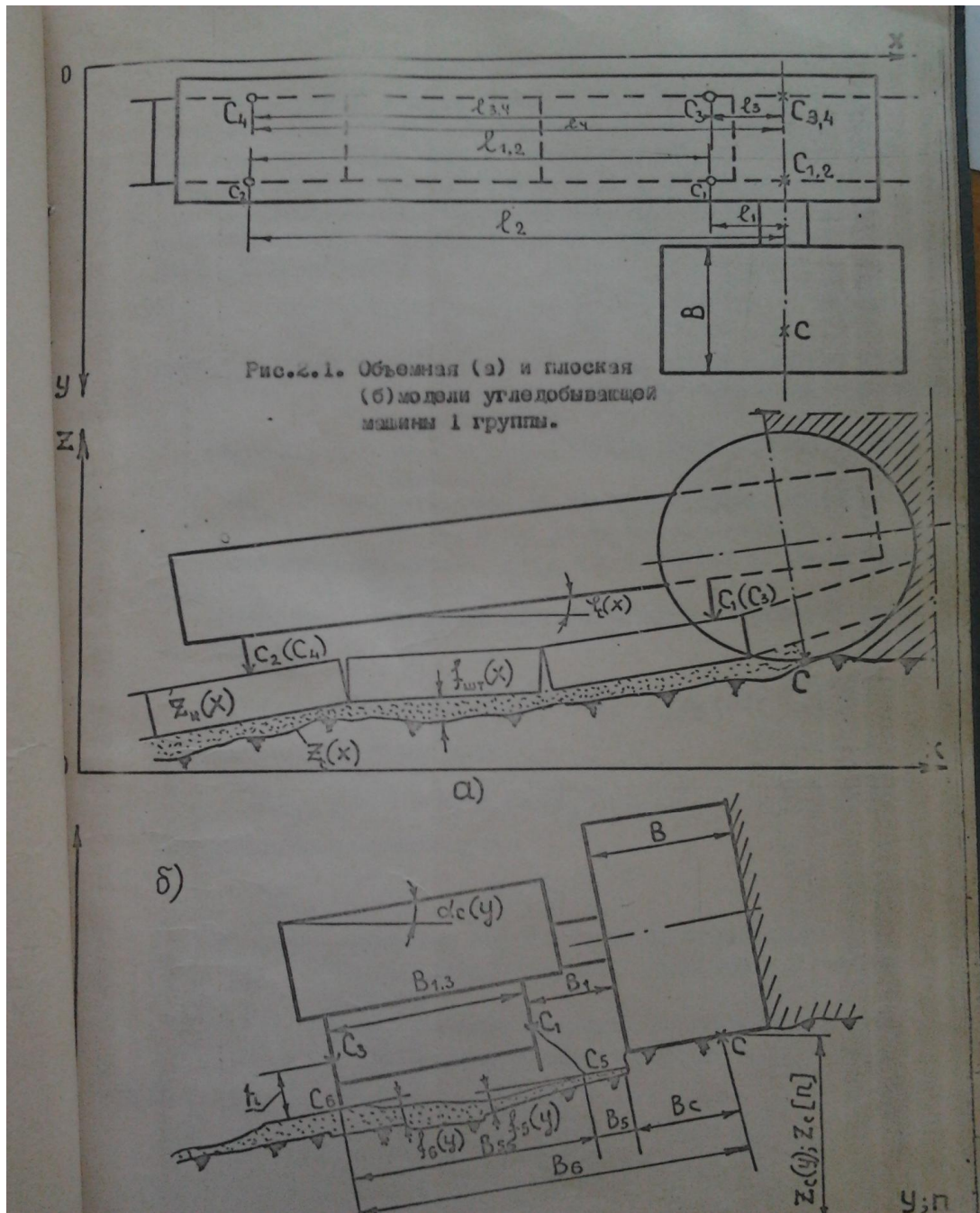


Рис.2.1. Объемная (а) и плоская (б) модели угледобывающей машины I группы.

Рис. 1

Розглянемо варіант побудови моделі, коли забійні опори  $C_1, C_2$ ; завальні  $C_3, C_4$  обладнані зворотними тримачами і кінематично не пов'язаних між собою.

Оскільки усі опори обладнані зворотними тримачами, то для безперешкодного руху машини по конвеєру повинна виконуватися умова:

$$\frac{Z_{C_1}^K(x) - Z_{C_2}^K(x)}{l_{1,2}} = \frac{Z_{C_3}^K(x) - Z_{C_4}^K(x)}{l_{3,4}} \quad (1)$$

де  $Z_{C_{1,2,3,4}}^K(x)$  — координати відповідні забійного і завального бортів конвеєра;

$l_{1,2}, l_{3,4}$  — відстані між опорами  $C_1$  і  $C_2, C_3$  і  $C_4$ .

Проте в переважній більшості випадків рівняння (1) не виконується. У наслідку цього і наявності обмежених проміжків в опорах відбувається заклинювання такого об'єкту, що призводить до появи додаткових навантажень на конвеєр. При цьому об'єкт стає неприцездатним.

Розглянемо об'ємну модель, коли опори  $C_3, C_4$  обладнані зворотними тримачами, а забійні опори  $C_1, C_2$  не мають їх і опори кінематично не пов'язані між собою.

Умова безперервного ковзання опор  $C_1$  і  $C_2$  по борту конвеєра виражається рівнянням (1), яке може бути виконане тільки в окремих випадках. Звідси витікає невизначеність в положенні опор  $C_1$  і  $C_2$  на борту конвеєра.

Невизначеність в положенні опор  $C_1$  і  $C_2$  машини породжує невизначеність положення робочого елемента. З урахуванням невеликих кутів можна записати:

$$Z_C(x) = Z_{C_{3,4}}(x) + B_6 \cdot \alpha_C(x) \quad (2)$$

де  $Z_{C_{3,4}}(x)$  — узагальнена координата центру повороту робочого елемента в поперечній площині, що проходить через точку  $C$ ;

$\alpha_C(x)$  — кут нахилу робочого елемента в поперечній площині;

$B_6$  — відстань від завального борту конвеєра (точки  $C_{3,4}$  і  $C_6$ ) до утворюючої точки  $C$ .

Узагальнена координата  $Z_{C_{3,4}}(x)$  залежить від розташування опор  $C_3$  і  $C_4$  на машині.

Тоді при розташуванні утворюючої точки  $C$  робочого елемента поза опорами  $C_3$  і  $C_4$  маємо наступні співвідношення:

$$Z_{C_{3,4}}(x) = K_3 \cdot Z_K(x - l_3) - K_4 \cdot Z_K(x - l_4) \quad (3)$$

при розташуванні між опор

$$Z_{C_{3,4}}(x) = K_3 \cdot Z_K(x + l_3) - K_4 \cdot Z_K(x - l_4) \quad (4)$$

при розташуванні проти опор

$$Z_{C_{3,4}}(x) = Z_K(x) \quad (5)$$

де  $Z_K(x)$  — координати завального борту конвеєра;

$x$  — поточна координата утворюючої точки робочого елемента;

$l_3$  — геометричний параметр, що враховує розташування опори  $C_3$  відносно точки  $C$ ;

$Z_K(x - l_3); Z_K(x + l_3)$  — координати  $Z_{C_3}(x)$  опори  $C_3$ ;

$Z_K(x - l_4)$  — координати  $Z_{C_4}(x)$  опори  $C_4$ ;

$K_3, K_4$  — коефіцієнти, що враховують взаємне розташування опор  $C_3, C_4$  і утворюючої точки  $C$  причому :

$$K_3 = \frac{l_4}{l_{3,4}}, K_4 = \frac{l_3}{l_{3,4}} \text{ — для рівняння (3)}$$

$$K_3 = \frac{l_4}{l_{3,4}}, K_4 = -\frac{l_3}{l_{3,4}} \text{ — для рівняння (4) і}$$

$$K_3 = 1, K_4 = 0 \text{ — для рівняння (5).}$$

Провівши заміну змінних ( $t = \frac{x}{v_n}$ ) і застосувавши перетворення Лапласа за нульових

початкових умов, для загального випадку отримуємо:

$$Z_{C_{3,4}}(p) = K_3 \cdot Z_K(p) \cdot e^{-\tau_3 p} - K_4 \cdot Z_K(p) \cdot e^{-\tau_4 p} \quad (6)$$

де  $v_n$  — швидкість руху машини по конвеєру;

$$\tau_3 = \frac{l_3}{v_n}, \tau_4 = \frac{l_4}{v_n} \text{ — при розташуванні утворюючої точки С поза опорами;}$$

$$\tau_3 = -\frac{l_3}{v_n}, \tau_4 = \frac{l_4}{v_n} \text{ — при розташуванні утворюючої точки С між опорами;}$$

$$\tau_3 = 0, \tau_4 = \frac{l_4}{v_n} \text{ — при розташуванні утворюючої точки С проти опори С}_3.$$

Кут нахилу робочого органу в поперечній площині у разі опор, жорстко пов'язаних з корпусом машини, визначається нахилом секцій конвеєра під опорами  $C_1$  і  $C_3$  або  $C_2$  і  $C_4$  по рівняннях :

$$\alpha_c(p) = \alpha_k(p) \cdot e^{-\tau_3 p} \quad (7)$$

при

$$\varphi_{1,2}(p) = \frac{Z_{C_1}^K(p) - Z_{C_2}^K(p)}{l_{3,4}} \geq \frac{Z_{C_3}^K(p) - Z_{C_4}^K(p)}{l_{3,4}} \quad (8)$$

і

$$\alpha_c(p) = \alpha_k(p) \cdot e^{-\tau_4 p} \quad (9)$$

при

$$\varphi_{1,2}(p) \langle \varphi_c(p) \quad (10)$$

де  $\varphi_c(p)$  — оператор, що характеризує кут траєкторії робочого елемента в повздовжньому напрямі;

$\varphi_{1,2}(p)$  — оператор, що характеризує нахил уявної лінії, що сполучає точки конвеєра під опорами  $C_1$  і  $C_2$ ;

$Z_{C_{1,2,3,4}}^K$  — оператори, що характеризують положення бортів конвеєра під опорами  $C_{1,2,3,4}$ ;

$\alpha_k(p)$  — оператор, що характеризує кутове положення конвеєра в поперечному напрямі.

З виразу (8) виходить, що

$$\varphi_c(p) = \frac{e^{-\tau_3 p} - e^{-\tau_4 p}}{l_{3,4}} Z_K(p) \quad (11)$$

Підставимо рівняння (6) і (7), (9) в перетворений по Лапласу вираз (2), отримаємо:

$$Z_C(p) = (K_3 \cdot e^{-\tau_3 p} - K_4 \cdot e^{-\tau_4 p}) \cdot Z_K(p) + B_6 \cdot \alpha_k(p) \cdot e^{-\tau_3 p} \quad (12)$$

при  $\varphi_{1,2}(p) \geq \varphi_c(p)$  і

$$Z_C(p) = (K_3 \cdot e^{-\tau_3 p} - K_4 \cdot e^{-\tau_4 p}) \cdot Z_K(p) + B_6 \cdot \alpha_k(p) \cdot e^{-\tau_4 p} \quad (13)$$

при  $\varphi_{1,2}(p) \langle \varphi_c(p)$ ,

де коефіцієнти  $K_3, K_4$  і запізнення  $\tau_3, \tau_4$  слід брати відповідно до розташування робочого елемента відносно опор машини (див. рівняння, наведені вище).

З рівнянь (7), (9), (13) випливає, що в загальному випадку кожна з вихідних змінних об'єкту  $Z_C(p)$  і  $\alpha_c(p)$  визначається величинами двох вхідних змінних  $Z_K(p)$  і  $\alpha_k(p)$ . Такий об'єкт є взаємозв'язним.

Відповідно до теореми багатозв'язного регулювання [1] виділимо в нашому об'єкті прямі або сепаратні канали передачі дії і перехресні зв'язки, в яких дія одного сепаратного каналу прикладена до іншого.

У загальному випадку в об'єктах, що розглядаються нами, можна виділити три сепаратні канали і не менше двох перехресних прямих або зворотних зв'язків. Прямий перехресний зв'язок

передає дії з одного сепаратного каналу на інший в напрямі, співпадаючому з напрямом передачі сигналів сепаратними каналами. У зворотному перехресному зв'язку ці напрями протилежні.

Домовимося передавальні функції сепаратних каналів означати далі  $W$ , прямих перехресних зв'язків - через  $M$ , а зворотних - через  $L$ .

Залежності  $Z_c(p) = f_k[Z_k(p)]$  і  $\alpha_c(p) = f_k[\alpha_k(p)]$  характеризують сепаратні канали об'єкту, а залежності  $Z_c(p) = f_\alpha[\alpha_k(p)]$ ,  $\alpha_c(p) = f_k[Z_k(p)]$ ,  $\varphi_c(p) = f_k[Z_k(p)]$  і  $\varphi_c(p) = f_\alpha[\alpha_k(p)]$  — природні перехресні зв'язки.

Введемо наступні позначення для передавальних функцій сепаратних каналів об'єкту і його перехресних зв'язків :

$W_{zz}(p) = \frac{Z_c(p)}{Z_k(p)}$  — власна передавальна функція сепаратного каналу вертикальної координати

об'єкту;

$W_{\alpha\alpha}(p) = \frac{\alpha_c(p)}{\alpha_k(p)}$  — власна передавальна функція сепаратного каналу кутової координати

об'єкту в поперечному напрямі;

$W_{\varphi\varphi}(p) = \frac{\varphi_c(p)}{\varphi_k(p)}$  — власна передавальна функція сепаратного каналу кутової координати

об'єкту в повздовжньому напрямі;

$M_{\alpha z}(p) = \frac{\alpha_c(p)}{Z_k(p)}$  — передавальна функція перехресного зв'язку першого каналу з другим

(невласна передавальна функція сепаратного каналу кутової координати об'єкту в поперечному напрямі);

$M_{z\alpha}(p) = \frac{Z_c(p)}{\alpha_k(p)}$  — невласна передавальна функція сепаратного каналу вертикальної

координати об'єкту;

$M_{\varphi z}(p) = \frac{\varphi_c(p)}{Z_k(p)}$  — невласна передавальна функція сепаратного каналу кутової координати

об'єкту в подовжньому напрямі по дії  $Z_k(p)$ ;

$M_{\varphi\alpha}(p) = \frac{\varphi_c(p)}{\alpha_k(p)}$  — те ж, але по дії  $Z_k(p)$ .

Тут  $\varphi_k(p)$  — оператор, що характеризує кутове положення конвеєра в повздовжньому напрямі. Проте дія  $\varphi_k(p)$  являється похідною від  $Z_k(p)$ . У зв'язку з цим  $\varphi_k(p)$  і в окремому випадку в розрахунках не бере участь, а тому  $W_{\varphi\varphi}(p), W_{z\varphi}(p), W_{\varphi\alpha}(p)$  рівні 0.

Багатозв'язний об'єкт зручно описувати як матричний функціональний блок, який відповідає залежності кожного виходу об'єкту від кожного його входу. Причому, число рівнянь матричного блоку дорівнює числу виходів об'єкту.

Запишемо рівняння матричного блоку :

$$Z_c(p) = W_{zz}(p) \cdot Z_k(p) + M_{z\alpha}(p) \cdot \alpha_k(p) \quad (14)$$

$$\alpha_c(p) = W_{\alpha\alpha}(p) \cdot \alpha_k(p) + M_{\alpha z}(p) \cdot Z_k(p) \quad (15)$$

$$\varphi_c(p) = W_{\varphi z}(p) \cdot Z_k(p) + M_{\varphi\alpha}(p) \cdot \alpha_k(p) \quad (16)$$

Тоді передавальна матриця об'єкту буде рівна:

$$W_{\sigma\delta}(p) = \begin{vmatrix} W_{zz}(p) & M_{z\alpha}(p) & 0 \\ M_{\alpha z}(p) & W_{\alpha\alpha}(p) & 0 \\ M_{\varphi z}(p) & M_{\varphi\alpha}(p) & 0 \end{vmatrix} \quad (17)$$

Об'єкти даної групи не мають зворотних перехресних зв'язків, і тому вони не входять в приведену передавальну матрицю. Вхідними і вихідними сигналами матричної ланки є вектори відповідно входу і виходу.

Для цього варіанту передавальні функції матриці (1) рівні:

$$W_{zz}(p) = K_3 \cdot e^{-\tau_3 p} - K_4 \cdot e^{-\tau_4 p} \quad (18)$$

$$W_{\alpha\alpha}(p) = \begin{cases} e^{-\tau_3 p} & \varphi_{1,2}(p) \geq \varphi_c(p) \\ e^{-\tau_4 p} & \varphi_{1,2}(p) < \varphi_c(p) \end{cases} \quad (19)$$

$$M_{\alpha z}(p) = 0 \quad (20)$$

$$M_{z\alpha}(p) = \begin{cases} B_6 \cdot e^{-\tau_3 p} & \varphi_{1,2}(p) \geq \varphi_c(p) \\ B_6 \cdot e^{-\tau_4 p} & \varphi_{1,2}(p) < \varphi_c(p) \end{cases} \quad (21)$$

$$M_{\varphi z}(p) = \frac{e^{-\tau_3 p} - e^{-\tau_4 p}}{l_{3,4}} \quad (22)$$

$$M_{\varphi\alpha}(p) = 0 \quad (23)$$

Висновки:

Розглянуті об'єкти є багатозв'язними і характеризуються чотирма ненульовими передавальними функціями. Оскільки кожна з передавальних функцій (13) - (22) є пропорційною ланкою, а перехресні зв'язки в об'єкті тільки прямі, то увесь об'єкт, як матрична ланка, є стійким. Проте, параметри цього об'єкту змінні, оскільки  $W_{\alpha\alpha}(p)$  і  $W_{z\alpha}(p)$  залежать від значень  $\varphi_{1,2}(p)$ . Наявність змінної структури дещо ускладнює управління таким об'єктом і вимагає введення в систему управління додаткових корегуючи ланок, які будуть розглянуті в наступних статтях.

1. Морозовский В.Т. Многосвязанные системы автоматического регулирования. Энергия, М., 1970г.
2. Картавий Н.Г., Шаев Л.В. Серов В.А. Расположение рабочих органов угледобывающих комбайнов относительно опор. Горные машины и автоматика, №10 «Недра», М. 1966г.
3. [WWW.COAL.IN.UA](http://www.coal.in.ua).
4. MiningWiki Шахтерская энциклопедия.
5. <http://rozrobkinauchni.vscorh.org>.

УДК 539.375

Крадінова Т.А.

Луцький національний технічний університет

## АЛГОРИТМ РОЗРАХУНКУ ЗАЛИШКОВОГО РЕСУРСУ ЗАЛІЗОБЕТОННОЇ БАЛКИ

**Крадінова Т.А. Алгоритм розрахунку залишкового ресурсу залізобетонної балки.** Розглянуто залізобетонну балку на двох опорах з попередньо напруженою арматурою, яка навантажена довготривалими рівнорозподіленими нормальними зусиллями інтенсивності  $p$ . Вважається, що в поперечному перерізі нижнього навантаженого розтягом арматурного стержня діаметра  $D$  утворилася система поверхневих сегментних тріщин початкової глибини  $c_0$ , які при заданому довготривалому навантаженні можуть поширюватися як тріщини низькотемпературної повзучості.

Задача полягає у визначенні такого часу  $t = t_*$ , за який тріщини підростуть від розміру  $C_0$  до критичного розміру  $C = C_*$  і арматура зруйнується.

Розглянута аналогічна задача у випадку згину.

Використовуючи енергетичний підхід, побудовано розрахункову методику для визначення залишкового ресурсу арматури в попередньо напружених залізобетонних конструкціях.

**Ключові слова:** попередньо напружені залізобетонні конструкції, тріщини низькотемпературної повзучості, залишковий ресурс, період докритичного росту тріщин низькотемпературної повзучості, коефіцієнт інтенсивності напружень.

**Крадінова Т.А. Алгоритм расчета остаточного ресурса железобетонной балки.** Рассмотрена железобетонная балка на двух опорах с предварительно напряженной арматурой, которая нагружена долговременными равномерно распределенными нормальными усилиями интенсивности  $p$ . Считается, что в поперечном сечении нижнего нагруженного растяжением арматурного стержня диаметра  $D$  образовалась система поверхностных сегментных трещин начальной глубины  $c_0$ , которые при заданной долговременной нагрузке могут распространяться как трещины низкотемпературной ползучести.

Задача заключается в определении такого времени  $t = t_*$ , за который трещины подрастут от размера  $c_0$  к критическому размеру  $C = C_*$  и арматура разрушится.

Рассмотрена аналогичная задача в случае изгиба.

Используя энергетический подход, построена расчетная методика для определения остаточного ресурса арматуры в предварительно напряженных железобетонных конструкциях.

**Ключевые слова:** предварительно напряженные железобетонные конструкции, трещины низкотемпературной ползучести, остаточный ресурс, период докритического роста трещин низкотемпературной ползучести, коэффициент интенсивности напряжений.

### T.A. Kradinova. Algorithm of calculation of remaining resource of reinforce-concrete beam

Considered reinforced concrete beam on two supports with pre-stressed reinforcement, which loaded long-term runs normal effort intensity  $p$ . It is believed that in the cross section of the loaded lower tensile rebar diameter  $D$  formed by a system of surface radial cracks the initial depth  $c_0$ , which when given a long load can be distributed as crack low-temperature creep.

The objective is to determine a time  $t = t_*$  for which the crack will grow from the size  $c_0$  to the critical size  $C = C_*$  and the armature will be destroyed.

A similar problem is considered in the case of the bend.

Using the energy approach is a calculation method to determine the residual life of the reinforcement in prestressed concrete structures.

**Keywords:** prestressed concrete structures, cracks in the low-temperature creep, residual life, period to critical crack growth of low-temperature creep, the stress intensity factor.

В даний час, коли відбувається інтенсивна експлуатація конструкцій і споруд, важливо передбачити і забезпечити надійну роботу їх елементів. Сюди можна віднести обладнання енергетичного устаткування теплових і атомних електростанцій, нафтогазопроводів, металевих і залізобетонних конструкцій, які працюють за довготривалих статичних навантажень в умовах невисоких температур.

Тут необхідно підкреслити, що на процес поширення тріщин впливають різні фактори, зокрема, такі як напруження, температура, час навантаження, структура матеріалу тощо. Під час довготривалого статичного навантаження руйнування металів проходить (за відсутності агресивних середовищ й інших фізико-хімічних чинників) за механізмами низькотемпературної або високо температурної повзучості. Серед праць, в яких досліджували руйнування такого типу,



можна назвати фундаментальні праці українських учених: Лебедева А.О., Писаренка Г.С., Русинка К.М., Шевченка Ю.М. та інших. Переважно це аналітичні і експериментальні дослідження для бездефектних тіл. Для випадку поширення тріщин низькотемпературної повзучості в полімерах й інших в'язкопружних тілах, така теорія створена в працях А.О. Камінського [6, 7] на базі відомої  $\delta_c$  – моделі і принципу Вольтера в лінійній теорії в'язкопружності. Однак, випадки поширення в металевих пружно-пластичних матеріалах тріщин повзучості не можуть бути описані в рамках такої теорії через відмінності механізмів сповільненого руйнування.

Використаємо експериментальну методику для розрахунку залишкової довговічності арматури у заздалегідь напружених залізобетонних елементах конструкцій. Застосуємо математичну модель, представлену раніше [1,5], для визначення періоду докритичного росту тріщин низькотемпературної повзучості за довготривалого статичного навантаження арматури і модифікацію відомого методу граничної інтерполяції на випадок наближеного визначення коефіцієнтів інтенсивності напружень біля тріщин в арматурі.

Під час навантаження залізобетонних конструкцій спостерігається утворення тріщин в розтягнутій зоні бетону [2,3]. Із збільшенням навантаження розтягу напруження сприймаються арматурою, а тріщини в бетоні розкриваються. Для великої кількості конструкцій, арматура яких має звичайну міцність (не високоміцна), ширина розкриття тріщин за дії передбачених розрахунком навантажень незначна і не порушує їх експлуатаційних якостей. У тих випадках, коли до конструкції пред'являються вимоги непроникності (резервуари, труби), коли конструкція забезпечена високоміцною арматурою або знаходиться в умовах агресивного середовища, поява тріщин або значне їх розкриття може призвести до втрати експлуатаційних характеристик. Щоб запобігти утворенню тріщин або обмежити ширину їх розкриття у розтягнутій зоні бетону під час виготовлення конструкції заздалегідь створюють значні стискаючі напруження шляхом натягу арматури. В умовах експлуатування такого елемента конструкції під навантаженням розтягальні напруження, що виникають, тільки погашають попереднє стиснення в бетоні. Тому утворення тут тріщин, або їх значне розкриття значно зменшується. Такі залізобетонні конструкції називають попередньо-напруженими.

Розрізняють два основних види залізобетонних конструкцій з попереднім напруженням: з натягом арматури до і після бетонування. У першому випадку арматуру попередньо розтягують і кінці її закріплюють на упорах ферми, потім заливають бетон. Після того, як бетонна суміш затверділа, кінці арматурних стрижнів звільняють від упорів. Другий спосіб передбачає виготовлення залізобетонних конструкцій з поздовжніми каналами, через які пропускають арматурні стрижні, потім їх розтягують і закріплюють на торцях конструкції. Канали заповнюють цементним розчином з метою захисту сталеві арматури від корозії.

Завдяки ефективному використанню високоміцної арматури в попередньо-напружених конструкціях, підвищеною їх жорсткістю і ряду інших переваг ці конструкції широко використовуються в будівництві.

Але попередньо-напружені залізобетонні конструкції мають багато недоліків. Одним з яких є, наприклад, перевантаження в них арматури, що за довготривалого статичного навантаження може привести, особливо за наявності дефектів типу тріщин, до сповільненого руйнування і непередбачуваної аварії.

Використовуючи енергетичний підхід [1,5] побудуємо розрахункову методику для визначення залишкового ресурсу арматури в попередньо-напружених залізобетонних конструкціях.

Розглянемо залізобетонну балку на двох опорах з попередньо напруженою арматурою, яка навантажена довготривалими рівнорозподіленими нормальними зусиллями інтенсивності  $p$ . Припустимо, що в результаті неякісного виготовлення арматури, впливу зовнішніх факторів або довготривалого навантаження в поперечному перерізі нижнього навантаженого розтягом арматурного стрижня діаметра  $D$  утворилася система поверхневих сегментних тріщин початкової глибини  $c_0$ , які при заданому довготривалому навантаженні надалі можуть поширюватися як тріщини низькотемпературної повзучості.

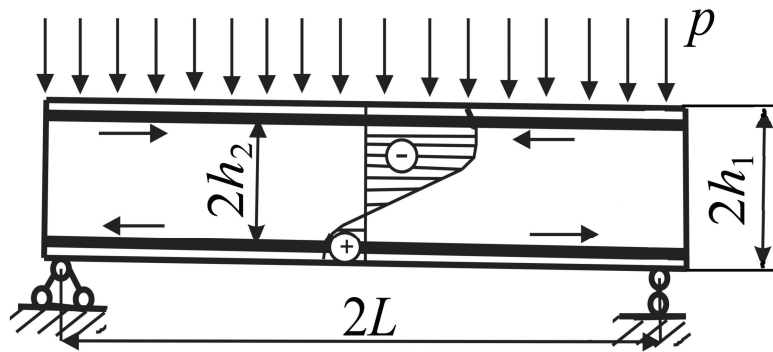


Рис. 1. Схема навантаження залізобетонної балки з наперед напруженою арматурою.

Задача полягає у визначенні такого часу  $t = t_*$ , за який тріщини підростуть від розміру  $c_0$  до критичного розміру  $c = c_*$  і арматура зруйнується.

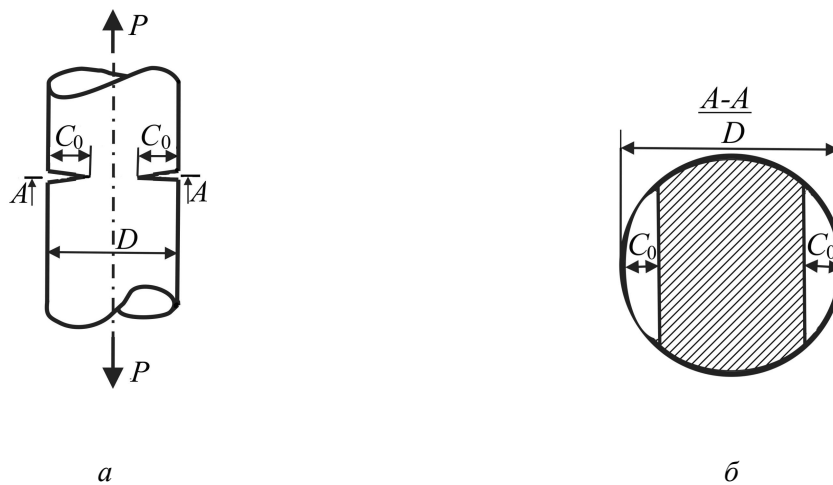


Рис. 2. Схема розтягу (а) циліндричного стрижня з двома сегментними тріщинами (б).

Розв'язок такої задачі здійснюємо на основі розрахункової моделі, яка запропонована в [1,5]. Узагальнивши задачу, розглянемо найгірше розміщення системи двох сегментних тріщин, тобто в одній площині, як показано на рис. 2. В результаті цього, а також вважаючи, що тріщини будуть поширюватися симетрично, задача зведеться до наступної системи рівнянь

$$\frac{dc}{dt} = \frac{A_{2t}(K_I^{2m} - K_{thc}^{2m})}{K_{cc}^{2m}(1 - K_I^2 K_{cc}^{-2})} \quad (1)$$

за початкових і кінцевих умов:

$$t = 0, c(0) = c_0; \quad t = t_*, c(t_*) = c_*, \quad K_I(p, c_*) = K_{cc}.$$

Тут  $K_I(p, c)$  – найбільше значення коефіцієнта інтенсивності напружень біля контурів сегментних тріщин, яке буде посередині хорд сегментів і для визначення якого необхідно знайти навантаження нижньої арматури в залізобетонній балці.

Необхідно визначити навантаження нижньої арматури в залізобетонній балці.

Потрібно зазначити, що при виготовленні залізобетонної балки арматуру натягають до початкового контрольованого напруження  $\sigma_{con}$  на опори форм, виконують бетонування, теплову

обробку і витримують бетон в формі до набуття ним необхідної передавальної міцності  $R_{bp}$ . В цьому стані проходять перші втрати  $\sigma_{los,1}$  попереднього натягу арматури [10].

Далі при звільненні з опорів форм і відпуску натягу арматури, яка щеплена з бетоном, проходить обтиснення бетону, в якому розвиваються деформації швидкої повзучості і проходять втрати обтиску бетону, а також попереднього натягу арматури  $\alpha\sigma_{bp}$ , тобто в ній залишаються напруження розтягу

$$\sigma_a = \sigma_{con} - \sigma_{los,1} - \alpha\sigma_{bp}. \quad (2)$$

Тут  $\sigma_{bp}$  – напруження обтиску бетону;  $\alpha$  – коефіцієнт зменшення напруження обтиску бетону під час його повзучості.

Прикладемо до балки рівномірно розподілене експлуатаційне навантаження інтенсивності  $p$ , тоді розподіл напружень в балці визначиться так, як показано на рис. 1.

Додаткові напруження розтягу  $\sigma_p$ , які виникають при цьому напружені в арматурі за рахунок експлуатаційного навантаження  $p$  буде визначитися на основі результатів [8] так

$$\sigma_p = 0,75ph_2L^2h_1^{-3}. \quad (3)$$

Тоді на основі співвідношень (2), (3) сумарне напруження  $\sigma_{ap}$  в арматурі можна буде визначити згідно формули

$$y_{ap} = y_{con} - y_{los,1} - \sigma_{yp} + \frac{3ph_2L^2}{4h_1^3}.$$

Аналогічним способом знайдемо залишковий ресурс залізобетонної балки за згину. Для розв'язання математичної задачі (1) і визначення часу до руйнування  $t = t_*$  нижньої арматури в залізобетонній балці необхідно знайти коефіцієнт інтенсивності напружень  $K_I$  біля контуру тріщини (рис. 2) в поперечному перерізі арматури. Таким чином, в даному випадку задача зведеться до визначення коефіцієнта інтенсивності напружень (КІН) біля контурів системи сегментних тріщин в поперечному перерізі циліндричного стрижня (рис. 2) діаметра  $D$ , який розтягується зусиллями  $P = 0.25\pi\sigma_{ap}D^2$ .

Дана задача розв'язана наближено у праці [7] методом граничної інтерполяції, де коефіцієнт інтенсивності напружень біля контурів тріщин визначається так

$$K_I = \frac{2\sqrt{2}P}{D\sqrt{D}} f(\varepsilon), \quad f(\varepsilon) = \frac{2,244\sqrt{\pi\varepsilon(1-\varepsilon)}}{S_1\sqrt{12,4247\varepsilon + 4(1-\varepsilon)}}, \quad \varepsilon = 2cD^{-1}, \quad (4)$$

$$S_1 = \pi - 2(\arcsin \sqrt{2\varepsilon - \varepsilon^2} - (1-\varepsilon)\sqrt{2\varepsilon - \varepsilon^2}).$$

Якщо підставити дане рівняння (4) в рівняння (1), то отримаємо таке співвідношення

$$\frac{dc}{dt} = \frac{A_{2t} \{ [2\sqrt{2}PD^{-1.5} f(\varepsilon)]^{2m} - K_{thc}^{2m} \}}{K_{cc}^{2m} [1 - 8P^2D^{-3} f^2(\varepsilon) K_{cc}^{-2}]} \quad (5)$$

за початкових і кінцевих умов:

$$t = 0, c(0) = c_0; \quad t = t_*, c(t_*) = c_*, \quad 2\sqrt{2}PD^{-1.5} f(\varepsilon_*) = K_{cc}.$$

Інтегруючи отриманий вираз (5) за вказаних початкових і кінцевих умов, отримаємо формулу для визначення часу до руйнування  $t = t_*$  нижньої арматури, а отже і в цілому залізобетонної балки

$$t_* = 0,5DK_{cc}^{2m} A_{2t}^{-1} \int_{\varepsilon_0}^{\varepsilon_*} \frac{[1 - 8P^2D^{-3} f^2(\varepsilon) K_{cc}^{-2}]}{[2\sqrt{2}PD^{-1.5} f(\varepsilon)]^{2m} - K_{thc}^{2m}} d\varepsilon.$$

Крім того, з отриманого результату, маючи діаметр та характеристики арматури, можна визначити  $K_{ce}$ ,  $K_{thc}$ , критичний розмір тріщини, а також визначити залежність залишкового ресурсу  $t_*$  арматури від початкового розміру тріщин  $c_0$ .

1. *Андрейків О.С.* Визначення довговічності пластин з системами тріщин в умовах дії довготривалого статичного розтягу і низькотемпературного поля. /О.С. Андрейків, Ю.Я. Матвіїв, Т.А. Крадінова // Мат. методи і фіз.-мех. поля. – 2011. – №4. – С 161–169.
2. *Гнідець Б.Г.* Залізобетонні конструкції з напружуваними стиками і регулюванням зусиль./Б.Г. Гнідець – Львів: Вид. Нац. універ. “Львівська політехніка”. – 2008. – 548 с.
3. *Байков В.Н.* Железобетонные конструкции: Общий курс: Учебное пособие для вузов./ В.Н. Байков, Э.Е. Сигалов – М.: Стройиздат. – 1991. – 767 с.
4. Визначення залишкового ресурсу арматури у попередньо напружених залізобетонних конструкціях/Ю.Я. Матвіїв, Т.А. Крадінова//Міжвузівський збірник “Наукові нотатки” - Луцьк, 2012. - №36. – С 195 -199.
5. Довговічність пластин з тріщинами за довготривалого статичного навантаження і локальної повзучості /О.С. Андрейків, В.Р. Скальський, Ю.Я. Матвіїв, Т.А. Крадінова // Фізико-хімічна механіка матеріалів. – 2012. – № 1. – С 39–46.
6. Каминский А. А. Механика разрушения вязкоупругих тел / А. А. Каминский. – К.: Наукова думка, 1980. – 157 с.
7. Kaminsky A. A. Subcritical Crack Growth in Polymer Composite Materials under Creep / А. А. Kaminsky // Advances in fracture resistance and structural integrity: Selected papers from the 8<sup>th</sup> International Conference on Fracture (8-14 June 1993), Kijev, Ukraine – Pergamon, 1994. – P. 513–520.
8. *Писаренко Г.С.* Справочник по сопротивлению материалов. /Г.С. Писаренко, А.П. Яковлев, В.В. Матвеев – К.: Наукова думка, 1988. – 734 с.
9. *Панько І.М.* Теоретичні основи інженерних методів для оцінки тріщинозміцності матеріалів і елементів конструкцій./І.М. Панько – Львів : ФМІ НАНУ.– 200.– 280 с.
10. *Скальський В.Р.* Методика оцінки характеристик повзучості матеріалів /В.Р. Скальський, Ю.Я. Матвіїв, Т.А. Крадінова //Машинознавство. – 2012. – №2.

УДК 681.586.773

Літвінов К.А.

Східноукраїнський національний університет ім. В. Даля

## МАТЕМАТИЧНІ МОДЕЛІ УЛЬТРАЗВУКОВОГО РІВНЕМІРА З КІЛЬЦЕВИМ РУХОМ ЕЛЕКТРИЧНОГО СИГНАЛУ

**Літвінов К.А. Математичні моделі ультразвукового рівнеміра з кільцевим рухом електричного сигналу.** Вимірювання рівня рідинних середовищ ультразвуковим методом засновано на перетворенні електричного збуджуючого імпульсу в ультразвукові коливання при його проходженні до п'єзокерамічного елемента, котрий розміщений у плаваючому на поверхні рідинного середовища поплавку. Досліджені перетворення електричного збуджуючого імпульсу в енергію електромагнітного поля та в ультразвукові коливання. Показано, що імпульс ультразвукових коливань сприймається таким же п'єзокерамічним елементом, котрий знаходиться в приймачі цього імпульсу. Вимірювання рівня рідинного середовища здійснюється при одноразовому проходженні ультразвукових коливань від випромінювача до приймача.

**Ключові слова:** вимірювання, перетворення, ультразвук, метод, імпульс, п'єзокераміка, випромінювач, газ, рідина.

**Литвинов К. А. Математические модели ультразвукового уровнемера с круговым движением электрического сигнала.** Измерение уровня жидких сред ультразвуковым методом основано на преобразовании электрического возбуждающего импульса в ультразвуковые колебания при его прохождении к пьезокерамическому элементу, который установлен у плавающем на поверхности жидкостной среды поплавке. Исследованы процессы преобразования возбуждающего импульса в энергию электромагнитного поля и в ультразвуковые колебания. Показано, что импульс ультразвуковых колебаний воспринимается таким же пьезокерамическим элементом, который находится в приемнике этого импульса. Измерение уровня жидкой среды осуществляется при одноразовом прохождении ультразвуковых колебаний от излучателя до приемника.

**Ключевые слова:** измерение, преобразование, ультразвук, метод, импульс, пьезокерамика, излучатель, газ, жидкость.

**Litvinov K. A. Mathematical models of ultrasonic level meter with a circular movement of the electrical signal.** Measurement of liquid media by ultrasound method is based on the conversion of electrical excitation pulse in the ultrasonic vibrations as it passes to the piezoceramic element, which is in floating on the surface of the liquid medium float. Researched converting electrical excitation pulse energy in the electromagnetic field and ultrasonic vibrations. It is shown that the pulse ultrasonic vibrations perceived in the same piezoceramic element, which is the receiver of the pulse. Measurement of liquid medium is performed in a single pass ultrasonic vibrations from the emitter to the receiver.

**Keywords:** measurement, transformation, ultrasound, method, impulse, piezoceramic, emitter, gas, liquid.

**Вступ.** Як відомо [1-3], принцип роботи ультразвукових засобів контролю рівня (УЗКР) рідинних середовищ (РС) заснований на тому, що при подачі на п'єзокерамічний елемент (ПКЕ), який має форму круглої мембрани, електричного збуджуючого імпульсу (ЕЗІ) в ньому створюється електромагнітне поле (ЕМП), котре перетворюється в електродинамічне зусилля, що призводить до відхилення цього елемента від стану рівноваги. Після припинення дії ЕЗІ ПКЕ за аналогією пласкої мембрани здійснює вільні коливання ультразвукової частоти. Створені ультразвукові коливання (УЗК) у формі відповідного імпульсу передаються розділювальній металевій мембрані і випромінюються у відповідне середовище – газове чи рідинне. Так як швидкість розповсюдження УЗК залежить від фізичних параметрів середовища та стану його чистоти, то УЗКР будуються за принципом випромінювання УЗК у газове середовище (ГС). Як правило, при їх побудові використовується принцип подвійного руху імпульсу УЗК – прямого руху від його випромінювача до поверхні рідинного середовища (РС) і зворотного руху від цієї поверхні до приймача імпульсу, яким служить той же випромінювач. При цьому вимірюється час від моменту подачі ЕЗІ на ПКЕ до моменту сприйняття відбитого від поверхні РС імпульсу УЗК. Рівень рідини в технологічному апараті розраховується в блоці обробки вимірювальної інформації (БОВІ) за відповідним алгоритмом, основою котрого є наступне рівняння [4]:

$$L = H - 0,5ct_2, \quad (1)$$

де  $L$  - рівень контрольованого РС;  $H$  - висота технологічного апарату;  $t_2$  - час, за який імпульс УЗК проходить подвійну відстань від випромінювача до приймача;  $c$  - швидкість розповсюдження ультразвуку в ГС.

Такий принцип вимірювального контролю має низку недоліків, які суттєво впливають як на діапазон вимірювального контролю (ДВК), так і на точність УЗКР. Множник «0,5» у рівнянні (1) враховує подвійність ходу імпульсу УЗК. Якщо би цей імпульс проходив одну відстань, то ДВК

можна було би розширити у двічі. Відомо [5, 6], що при проходженні імпульсом подвійної відстані в ГС мають місце відповідні реологічні перетворення, котрі призводять до деформації обвідної ультразвукового сигналу (УЗС), а відповідно, до похибки вимірювального контролю (ВК). Окрім того, у науковій літературі не виявлено досліджень щодо впливу РС на відбитий імпульс УЗС. Але можна стверджувати, що енергетична потужність і форма падаючого на поверхню РС імпульсу УЗС будуть відрізнятися від цих параметрів відбитого сигналу. Причому ступінь поглинання енергії УЗС і його форма залежатимуть від фізичних параметрів РС. Тобто має місце методична похибка ВК, яка обумовлена параметрами РС. Окрім того, такі УЗКР мають достатньо велику зону нечутливості, яка у багатьох випадках досягає до 0,6 м [7, 8], і обумовлена накладанням в часі випроміненого та сприйнятого УЗС. З метою підвищення точності ВК у багатьох випадках використовують реперні пристрої, котрі являють собою пластини або кільця, закріплені на деякій відстані від випромінювача в ГС. При кожному вимірюванні випромінений імпульс УЗК спочатку відбивається від реперного пристрою і сприймається приймачем, котрим служить той же випромінювач, і запам'ятовується в БОВІ. Той же випромінений імпульс УЗК проходить товщину ГС, відбивається від поверхні РС і теж сприймається тим же приймачем. Сприйнятий сигнал запам'ятовується у БОВІ, в якому ці два сигнали (реперний і робочий) підлягають фільтрації, порівнянню тощо. У деякій науковій літературі вказується, що УЗКР з реперним пристроєм зменшують похибку ВК, обумовлену зміною параметрів газу, як температури, тиску та його компонентного складу [9, 10]. Експериментальні дослідження показують, що зменшення впливу цих параметрів ГС є незначним, за рахунок того, що поглинання інтенсивності УЗК в ГС відбувається за нелінійним законом Ламберта-Бера [11, 12]. Враховуючи недоліки сучасних УЗКР, є актуальною задача покращення метрологічних характеристик ультразвукового методу вимірювального контролю рівня рідинних середовищ.

**Мета статті і постановка досліджень.** Метою статті є теоретичні дослідження нового принципу вимірювального контролю рівня рідинних середовищ ультразвуковим методом, побудованого за кільцевим рухом ультразвукового сигналу. Запропонований ультразвуковий метод дозволяє підвищити точність ВК рівня практично в два рази. Принцип роботи такого УЗКР рівня полягає в наступному (рис. 1).

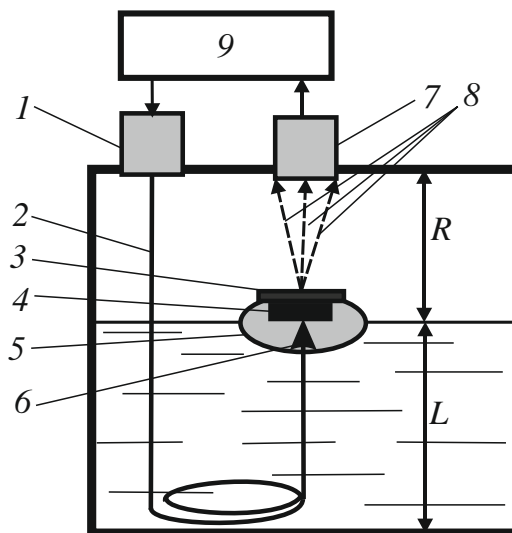


Рис. 1. Схема ультразвукового УЗКР з одноразовим рухом УЗК

Випромінювач УЗК, який являє собою ПКЕ круглої форми (поз. 4), розташовується в поплавку (поз. 5), який плаває на поверхні РС. ЕЗІ від джерела 1 по ізолюваному гнучкому кабелю 2 подається через електрод 6 до ПКЕ 4, у результаті чого створюється імпульс УЗК, котрий через розділювальну металеву мембрану 3 передається в ГС товщиною  $R$ . Пройшовши ГС, імпульс УЗК 8 сприймається приймачем 7, котрий перетворює ці коливання в електрорушійну силу. Остання поступає в БОВІ (поз.9). Таким чином, імпульс УЗК розповсюджується в ГС тільки один раз, що приводить до можливого розширення ДВК і зменшення зони нечутливості практично вдвічі, а відповідно, зменшення похибки ВК.

**Фізичні моделі перетворень в УЗКР.** Електричний збуджуючий імпульс, як правило, формується конденсатором, який знаходиться в блоку джерела живлення (БДЖ) 1. Як показують

результати експериментальних досліджень, форма імпульсу в гнучкому кабелі 2 практично не змінює своєї форми. Тому прийmemo, що ЕЗІ, який формується БДЖ і подається на ПKE є незмінним. Так як процес ВК є циклічним, то час такого циклу можна вважати тривалістю інтегральної імпульсної дельта-функції Дірака з джерелом електричної енергії напругою  $U_E$  та стоком цієї енергії у формі імпульсу УЗК інтенсивністю  $I_0$ , як показано на рис. 2, а, а її графічне зображення у формі графіків незворотних реологічних перетворень на рис. 2, б. З фізичної моделі видно, що напруга, яка подається в БДЖ, перетворюється в ньому на імпульс напруги  $U_E$  ЕЗІ. При дії цього імпульсу на ПKE, за рахунок створюваного електромагнітного зусилля пластина ПKE відхиляється від свого усталеного значення на величину  $x$ . Фактично це відхилення є джерелом імпульсу кількості руху, яке створює УЗК. Останні з інтенсивністю  $I_0$  випромінюються в ГС, у котрому його енергія дещо поглинається згідно з законом Ламберта-Бера, сприймається ПKE приймача і перетворюється в ньому в електрорушійну силу ультразвукової частоти, яка направляєється в БОВІ.

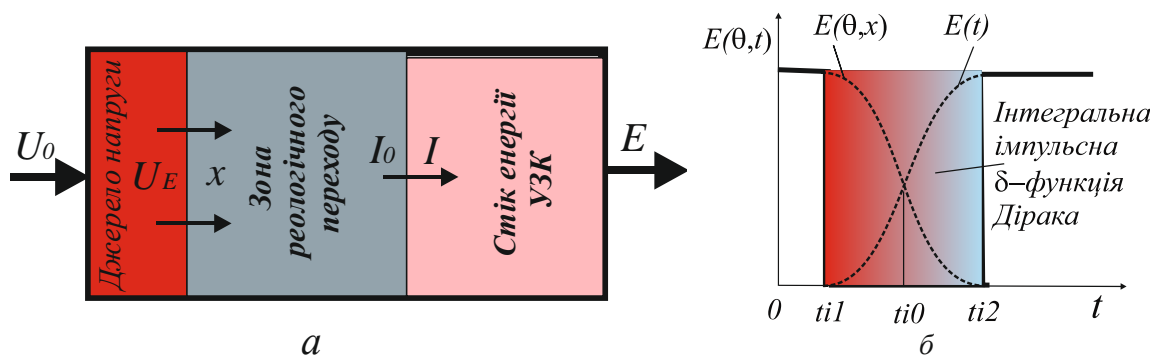


Рис. 2. Узагальнена фізична модель УЗКР

Виходячи з описання фізичної моделі, можна відмітити наступні реологічні переходи з відповідними перетвореннями:

- імпульсу електричної енергії напругою  $U_E$  у напруженість електромагнітного поля  $H_E$  (ЕМП);
- напруженості  $H_E$  ЕМП в електродинамічне зусилля  $N_E$  (ЕДЗ);
- зусилля  $N_E$  у механічне переміщення  $x$  пластини ПKE;
- потенціальної енергії переміщеної на відстань  $x$  пластини ПKE у кінетичну енергію вільних загасаючих коливань ультразвукової частоти;
- перенесення енергії імпульсу УЗК в газовому середовищі;
- перетворення енергії імпульсу УЗК приймачем і створення високочастотної ЕРС з допомогою вимушеного руху пластини ПKE.

Прийmemo до уваги, що кожний реологічний перехід являє собою інтегральну імпульсну дельта-функцію Дірака. Якщо функція перетворення  $f_i(t)$  описує процес перенесення імпульсу енергії чи кількості руху зі спадковістю, то інтегральна імпульсна дельта-функція Дірака є асимптотичною та описується таким рівнянням

$$\int_{t_{i+0}}^{t_{2i}} f_i(\xi) \delta(\xi - t_{i0}) d\xi = \begin{cases} 0 & t_{i0} < t_{1i}, \quad t_{i0} \geq t_{2i} \\ f(t_{i0} + 0) & t_{1i} < t_{i0} < t_{2i} \end{cases}, \quad (2)$$

де  $\xi$  - деяка змінна, яка пропорційна часу перетворення  $t$ ;  $\delta(\xi - t_{i0})$  - ядро лінійного інтегрального перетворення;  $t_{i0}$  - середнє значення часу фазового переходу;  $t_{1i}$  - час початку переходу енергії від джерела в зону реологічного перетворення (ЗРП);  $t_{2i}$  - час виходу перетвореної енергії з ЗРП.

Як показано в [13, 14], функція перетворення імпульсу енергії описується наступним нелінійним диференціальним рівнянням:

$$\frac{\partial E_i(\theta, x)}{\partial \theta} + D_i \frac{\partial^2 E_i(\theta, x)}{\partial x^2} + v_i \frac{\partial E_i(\theta, x)}{\partial x} = \gamma_j(t), \quad (3)$$

де  $E_i(\theta, x)$  - енергія перетворення, яка змінюється за час  $\theta$  на елементарному відрізку  $x$ ;  $D_i$  - коефіцієнт перетворення  $i$ -го реологічного переходу;  $v_i$  - лінійна швидкість реологічного переходу;  $\gamma_j(t)$  - функція швидкості стоку перетвореної енергії.

Якщо енергія джерела приймається за безмежною, тобто передається в ЗРП безперервно згідно з рівнянням (3), то у відповідності з законом зберігання маси, енергії та кількості руху стік деякої перетвореної енергії  $E_j(t)$  теж буде безперервним і матиме накопичувальний характер. Виходячи з цього, а також з фізичних моделей реологічних переходів швидкість стоку енергії описуватиметься такими рівняннями:

- при одностадійному процесі перетворення енергії  $E_i(\theta, x)$  в енергію  $E_j(t)$ :

$$\gamma_j(t) = \tau_1 \frac{d^2 E_j(t)}{dt^2} + \frac{dE_j(t)}{dt}; \quad (4)$$

- при двостадійному процесі перетворення енергії  $E_i(\theta, x)$ :

$$\gamma_j(t) = \tau_{22} \frac{d^3 E_j(t)}{dt^3} + \tau_{21} \frac{d^2 E_j(t)}{dt^2} + \frac{dE_j(t)}{dt}. \quad (5)$$

де  $\tau_1, \tau_{21}, \tau_{22}$  - сталі часу відповідних процесів реологічних перетворень.

Прийmemo до уваги, що процеси реологічного перетворення ЕЗІ в напруженість ЕКМ ПКЕ і останньої в ЕДЗ є практично лінійними та описуються динамічною ланкою нульового порядку. Тому ці два реологічні переходи можна описати наступним рівнянням:

$$N_E = k_H k_N U_E. \quad (6)$$

Процес реологічного перетворення ЕДЗ у лінійне переміщення пластини ПКЕ є достатньо інерційним, тому для нього, враховуючи рівняння (3) і (4) можна записати наступне нелінійне диференціальне рівняння:

$$\frac{\partial N_E(\theta, x)}{\partial \theta} + D_i \frac{\partial^2 N_E(\theta, x)}{\partial x^2} + v_i \frac{\partial N_E(\theta, x)}{\partial x} = \tau_x \frac{d^2 E_{\Pi}(t_1)}{dt_1^2} + \frac{dE_{\Pi}(t_1)}{dt_1}, \quad (7)$$

де  $\tau_x = \rho_{\Pi} v_{\Pi} / D_{\Pi} S_{\Pi}$  - стала часу ПКЕ;  $\rho_{\Pi}$  - щільність керамічного матеріалу ПКЕ;  $v_{\Pi} = v_i$  - лінійна швидкість руху пластини ПКЕ;  $D_{\Pi}$  - жорсткість пластини;  $S_{\Pi}$  - активна площа пластини ПКЕ;  $E_{\Pi}$  - потенціальна енергія, яка накопичується пластиною ПКЕ.

Згідно з теорією реологічних переходів [15] і методом нульового градієнта [16] рівняння (7) приводиться до системи наступних диференціальних рівнянь:

$$\frac{\partial N_E(\theta, x)}{\partial \theta} + D_i \frac{\partial^2 N_E(\theta, x)}{\partial x^2} + v_i \frac{\partial N_E(\theta, x)}{\partial x} = 0; \quad (8)$$

$$\tau_x \frac{dE_{\Pi}(t_1)}{dt_1} + E_{\Pi}(t_1) = k_{\Pi} N_E(\theta, x), \quad (9)$$

де  $k_{\Pi}$  - коефіцієнт передачі.

З достатньою для практики точністю можна прийняти, що  $\partial \theta = (1/v_{\Pi}) \partial x$ . Тоді (8) приводиться до наступного лінійного диференціального рівняння:

$$\frac{D_i}{2v_{\Pi}} \frac{\partial N_E(\theta, x)}{\partial x} + N_E(\theta, x) = k_H k_N U_E. \quad (10)$$



При нульових початкових умовах рішенням рівняння (10) буде:

$$N_E(\theta, x) = k_H k_N U_E [1 - \exp(-2v_{\Pi} x / D_i)]. \quad (11)$$

При нульових початкових умовах рівняння (9) має наступне рішення

$$E_{\Pi}(t) = k_{\Pi} N_E(\theta, x) [1 - \exp(-t / \tau_x)]. \quad (12)$$

Позначимо відхилення пластини ПKE  $x(t_1) = k_x E_{\Pi}(t_1)$ . Тоді, підставивши (11) у рівняння (12), отримуємо математичну модель в аналітичній формі для реологічного перетворення напруги ЕЗІ в переміщення пластини ПKE у такій формі:

$$x_{\Pi}(t) = k_x k_{\Pi} k_H k_N U_E [1 - \exp(-2v_{\Pi} x / D_i)] [1 - \exp(-t / \tau_x)]. \quad (13)$$

Для процесу перенесення імпульсу руху пластини можна прийняти, що  $v_{\Pi} x = S_{\Pi} / \tau_x$ . Тоді, позначивши  $k_U = k_x k_{\Pi} k_H k_N$ , маємо

$$x_{\Pi}(t) = k_U U_E [1 - \exp(-2S_{\Pi} / \tau_x D_i)] [1 - \exp(-t / \tau_x)]. \quad (14)$$

З рівняння (14) випливає два важливі висновки:

- можливість побудови УЗКР з кільцевим рухом УЗС за зміною напруги ЕЗІ;
- так як при роботі УЗКР використовують максимальне відхилення пластини ПKE, що забезпечує максимальну амплітуду УЗК, то рівняння (14) при  $t_1 \rightarrow \infty$ , розкладання в ряд Тейлора та обмеження тільки лінійною складовою приводиться до вигляду:

$$x_{\Pi} = 2k_U U_E S_{\Pi} / \tau_x D_i. \quad (15)$$

У рівняння (15) підставимо  $\tau_x = \rho_{\Pi} v_{\Pi} / D_{\Pi} S_{\Pi}$  і після відповідних перетворень, приймаючи, що швидкість  $v_{\Pi} \approx x_{\Pi} / \tau_x$ , отримуємо:

$$x_{\Pi} \approx S_{\Pi} \sqrt[3]{(2k_U U_E / D_i)^2 (D_{\Pi} / \rho_{\Pi})}. \quad (16)$$

Рівняння можна використовувати для попереднього вибору ПKE за його фізичними параметрами (жорсткістю та щільністю п'єзокерамічного матеріалу) та активною площиною пластини. З цього рівняння також видно, що залежність  $x_{\Pi} = f(U_E)$  є нелінійною. Потенціальна енергія пластини ПKE при припиненні на неї дії ЕЗІ, перетворюється в кінетичну енергію вільного її руху, створюючи імпульс УЗК. При виробництві ультразвукових перетворювачів у багатьох випадках з метою скорочення часу коливання пластини ПKE, навколишній її об'єм часто заповнюють спеціальною рідиною, наприклад, силіконовою, котра мало впливає на початкову амплітуду коливань, але призводить до швидкого загасання УЗК. Зменшення тривалості імпульсу УЗК приводить до зменшення зони нечутливості УЗКР.

Реологічне перетворення потенціальної енергії ПKE в УЗК можна описати наступним нелінійним диференціальним рівнянням:

$$\frac{\partial E_k(\mathcal{G}, x)}{\partial \mathcal{G}} + D_k \frac{\partial^2 E_k(\mathcal{G}, x)}{\partial x^2} + v_k \frac{\partial E_k(\mathcal{G}, x)}{\partial x} = \tau_{22}^2 \frac{d^3 y(t)}{dt^3} + \tau_{21} \frac{d^2 y(t)}{dt^2} + \frac{dy(t)}{dt}, \quad (17)$$

де  $E_k(\mathcal{G}, y)$  - зміна кінетичної енергії за час  $\mathcal{G}$  за координатою;  $D_k$  - коефіцієнт перетворення кінетичної енергії;  $v_k$  - швидкість зміни цієї енергії.

Враховуючи, що цей реологічний перехід те ж являє собою інтегральну імпульсну дельта-функцію Дірака, то рівняння (17) розділяється на систему наступних диференціальних рівнянь:

$$\frac{\partial E_k(\mathcal{G}, x)}{\partial \mathcal{G}} + D_k \frac{\partial^2 E_k(\mathcal{G}, x)}{\partial x^2} + v_k \frac{\partial E_k(\mathcal{G}, x)}{\partial x} = 0; \quad (18)$$

$$\tau_{22}^2 \frac{d^2 y(t)}{dt^2} + \tau_{21} \frac{dy(t)}{dt} + y(t) = k_y E_k(\vartheta, x), \quad (19)$$

де  $\tau_{21} = k_1 / D_{\Pi}$ ;  $\tau_{22} = \sqrt{m / D_{\Pi}}$  - сталі часу;  $m$  - маса коливальної системи;  $D_{\Pi}$  - жорсткість системи;  $k_1$  - коефіцієнт демпфірування її руху (приймається, що пластина ПКЕ жорстко кріпиться до металевої захисної мембрани і створює коливальну систему).

Приймаючи до уваги, що час  $\vartheta$  зміни кінетичної енергії пластини пропорційний зміні положення пластини  $y$ , тобто  $\partial \vartheta = (1/v_k) \partial y$ , з (18) отримуємо:

$$\frac{D_k}{2v_k} \frac{\partial E_k(\vartheta, x)}{\partial x} + E_k(\vartheta, x) = k_k x_0, \quad (20)$$

де  $k_k$  - коефіцієнт передачі кінетичної енергії;  $x_0$  - максимальне відхилення мембрани.

Рішенням рівняння (20) при нульових початкових умовах буде:

$$E_k(\vartheta, x) = k_k x_0 \exp(-2v_k x / D_k). \quad (21)$$

Так як заздалегідь відомо, що перехідний процес стоку є коливально-загасаючим, то при нульових початкових умовах рівняння (19) набуває такого рішення:

$$y(t) = k_y E_k(\vartheta, x) \exp(\alpha t) [\cos(\omega t) + (\alpha/\omega) \sin(\omega t)], \quad (22)$$

де  $\alpha = -\tau_{21} / 2\tau_{22}^2$  - ступінь загасання УЗК;  $\omega = \sqrt{(1/\tau_{22}^2) - (\tau_{21}/2\tau_{22}^2)^2}$  - власна частота коливань.

Підставивши рівняння (21) у (22), отримуємо математичну модель в аналітичній формі для механічних УЗК, котрі випромінюються в ГС апарату з рідиною:

$$y(t) = k_y k_k x_0 \exp(-2v_k x / D_k) \exp(\alpha t) [\cos(\omega t) + (\alpha/\omega) \sin(\omega t)]. \quad (23)$$

У рівнянні (23) позначимо  $x_{\Pi} = k_y k_k x_0 \exp(-2v_k x / D_k)$  і запишемо його таким чином:

$$y(t) = x_{\Pi} \exp(\alpha t) [\cos(\omega t) + (\alpha/\omega) \sin(\omega t)]. \quad (24)$$

Інтенсивність  $I_0$  випромінювання УЗК пропорційна кінетичній енергії, яка витрачається на пружну деформацію коливальної системи [17, 18]:

$$W_{k0} = 0,5 m_c v_k^2 = 0,5 m_c \left( \frac{dy(t)}{dt} \right)^2. \quad (25)$$

Диференціюючи рівняння (24) за часом  $t$  і підставляючи в (26), маємо:

$$W_{k0} = 0,5 m_c x_{\Pi}^2 \alpha^2 [\exp(\alpha t) \sin(\omega t)]^2 \left\{ (\alpha/\omega) [1 - (\omega/\alpha)^2] \right\}^2. \quad (26)$$

Підставимо рівняння (16) в (26). Тоді математична модель для енергії, яка випромінюється в ГС, приймає наступну форму:

$$W_{k0} = m_c S_{\Pi}^2 \alpha^2 (k_U U_E / D_i) \sqrt{(2k_U U_E / D_i)(D_{\Pi} / \rho_{\Pi})^2} * \\ * [\exp(\alpha t_2) \sin(\omega t_2)]^2 \left\{ (\alpha/\omega) [1 - (\omega/\alpha)^2] \right\}^2. \quad (27)$$

У рівнянні (27) введемо наступне позначення:  $B_0 = k_U m_c S_{\Pi}^2 (\alpha^2 / D_i) \sqrt{(2k_U / D_i)(D_{\Pi} / \rho_{\Pi})}$ . Тоді отримуємо:

$$W_{k0} = B_0 U_E \sqrt{U_E} [\exp(\alpha t) \sin(\omega t)]^2 \left\{ (\alpha/\omega) [1 - (\omega/\alpha)^2] \right\}^2. \quad (28)$$

З рівняння (28) випливає, що випромінююча енергія УЗК дещо зменшується з підвищенням напруги ЕЗІ, що є важливим при проектуванні УЗКР зі змінним збуджуючим сигналом.

Розглянемо процес розповсюдження ультразвукової енергії в ГС з точки зору теорії реологічних перетворень. Енергія, яка поглинається в ГС, може бути описана наступним нелінійним диференціальним рівнянням:

$$\frac{\partial W_k(\zeta, R)}{\partial \zeta} + D_c \frac{\partial^2 W_k(\zeta, R)}{\partial R^2} + v_c \frac{\partial W_k(\zeta, R)}{\partial R} = \tau_c \frac{d^2 W_c(t)}{dt^2} + \frac{dW_c(t)}{dt}, \quad (29)$$

де  $W_k(\zeta, R)$  - енергія УЗК, яка розповсюджується в ГС, за час  $\zeta$  на відстані  $R$  від випромінювача;  $W_c(t)$  - енергія, котра сприймається приймачем УЗК;  $\tau_c$  - стала часу.

Реологічне перетворення, яке описується рівнянням (29) є інтегральною імпульсною дельта-функцією Дірака і згідно з методом нульового градієнта його можна розділити на наступну систему диференціальних рівнянь:

$$\frac{\partial W_k(\zeta, R)}{\partial \zeta} + D_c \frac{\partial^2 W_k(\zeta, R)}{\partial R^2} + v_c \frac{\partial W_k(\zeta, R)}{\partial R} = 0; \quad (30)$$

$$\tau_c \frac{dW_c(t)}{dt} + W_c(t) = k_c W_c(\zeta, R). \quad (31)$$

Залежність між часом  $\zeta$  і відстанню  $R$  є нелінійною та описується наступним рівнянням:  
 $\zeta = (R/v_c)[1 - \exp(-\varepsilon_\lambda R Q_c)]$ . Похідна цієї функції має вигляд:

$$\partial \zeta = \frac{1}{v_c} [1 - (1 - \varepsilon_\lambda R_L Q_c) \exp(-\varepsilon_\lambda R_L Q_c)] \partial R, \quad (32)$$

де  $R_L$  - відстань від випромінювача УЗК до поверхні РС.

Підставимо (32) у рівняння (30) і в результаті відповідних перетворень отримуємо:

$$\frac{D_c}{v_c} \left[ \frac{1 - (1 - \varepsilon_\lambda R_L Q_c) \exp(-\varepsilon_\lambda R_L Q_c)}{2 - (1 - \varepsilon_\lambda R_L Q_c) \exp(-\varepsilon_\lambda R_L Q_c)} \right] \frac{\partial W_k(R)}{\partial R} + W_k(R) = k_w W_{k0}. \quad (33)$$

При нульових початкових умовах рішенням рівняння (33) буде:

$$W_k(R) = k_w W_{k0} \exp \left\{ -R_L \frac{v_c}{D_c} \left[ \frac{2 - (1 - \varepsilon_\lambda R_L Q_c) \exp(-\varepsilon_\lambda R_L Q_c)}{1 - (1 - \varepsilon_\lambda R_L Q_c) \exp(-\varepsilon_\lambda R_L Q_c)} \right] \right\}. \quad (34)$$

Рішенням рівняння (31) при нульових початкових умовах буде:

$$W_c(t_3) = k_c W_c(\zeta, R) \exp(-t/\tau_3). \quad (35)$$

Підставивши (34) у рівняння (35) і враховуючи, що стала часу  $\tau_c = R_L/v_c$ , отримуємо математичну модель реологічного перетворення УЗК в ГС у такому вигляді:

$$W_c(t_3) = k_c k_w W_{k0} \exp \left\{ -R_L \frac{v_c}{D_c} \left[ \frac{2 - (1 - \varepsilon_\lambda R_L Q_c) \exp(-\varepsilon_\lambda R_L Q_c)}{1 - (1 - \varepsilon_\lambda R_L Q_c) \exp(-\varepsilon_\lambda R_L Q_c)} \right] \right\} \exp(-t v_c / R_L). \quad (36)$$

Аналізуючи рівняння (36) бачимо, що множник  $\exp(-\varepsilon_\lambda R_L Q_c)$  є відносним значенням інтенсивності, де  $I_0$  - початкове її значення. Підставивши у (36) рівняння (28), отримуємо математичну модель реологічного перетворення в такій формі:

$$W_c(t_3) = k_c k_w B_0 U_E \sqrt[3]{U_E} [\exp(\alpha t) \sin(\omega t)]^2 \left\{ (\alpha/\omega) [1 - (\omega/\alpha)^2] \right\}^* \\ * \exp \left\{ -R_L \frac{v_c}{D_c} \left[ \frac{2 - (1 - \varepsilon_\lambda R_L Q_c) \exp(-\varepsilon_\lambda R_L Q_c)}{1 - (1 - \varepsilon_\lambda R_L Q_c) \exp(-\varepsilon_\lambda R_L Q_c)} \right] \right\} \exp(-t v_c / R_L). \quad (37)$$

Енергія  $W_c(t) = f(R_L)$ , досягнувши мембранного блоку приймача УЗК, створює механічне зусилля на мембрану з жорстко закріпленою до неї пластиною ПКЕ, котра почне рухатися з частотою цих коливань, деформуючи ПКЕ, відповідно, створюючи електрорушійну силу (ЕРС), яка з допомогою електроду передається до БОВІ для обробки та розрахунку рівня РС. Це перетворення є реологічним і може бути описане наступним нелінійним диференціальним рівнянням:

$$\frac{\partial W_c(\zeta, z)}{\partial \zeta} + D_M \frac{\partial^2 W_c(\zeta, z)}{\partial z^2} + v_M \frac{\partial W_c(\zeta, z)}{\partial z} = \tau_x \frac{d^2 e(t)}{dt^2} + \frac{de(t)}{dt}, \quad (38)$$

де  $W_c(\zeta, z)$  - енергія УЗК, яка перетворюється в мембранному блоці приймача за час  $\zeta$  на відстані  $z$ ;  $D_M$  - коефіцієнт перетворення;  $v_M$  - швидкість перенесення енергії;  $e(t)$  - ЕРС, яка створюється в пластині ПКЕ за час  $t$ .

Так як рівняння (38) описує інтегральну імпульсну дельта-функцію Дірака, то згідно до методу нульового градієнта його можна подати наступною системою диференціальних рівнянь:

$$\frac{\partial W_c(\zeta, z)}{\partial \zeta} + D_M \frac{\partial^2 W_c(\zeta, z)}{\partial z^2} + v_M \frac{\partial W_c(\zeta, z)}{\partial z} = 0; \quad (39)$$

$$\tau_x \frac{de(t)}{dt} + e(t) = k_{\Pi} W_c(\zeta, z). \quad (40)$$

Якщо прийняти, що механічне зусилля  $W_c(\zeta, z) = k_M N_M(\zeta, z)$ , яке створюється енергією УЗК, а залежність  $\partial \zeta = f(z)$  є лінійною, то рівняння (39) приводиться до наступного лінійного диференціального рівняння:

$$\frac{D_M k_{\zeta}}{(1 + k_{\zeta} v_M)} \frac{\partial N_M(z)}{\partial z} + N_M(z) = k_N N_{M0}, \quad (41)$$

де  $k_{\zeta}$  - стала.

При нульових початкових умовах рішенням рівняння (41) буде:

$$N_M(\zeta, z) = k_M N_M(t) \left[ 1 - \exp\left(-z \frac{(1 + k_{\zeta} v_M)}{D_M k_{\zeta}}\right) \right], \quad (42)$$

де  $N_M(t) = W_c(t)/k_M$  - механічне зусилля, яке створюється енергією УЗК.

Так як випромінювач УЗК та приймач є одним і тим же ПКЕ, то стала часу  $\tau_x$  буде такою ж, як і для рівняння (7). При початкових умовах рішення рівняння (40) буде наступним:

$$e(t) = k_e W_c(\zeta, z) \exp(-t/\tau_x). \quad (43)$$

Приймаючи, що  $N_M(t) = W_c(t)/k_M$  і враховуючи рівняння (13) і (37), отримуємо наступну математичну модель УЗКР з кільцевим рухом електричного сигналу в такій формі:

$$e(t, R_L) = K_{\Pi} U_E \sqrt[3]{U_E} \left[ 1 - \exp\left(-\frac{t}{\tau_x}\right) \right] \left[ \exp(\alpha t) \sin(\omega t) \right]^2 \exp\left(-2t \frac{v_c}{R_L}\right) \left[ 1 - \exp\left(-\frac{2S_{\Pi}}{\tau_x D_i}\right) \right] * \quad (44)$$

$$* \exp\left\{ -R_L \frac{v_c}{D_c} \frac{[2 - (1 - \varepsilon_{\lambda} R_L Q_c) \exp(-\varepsilon_{\lambda} R_L Q_c)]}{[1 - (1 - \varepsilon_{\lambda} R_L Q_c) \exp(-\varepsilon_{\lambda} R_L Q_c)]} \right\} \left[ 1 - \exp\left(-R_L \frac{(1 + k_{\zeta} v_M)}{D_M k_{\zeta}}\right) \right],$$

де  $K_{\Pi} = k_e k_M k_c k_w B_0 (\alpha/\omega) [1 - (\omega/\alpha)^2]$  - коефіцієнт передачі УЗКР.

Як правило, на практиці УЗКР виконуються за часом проходження УЗК від моменту подачі ЕЗІ до моменту, коли ЕРС  $e(t, R_L)$  на вході стане рівною деякій опорній напрузі  $u_0$ . Тоді, позначивши  $e(t, R_L) = u_0$ ,  $B_1 = K_{\Pi} U_E \sqrt[3]{U_E} \left[ 1 - \exp\left(-\frac{2S_{\Pi}}{\tau_x D_i}\right) \right]$  і прийнявши, що коефіцієнт поглинання УЗК ГС є незначним, яким можна знехтувати, з (44) отримуємо:

$$u_0 = B_1 [\sin(\omega t_k)]^2 \left[ 1 - \exp\left(-R_L \frac{(1 + k_{\xi} v_M)}{D_M k_{\xi}}\right) \right] * \left[ 1 - \exp\left(-\frac{t}{\tau_x}\right) \right] \left[ \exp(\alpha t) \right]^2 \exp\left(-2t \frac{v_c}{R_L}\right) \exp\left\{-R_L 2 \frac{v_c}{D_c}\right\}. \quad (45)$$

Позначимо  $\Omega = u_0 / B_1 [\sin(\omega t_k)]^2$ , розкладемо функції  $\exp(x)$  у ряд Тейлора та обмежимося їх лінійними складовими. Після відповідних перетворень отримуємо наступну спрощену математичну модель УЗКР для часу проходження УЗК відстані в ГС у такому вигляді:

$$t = R_L \left[ \frac{\Omega (D_M k_{\xi} \tau_x D_c) - (D_c - 2v_c R_L)}{v_c (1 + k_{\xi} v_M) (D_c - 4v_c R_L)} \right]. \quad (46)$$

Як показали експериментальні дослідження промислових УЗКР залежність  $t = f(R_L)$  є практично лінійною до  $R_L > 20m$ . Тому можна вважати, що  $D_c \gg v_c R_L$  і тоді рівняння (46) приводиться до такої форми:

$$t = R_L \left[ \frac{\Omega (D_M k_{\xi} \tau_x) - 1}{v_c (1 + k_{\xi} v_M)} \right]. \quad (47)$$

Рівняння (47) запишемо в розширеній формі, враховуючи наступні параметри:

$$\Omega = u_0 / B_1 [\sin(\omega t_k)]^2; \quad B_1 = K_{\Pi} U_E \sqrt[3]{U_E} \left[ 1 - \exp\left(-2S_{\Pi} / \tau_x D_i\right) \right];$$

$$K_{\Pi} = k_e k_M k_c k_w B_0 (\alpha / \omega) \left[ 1 - (\omega / \alpha)^2 \right]; \quad B_0 = k_U m_c S_{\Pi}^2 (\alpha^2 / D_i) \sqrt[3]{(2k_U D_{\Pi} / D_i \rho_{\Pi})};$$

$$\alpha = -\tau_{21} / 2\tau_{22}^2; \quad \tau_x = \rho_{\Pi} v_{\Pi} / D_{\Pi} S_{\Pi}; \quad \tau_{21} = k_1 / D_{\Pi} \quad \text{і} \quad \tau_{22} = \sqrt{m / D_{\Pi}}.$$

У результаті отримаємо:

$$t = R_L \left\{ \frac{u_0}{U_E \sqrt[3]{U_E}} \left[ \frac{2k_{\xi} m_c (v_{\Pi} \rho_{\Pi})^2 D_M D_i^2 \sqrt[3]{D_i \rho_{\Pi}} \omega - 1}{2K_0 v_c (1 + k_{\xi} v_M) k_1 S_{\Pi}^5 D_{\Pi}^3 \sqrt[3]{D_{\Pi}} [\sin(\omega t_k)]^2} \right] \right\}, \quad (48)$$

де  $K_0 = k_e k_M k_c k_w k_U \sqrt[3]{2k_U}$  - еквівалентний коефіцієнт передачі УЗКР.

**Висновки.** Дослідження УЗКР рідинних середовищ показали, що вони можуть бути побудовані за наступними принципами: вимірюванням часу  $t$ , за який УЗС проходить відстань  $R_L$  у ГС; вимірюванням опорної напруги  $u_0$ , яка порівнюється на компараторі з ЕРС ультразвукового приймача, і за зміною напруги  $U_E$  ЕЗІ. У перших двох випадках статична характеристики УЗКР є практично лінійною в широкому діапазоні ВК рівня. У множнику (вказаний у квадратних дужках) рівняння (48) приведені конструктивні й технологічні параметри, від яких залежить точність контролю. Найбільший вплив на точність ВК чинять такі параметри, як: зміна щільності  $\rho_{\Pi}$  та жорсткості  $D_{\Pi}$  матеріалу ПКЕ, які є функціями температури. Окрім геометричних ці параметри є основними при взаємозамінності ультразвукового випромінювача та приймача. Достатньо сильний вплив на точність ВК чинить зміна швидкості розповсюдження в ГС УЗК, яка визначається за формулою  $v_c = \sqrt{k_a P / \rho_c}$ , де  $k_a$  - адіабатичний коефіцієнт для газів,  $P$

– тиск ГС,  $\rho_c$  – густина цього середовища, а згідно зі законом Менделєєва-Клапейрона  $\rho_c = M_c P / R_c T$ , де  $M_c$  – молекулярна вага;  $R_c$  – універсальна газова стала,  $T$  – температура. Тоді  $v_c = \sqrt{k_a R_c T / M_c}$ . Звідси випливає, що з підвищенням температури або зменшенні молекулярної ваги ГС вимірюваний час  $t$  зменшуватиметься, що аналогічно підвищенню рівня. Як видно з рівняння (48), вимірювальний час  $t$  суттєво залежить від густини  $\rho_{\Pi}$  матеріалу ПКЕ. Так як для твердого матеріалу  $\rho_{\Pi 2} = \rho_{\Pi 1} / [1 + 3\alpha_l (T_2 - T_1)]$ , де  $\rho_{\Pi 1}$ ,  $\rho_{\Pi 2}$  – густина при температурах  $T_1$  і  $T_2$  відповідно;  $\alpha_l$  – коефіцієнт лінійного розширення, то не трудно бачити, що вплив зміни температури на ГС та матеріал ПКЕ є однонаправленим, що збільшує температурну похибку вимірювального контролю УЗРК.

1. Баби́ков О. И. Контроль уровня с помощью ультразвука / О. И. Баби́ков. – Л.: Энергия, 1971. – 98 с.
2. Фрайден Дж. Современные датчики. Справочник / Дж. Фрайден; пер. с англ. Ю. А. Заболотной; под ред. Е. Л. Сви́нцева. – Москва: Техносфера, 2006. – 592 с.
3. Шарапов В. М. Пьезоэлектрические датчики / В. М. Шарапов, М. П. Муси́енко, Е. В. Шарапова; под ред. В. М. Шарапова. – Москва: Техносфера, 2006. – 632 с.
4. Жданкин В. К. Приборы для измерения уровня / В. К. Жданкин // Современные технологии автоматизации. – М.: СТА-ПРЕСС, 2002. – №2. – С. 6-19.
5. Стенцель Й. І. Математичні моделі ультразвукових рівнемірів рідин / А. В. Томсон, А. В. Рябіченко // МНТЖ «Вимірювальна та обчислювальна техніка в технологічних процесах». – Хмельницький, 2006. – №2 (28). – С. 55-58.
6. Стенцель Й. І. Аналіз похибок вимірювання ультразвукових рівнемірів / А. В. Томсон, А. В. Рябіченко // Вісник національного технічного університету «ХПІ». Збірник наукових праць. Тематичний випуск: Прилади та методи неруйнівного контролю. – Харків: НТУ «ХПІ» – № 48. – 2008. – С. 55-60.
7. Уровнемеры. Тематический каталог №5. Выпуск 6 / ПГ «Метран». – Челябинск, 2007. – 206 с.
8. Уровнемеры ультразвуковые МТМ900. Технические условия: ТУ 32.2-19081403-016-2004. – Северодонецк: ТОВ НВП «Мікротерм», 2004. – 35 с. – (Нормативний документ ТОВ НВП «Мікротерм»).
9. Стенцель Й. І. Вплив температури хімічних процесів на точність вимірювання рівня реакційних рідин ультразвуковим рівнеміром / А. В. Томсон, А. В. Рябіченко // Вісник СНУ ім. Володимира Даля. – 2007. – № 5(111). – Ч.2. – С.97-101.
10. Крым А. Е. Применение ультразвуковых уровнемеров в различных областях промышленности / А. Е. Крым // Сборник докладов семинара ЗАО «Взлёт». – Санкт-Петербург, 2007. – С. 97–100. Режим доступа: [www.vzljet.ru/files/publications2007/014.pdf](http://www.vzljet.ru/files/publications2007/014.pdf).
11. Томсон А. В. Ультразвуковой рівнемір зі змінним збуджуючим імпульсом / А. В. Томсон // Дис. на здобуття вчен. ступеня канд. техн. наук. – Вінниця: ВНУ 2009. – 215 с.
12. Рябіченко А. В. Ультразвуковой рівнемір з компенсуючим сприймаючим сигналом / А. В. Рябіченко // Дис. на здобуття вчен. ступеня канд. техн. наук. – Вінниця: ВНУ 2012. – 218 с.
13. Стенцель Й. И. Математические модели ультразвуковых датчиков уровня веществ / Й. И. Стенцель, А. В. Томсон. Вопросы химии и химической технологии. – Днепропетровск: УГХТУ, 2007. – Выпуск №5. – С. 182–185.
14. Стенцель Й. І. Математичні моделі ультразвукових рівнемірів рідин / Й. І. Стенцель, А. В. Томсон, А. В. Рябіченко // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2006. - №2. – С. 55–58.
15. Гораздовский Т. Я. Научные основы реологии. Монография / Т. Я. Гораздовский. – Луганск: Изд-во ВНУ им. В. Даля, 2009. – 699 с.
16. Стенцель Й. І. Фотоколориметричні газоаналізатори: Монографія / Й. І. Стенцель – К.: НМК ВО, 1992. – 120 с.
17. Кузьмичев В.Е. Законы и формулы физики / В. Е. Кузьмичев; отв. ред. В. К. Тартаковский. – К.: Наук. думка, 1989. – 864 с.
18. Кухлинг Х. Справочник по физике: Пер. с нем / Х. Кухлинг. – М.: Мир, 1982. – 520 с.

УДК 532.135.66.012

Проказа О.І.

Східноукраїнський національний університет ім. В. Даля

## МАТЕМАТИЧНІ МОДЕЛІ ТЕРМОДИНАМІЧНИХ ПРОЦЕСІВ В РЕАКЦІЙНІЙ СКЛЯНЦІ АПАРАТУ НЕЙТРАЛІЗАЦІЇ АМІАЧНОЇ СЕЛІТРИ

**Проказа О.І. Математичні моделі термодинамічних процесів в реакційній склянці апарату нейтралізації аміачної селітри.** Експериментальними дослідженнями встановлено, що температура у реакційній склянці апарату нейтралізації за її висотою змінюється нерівномірно. Для теоретичного дослідження термодинамічного режиму процесу нейтралізації використано принцип реологічних перетворень та методу нульового градієнта. Розроблені аналітичні моделі термодинамічних процесів. Показані причини зміни термодинамічного поля за висотою реакційної склянки.

**Ключові слова:** кислота, селітра, температура, швидкість, реакція, склянка, витрата, реологія, енергія.

**Проказа Е.И. Математические модели термодинамических процессов в реакционном стакане аппарата нейтрализации аммиачной селитры.** Экспериментальными исследованиями установлено, что температура в реакционном стакане аппарата нейтрализации за ее высотой изменяется неравномерно. Для теоретического исследования термодинамического режима процесса нейтрализации использован принцип реологических преобразований и метода нулевого градиента. Разработаны аналитические модели термодинамических процессов. Показаны причины изменения термодинамического поля по высоте реакционного стакана.

**Ключевые слова:** кислота, селитра, температура, скорость, реакция, стакан, расход, реология, энергия.

**Prokaza O.I. The mathematical models of thermodynamic processes in the reaction glass of the neutralization apparatus of ammonium nitrate.** Experimental studies established that the reaction temperature in a glass apparatus for neutralizing its height varies irregularly. For the theoretical study of thermodynamic regime neutralization process uses the principle of rheological changes and zero gradient method. Analytical models of thermodynamic processes are developed. It is shown the reasons thermodynamic field height glass reaction.

**Keywords:** acid, nitrate, temperature, speed, reaction, glass, consumption, rheology, energy.

**Вступ.** Виробництво аміачної селітри (АС) відрізняється значним енерго- та ресурсоспоживанням, що призводить до високої вартості готової продукції, викидів шкідливих речовин в атмосферу. На сучасних виробництвах АС конденсат сокової пари, як побічний продукт виробництва, використовується в невеликих кількостях для охолодження апарату нейтралізації (далі ВТН) тільки в тих випадках, коли температура реакційної маси досягає гранично допустимого значення. Перевищення температури в реакційній зоні склянки ВТН над температурою розчину АС на виході з нього на  $10^{\circ}\text{C}$  від номінальної приводить до різкого зростання викидів шкідливих речовин в атмосферу і створює пожежо- і вибухонебезпечну ситуацію на виробництві. На сучасних виробництвах АС процес нейтралізації азотної кислоти (АК) газоподібним аміаком (ГПА), як правило, ведуть за концентрацією йонів водню в реакційній зоні та в розчині АС на виході з ВТН за допомогою *pH*-метрів. За їх різницею змінюється витрата АК, яка подається в барботерну систему апарату ВТН. Такий спосіб керування технологічним процесом є недосконалим, так як, по-перше, температура реакційної маси на протязі деякого часу може підвищуватися за рахунок запізнення зміни витрати неконцентрованої азотної кислоти (НАК), по-друге, такий спосіб не дозволяє контролювати безпосередньо температуру, як основний впливовий фактор, і, по-третє, надійність і точність керування є достатньо низькими. Так як виробництва АС є пріоритетними не тільки з точки зору виготовлення мінерального добрива, але й як стратегічна сировина для багатьох галузей хімічних виробництв, тому задача підвищення його ефективності за рахунок зменшення енерго- і ресурсозбереження є актуальною задачею. Враховуючи той факт, що НАК й аміак виготовляються іншими виробництвами азотного комплексу, основна задача щодо підвищення ефективності виробництва АС може бути вирішена за рахунок вдосконалення існуючих і розробки нових систем автоматизованого контролю та керування.

**Аналіз останніх досліджень та літератури.** Сучасні агрегати виробництва АС мають низку специфічних особливостей, які потрібно враховувати при розробці систем автоматизації [1]. Велика потужність і послідовна структура схеми виробництва АС задають підвищені вимоги щодо надійності контролю, регулювання та захисту, оскільки вихід з ладу окремого елемента часто приводить до повної зупинки ВТН і, як наслідок, до великих економічних втрат. Організація управління об'єктами такого типу вимагає перегляду традиційних схем керування та нового -

системотехнічного підходу до розробки систем вимірювального контролю (ВК), автоматичного регулювання та автоматизованих систем управління.

Існуючі системи автоматизації виробництва АС побудовані за принципом стабілізації технологічних параметрів, що визначають режим процесу згідно з технологічним регламентом, але не дозволяють отримувати результати, які гарантують оптимальну його роботу. Для автоматизованої системи управління технологічним процесом (АСУТП) характерна наявність обчислювальної підсистеми, що забезпечує математичну й логічну обробку інформації за заданими алгоритмами, розробку та видачу керуючих дій на регулятори або безпосередньо на виконавчі механізми. На неї повністю або частково переносяться функції інформаційної підсистеми, а також функція контролю роботи протиаварійного захисту (ПАЗ) [2]. У роботі [3] розглянуті можливі напрямки маловитратної модернізації агрегату аміачної селітри. Відмінність техніко-економічних показників (ТЕП) виробництва АС залежно від особливості конкретних технологічних схем знаходиться, головним чином, в області витрати енергоресурсів, які залежать від концентрації використовуваної НАК та утилізації тепла нейтралізації. Аналіз багатотоннажних агрегатів отримання азотної кислоти АК-72М та аміачної селітри АС-72М, з точки зору їх керування проведений на Ферганському азотному комплексі «Азот» [4] дозволив виділити низку істотних особливостей, а саме: кожен з об'єктів є послідовною структурою виробництва кінцевого продукту, в якій технологічні лінії та вузли агрегатів знаходяться в тісному технологічному й інформаційному взаємозв'язку. У роботі [5] запропоновані нові хіміко-технологічні рішення виробництва аміачної селітри, що дозволяють отримувати добрива вищої якості і поліпшити екологічні умови виробництва.

Системи автоматичного регулювання (САР) призначені для автоматичної стабілізації окремих параметрів технологічного процесу, як: співвідношенням між витратами НАК і ГПА, температурою нейтралізації в склянці ВТН і концентрацією азотної кислоти в плаві АС. В основу САР покладено принцип одноконтурних систем регулювання. На діючих підприємствах виробництва аміачної селітри здійснюється за рахунок нейтралізації 56-62%-ої азотної кислоти. Використання більш концентрованої азотної кислоти, з однієї сторони, стримується проблемою її отримання, а з іншої - підвищенням температури у ВТН вище 180°C, що може привести до термічного розкладання селітри [6]. У результаті реакції нейтралізації в промислових агрегатах аміачної селітри виділяється 144936 Дж/моль (34624 кал/моль) тепла (з врахуванням технологічних втрат). Цього тепла достатньо для отримання розчину АС концентрацією вищою 86%.

Відомий спосіб автоматичного керування процесом нейтралізації НАК у виробництві аміачної селітри [7] шляхом регулювання її подачі в залежності від потенціалу платинового електроду відносно стандартного, причому платиновий електрод знаходиться в електролітичній комірці. Стандартний електрод розташований у посудині з еталонним розчином. Посудина з еталонним розчином розташована в тій самій електрохімічній комірці, яка знаходиться на виході апарату нейтралізації. Через вимірювальну комірку протікає розчин аміачної селітри. Різниця потенціалів вимірювального платинового і стандартного електроду змінює витрату НАК в апарат ВТН до тих пір, поки кислотність розчину АС не буде дорівнювати кислотності еталонного розчину. Недоліком цього способу є те, що вимірювання кислотності здійснюється тільки на виході розчину АС з реакційної склянки. Удосконалений спосіб автоматичного керування процесом нейтралізації кислоти, в якому керуюча дія на подачу НАК обумовлюється сигналом платинового електроду з реакційної зони апарату, за рахунок чого досягається збільшення точності управління процесом, виключення збоїв у роботі, забезпечення можливості автоматичного пуску виробництва, зниження втрат та викидів в оточуюче середовище кислоти і аміаку, описаний в [8]. Таким чином, з аналізу сучасних способів автоматичного керування процесом виробництва АС випливає, що відомі принципи керування не забезпечують відповідної точності, надійності та оптимальності.

**Метою статті** є дослідження термодинамічного поля склянки апарату ВТН для оцінки можливості використання більш ефективних і точних методів управління таким процесом.

**Постановка проблеми.** Як показано в [9, 10], існуючі способи, як правило, направлені на стабілізацію окремих технологічних параметрів, до яких відносять регулювання співвідношення витрат НАК і ГПА з метою забезпечення матеріального та стехіометричного балансу реакції за рахунок зміни витрати НАК і стабілізацію концентрації АК в розчині АС на виході з реакційної склянки апарату ВТН за рахунок зміни витрати рідкого аміаку. Причому концентрацію АК в



розчині АС вимірюють посереднім потенціометричним методом за концентрацією водню в цьому розчині без врахування того, що водневий показник, який визначається платиновими електродами не є достовірним, так як розчин у реакційній зоні ще знаходиться в зоні реологічного переходу, у котрому має місце сукупність непрореагованої АК, ГПА, аміачної селітри та інших складових, що достатньо сильно впливає на показання рН-метрів. У науково-технічній літературі практично не приводяться результати досліджень термодинамічного режиму роботи апарату нейтралізації. Вказується, що температура в зоні реакції не повинна перевищувати 165 °С, розчину АС на виході з реакційної склянки знаходиться в діапазоні від 150 до 160 °С. Не розглядаються питання залежності температури в реакційній зоні як за лінійною так і за часовою координатами. Апарат ВТН відноситься до багатопараметричних об'єктів управління, який формально має чотири вихідних регульованих координати, як: концентрація  $Q_{AC}$  аміачної селітри в розчині, температура  $T_P$  реакції, тиск  $P_C$  сокової пари та рівень і трьома регулюючими координатами, як: витрати НАК  $F_{AK}$ , ГПА  $F_A$ , конденсату сокової пари  $F_K$  та сокової пари  $F_C$ . Між технологічними параметрами мають місце перехресні зв'язки. У науковій літературі відсутні дослідження процесу нейтралізації азотної кислоти як багатопараметричного об'єкта керування.

Як показано у [11, 12] технологічні процеси в хімічній технології супроводжуються реологічними перетвореннями, в основі яких є перенесення кількості маси, енергії та руху. Наявність того чи іншого реологічного переходу призводить до зміни сталих часу перехідних процесів, а відповідно до появи як динамічних так і статичних похибок вимірювального контролю управління. Особливо це стосується оптимального управління процесу нейтралізації АК. Тому визначення точності вимірювального контролю при керуванні хіміко-технологічним процесом нейтралізації НАК ГПА є важливою науковою задачею.

**Матеріали досліджень.** Згідно з фізичною моделлю реологічного перетворення НАК та ГПА у плав аміачної селітри з врахуванням процесу реакції [13] перенесення маси цих матеріальних потоків через зону реологічного перетворення (ЗРП) описується наступним нелінійним диференціальним рівнянням

$$\frac{\partial T}{\partial \theta} + D_T \nabla^2 T + v_i \nabla T = F_{\Pi} \frac{d^3 T}{dx^3} + a \frac{d^2 T}{dx^2} + v_{AC} \frac{dT}{dx}, \quad (1)$$

де  $T$  - температура плаву АС, яка створюється в реакційній зоні склянки за час  $\theta$ ;  $D_T$  – коефіцієнт термодифузії;  $v_i$  – конвекційна складова швидкості теплового потоку;  $\nabla^2$  – лапласіан;  $\nabla$  – оператор Лапласа;  $F_{\Pi}$  – об'ємна витрата плаву АС;  $\alpha = \lambda / c_p \rho$  – температуропровідність плаву;  $\lambda, c_p, \rho$  теплопровідність, теплоємність і густина плаву відповідно;  $v_{AC}$  – швидкість переміщення плаву в реакційній склянці.

Для спрощення досліджень приймемо, що рух реакційної маси за висотою  $x$  реакційної склянки є одностороннім і рівномірним. Тоді рівняння (1) спрощується і приводиться до наступної форми

$$\frac{\partial T(y, \theta)}{\partial \theta} + D_T \frac{\partial T(y, \theta)}{\partial y^2} + v_i \frac{\partial T(y, \theta)}{\partial y} = F_{\Pi} \frac{d^3 T(x)}{dx^3} + a \frac{d^2 T(x)}{dx^2} + v_{AC} \frac{dT(x)}{dx}, \quad (2)$$

де  $y$  – лінійна координата зони джерела теплової енергії (реакційна зона).

Так як фізико-хімічний процес, який описується рівнянням (2), протікає в ЗРП (у реакційній склянці апарату ВТН), то його можна описати такою інтегральною імпульсною дельта-функцією Дірака

$$\int_{\theta=0}^t f(\xi) \delta(\xi - \theta_0) d\xi = \begin{cases} 0 & t_0 < t_1, t_0 \geq t_2; \\ f(t+0) & t_1 < t_0 < t_2, \end{cases}, \quad (3)$$

де  $f(\xi)$  – функція перетворення;  $\xi$  – деяка змінна, залежна від часу  $\theta$ ;  $\delta(\xi - \theta_0)$  – ядро інтегральної імпульсної дельта-функції Дірака;  $f(t_0 + 0)$  – функція стоку аміачної селітри;  $t_0$  – номінальне значення часу стоку АС;  $t_1, t_2$  – допустимі межі зміни часу стоку АС.

У нашому випадку функцією перетворення буде залежність  $T(x) = fT(y, \theta)$ , а ядром функції (3) буде варіація цієї функції, а фактично її інтегральний функціонал. Згідно з методом

нульового градієнта [14] нелінійне диференціальне рівняння (2) розкладається на наступну систему диференціальних рівнянь:

$$\frac{\partial T(y, \theta)}{\partial \theta} + D_T \frac{\partial^2 T(y, \theta)}{\partial y^2} + v_i \frac{\partial T(y, \theta)}{\partial y} = 0; \quad (4)$$

$$\frac{F_{\Pi}}{v_{AC}} \frac{d^2 T(x)}{dx^2} + \frac{a}{v_{AC}} \frac{dT(x)}{dx} + T(x) = T(y, \theta). \quad (5)$$

Для аналітичного розв'язку рівняння (4) потрібно обґрунтувати початкові та кінцеві умови. Якщо припустити, що швидкість конвекційної складової незначна і нею можна знехтувати, то рівняння (4) приводиться до вигляду:

$$\frac{\partial T(y, \theta)}{\partial \theta} + D_T \frac{\partial^2 T(y, \theta)}{\partial y^2} = 0, \quad (6)$$

рішенням якого при нульових початкових умовах буде:

$$T(y, \theta) = T_0 \operatorname{erfc}(y/2\sqrt{D_T \theta}). \quad (7)$$

Якщо швидкістю конвекційної складової знехтувати не можна, а  $y = v_i \theta$  є лінійною функцією, то рівняння (4) приймає вигляд

$$\frac{D_T}{2v_i^2} \frac{\partial T(y, \theta)}{\partial \theta} + T(y, \theta) = T_0, \quad (8)$$

де  $T_0$  – номінальна температура в ЗРП.

Розв'язком рівняння (8) при нульових початкових умовах буде:

$$T(y, \theta) = T_0 [1 - \exp(-2\theta v_i^2 / D_T)]. \quad (9)$$

Рівняння (5) описує розподілення термодинамічного поля вздовж висоти  $x$  реакційної склянки, а фактично від місця уведення реагуючих компонентів до стоку реакційної маси з апарату ВТН. Приймаючи до уваги нульові початкові умови, рішенням цього рівняння буде:

$$T(x) = T(\theta, y) \{1 - \exp(-\beta x) [\cos(\psi x) + (\beta/\psi) \sin(\psi x)]\}, \quad (10)$$

де  $\beta = -a/2F_{\Pi}$ ;  $\psi = \sqrt{v_{AC}/2F_{\Pi}} \sqrt{1 - (a\beta/v_{AC})}$ .

Підставивши рівняння (7) або (9) у (10), отримуємо математичні моделі процесу перенесення теплової енергії в ЗРП склянки апарату ВТН в аналітичній формі. Для процесу, коли можна знехтувати швидкістю конвекційного потоку:

$$T(x, v_i = 0) = T_0 \operatorname{erfc}(y/2\sqrt{D_T \theta}) \{1 - \exp(-\beta x) [\cos(\psi x) + (\beta/\psi) \sin(\psi x)]\}. \quad (11)$$

Якщо швидкість розповсюдження тепла в ЗРП є лінійною, то маємо:

$$T(x) = T_0 [1 - \exp(-2\theta v_i^2 / D_T)] \{1 - \exp(-\beta x) [\cos(\psi x) + (\beta/\psi) \sin(\psi x)]\}. \quad (12)$$

Експериментальними дослідженнями встановлено, що розподілення температури реакційної маси в ЗРП апарату ВТН від системи уведення реагентів (барботерної системи) до виходу плаву зі склянки є нерівномірним як показано на рис. 1.

Згідно з технологічним регламентом допустимий діапазон зміни температури плаву АС може змінюватися від  $T_1=150^{\circ}\text{C}$  до  $T_3=165^{\circ}\text{C}$ . При температурі  $T < T_1$  появляється ймовірність кристалізації плаву та зменшення його концентрації, а при  $T > T_3$  плав може бути вибухонебезпечним. Як правило, стабілізацію температури плаву АС виконують шляхом уведення в склянку охолодженого конденсату сокової пари (КСП).

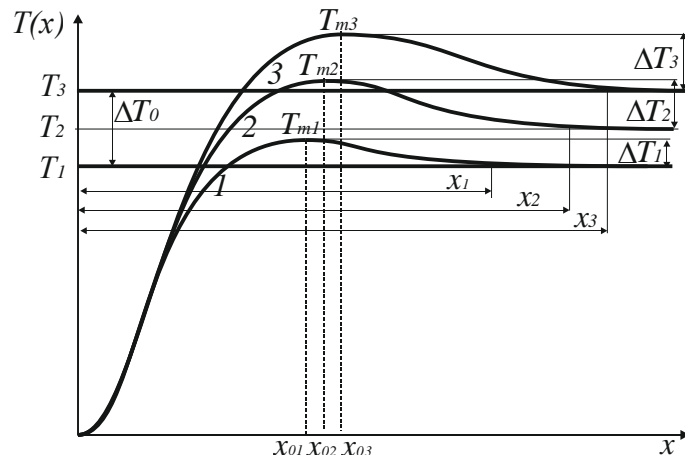


Рис. 1. Експериментальні криві розподілення температури плаву в ЗРП за висотою склянки

З рисунка видно, що у допустимому діапазоні  $\Delta T_0$  максимум температури в склянці знаходиться на достатньо вузькій лінійній ділянці – від  $x_{01}$  до  $x_{03}$ . У той же час різниця температур  $\Delta T_1, \Delta T_2, \Delta T_3$  збільшується зі збільшенням температури  $T_1, T_2, T_3$  відповідно.

**Результати досліджень.** Аналіз теоретичних та експериментальних досліджень показує, що зміна усталеного значення температури плаву АС  $\Delta T_i$  визначається сумарною швидкістю матеріальних потоків, котрі уводяться в реакційну склянку. Якщо в реакційну зону склянки вводиться КСП, котрий рівномірно розподіляється за рахунок зустрічного руху потоків, то рівняння (11) і (12) приймають наступну форму:

$$T(x, v_i = 0) = T_0 \operatorname{erfc} \left( v_k \sqrt{\frac{\theta}{4D_T}} \right) \{1 - \exp(-\beta x) [\cos(\psi x) + (\beta/\psi) \sin(\psi x)]\}; \quad (13)$$

$$T(x) = T_0 \{1 - \exp(-2\theta(v_i + v_k)^2/D_T)\} \{1 - \exp(-\beta x) [\cos(\psi x) + (\beta/\psi) \sin(\psi x)]\}, \quad (14)$$

де  $v_k$  – лінійна швидкість уведення КСП в реакційну зону склянки.

Позначивши у рівнянні (5)  $\zeta = F_{\Pi} / v_{AC}$  і  $\sigma = a / v_{AC}$ , знаходимо наступне відношення:  $\phi = a / \sqrt{(v_{AC}) F_{\Pi}}$ . Якщо  $\phi \geq 2$ , то функція  $T(x) = f(x)$  буде монотонно зростаючою до деякого максимального значення температури, котре дорівнюватиме максимальній температурі в реакційній зоні (температури в точках  $x_{01}, x_{02}, x_{03}$ ). Якщо в реакційну зону вводиться КСП з витратою  $F_k = \text{const}$  і температурою  $T_k$ , то параметр  $\phi$  зменшиться на величину  $k = T_m / T_k$ , де  $T_m$  – максимальна температура плаву АС в склянці;  $T_k$  – температура КСП, цей параметр стає меншим 2 і розподілення за висотою склянки стає таким, як показано на рис. 1. Експериментальні дослідження показують, що значення максимальної температури  $T_m$ , в основному, залежить від двох факторів: масової витрати НАК і ГПА, які подаються в барботерну систему, і зміни концентрації азотної кислоти, яка може змінюватися від 56 до 65%. Практично КСП є тим динамічним елементом системи управління процесом нейтралізації АК, який забезпечує необхідний термодинамічний режим роботи апарату ВТН. Як вказувалося вище, забезпечення оптимального режиму роботи агрегату виконують за рахунок вимірювального контролю кислотності в зоні максимальної температури  $T_m$  і в плаві АС на виході з апарату ВТН. Як показали експериментальні дослідження роботи апарату ВТН, стабілізація режиму роботи апарату ВТН на рівні мінімальної робочої температури  $T_1$  (див. рис. 1) приводить до зменшення активної робочої висоти реакційної склянки до  $x_1$ , а значить до зменшення продуктивності роботи апарату ВТН, а відповідно, - до виробництва АС в цілому. При роботі апарату ВТН на максимальній температурі  $T_3$  продуктивність апарату є високою, але виникає висока ймовірність появи небезпечних ситуацій з однієї сторони та викидів пари азотної кислоти в сокову пару – з іншої.

Тому найбільш оптимальним є режим, для якого зміна термодинамічного поля показана на рис. 1, крива 2. Окрім того, експериментальні дослідження показали, що при уведенні в апарат ВТН КСП перепад температур  $\Delta T_j = T_{mj} - T_j$ , де  $j=1, 2, 3$ , залежить від значення максимальних температур  $T_{mj}$ .

**Висновки.** Приведені вище теоретичні та експериментальні дослідження реологічних перетворень, які мають місце в ЗРП реакційної склянки апарату ВТН показують, що на технологічний процес нейтралізації азотної кислоти суттєвий вплив чинить термодинамічний режим, який є наслідком екзотермічної реакції неконцентрованої азотної кислоти та газоподібного аміаку. Показано, що розподілення температурного поля за висотою реакційної склянки апарату ВТН є нелінійним. Максимум температури наступає в зоні реологічного переходу та обумовлений уведенням конденсату сокової пари. За рахунок цього температура плаву на виході з апарату ВТН зменшується на величину  $\Delta T$ . Так як різниця температур  $\Delta T$  є практично лінійною функцією від витрати  $F_k$  конденсату сокової пари, то її можна використати для управління температурним режимом за допомогою стандартних термопар замість потенціометричного методу на основі дорогих платинових електродів, які використовуються на виробництвах АС. Так як на зміну температурного режиму в апараті ВТН впливає як витрата КСП, так і його температура, то для підвищення точності стабілізації температури плаву на виході з апарату рекомендується стабілізувати останню за каскадним принципом.

1. Иванов М. Е. Производство аммиачной селитры в агрегатах большой единичной мощности / М. Е. Иванов, В. М. Олевский, Н. Н. Поляков и др. – М.: Химия, 1990. – 228 с.
2. Азизов А. М. Информационные системы контроля параметров технологических процессов / А. М. Азизов. – Л.: Химия, 1983. – 328 с.
3. Митронов А. П. Направления модернизации производственного комплекса азотная кислота – аммиачная селитра / А. П. Митронов, А. П. Мудрый, А. Н. Кочергин // Хімічна промисловість України. – 2000. – №1-2. – С. 30-34.
4. Адылов Ф. Т. Система управления технологическим процессом производства слабой азотной кислоты и аммиачной селитры в агрегатах АК-72М и АС-72М на ферганском ПО «Азот» / Ф. Т. Адылов, Н. Н. Турапина, В. В. Перекрестов // Промышленные контроллеры АСУ. – 2004. – №4. – С. 1-6.
5. Щегров Л. Н. Азотные удобрения. Пути производства аммиачной селитры улучшенного качества / Л. Н. Щегров, В. В. Манк, И. Н. Новиков // Хімічна промисловість України. – 2004. – №6(65). – С. 6.
6. Правила безопасности для неорганических производств азотной промышленности (ПБНП-74). – М.: Недра, 1978. – 37 с.
7. Миниович М. А. Производство аммиачной селитры / М. А. Миниович. – М.: Химия, 1974. – 239 с.
8. Пат. 43404 Україна, С01N27/48. Спосіб автоматичного управління процесом нейтралізації кислоти у виробництві аміачної селітри в апараті використання тепла нейтралізації / Анохін А. О., Белік С. В., Довгалов Л. Ю., Циглевський П. М., Чистоклетов М. В.; заявник та патентовласник Державний науково-дослідний і проектний інститут хімічних технологій – №97115422; заявл. 12.11.1997; опубл. 17.12.2001, Бюл.№11.
9. Stentsel I. Researches of the system of neutralization process control in the production of ammonium nitrate on the basis of rheological transitions principles / I. Stentsel, O. Prokaza, E. Prokaza // An International journal on motorization, vechicle, operation, energy efficiency and mechanical engineering "ТЕКА Commission of motorization and Energetics in Agriculture". – 2012. – Vol.12, № 4. – P. 274-278.
10. Проказа О. І. Теоретичні основи управління апаратом нейтралізації азотної кислоти за температурною депресією у виробництві аміачної селітри / О. І.Проказа // Науковий журнал «Вісник Східноукраїнського національного університету ім. Володимира Даля». – 2013. – №15(194). – Ч. 1 – С. 222-228.
11. Стенцель Й. І. Математичні моделі управління апаратом нейтралізації кислоти за температурою реакції у виробництві аміачної селітри / Й. І. Стенцель., О. І. Проказа., О. В. Поркуян // Науковий журнал «Вісник Східноукраїнського національного університету ім. Володимира Даля». – 2012. – №15(186). – Ч. 2. – С. 114-122.
12. Стенцель Й. І. Дослідження оптимальності процесу нейтралізації азотної кислоти аміаком за температурою в реакційній зоні / Й. І. Стенцель, О. І. Проказа, О. В. Поркуян // Матеріали 8-ї Міжнародної науково-практичної конференції "Розвиток наукових досліджень - 2012". – Полтава, 19-21 листопада 2012 р. – Т. 9. – С. 73-75.
13. Проказа О. І. Автоматизоване керування процесом нейтралізації у виробництві аміачної селітри за принципами реологічних переходів / Дис. на здобуття канд. техн. наук. – Луганськ, 2014. – 171 с.
14. Стенцель Й. І. Фотоколориметричні газоаналізатори: Монографія / Й. І. Стенцель – К.: НМК ВО, 1992. – 120 с.

УДК 681.5:656.8

Шолом П.С., Жигаревич О.К., Рудник О.Ю.  
Луцький національний технічний університет

## АВТОМАТИЗОВАНА ІНФОРМАЦІЙНА СИСТЕМА УПРАВЛІННЯ ВІДДІЛЕННЯМИ ПОШТОВОГО ЗВ'ЯЗКУ

**Шолом П.С., Жигаревич О.К., Рудник О.Ю.** Автоматизована інформаційна система управління відділеннями поштового зв'язку. Здійснено аналіз потреб українського державного підприємства поштового зв'язку «Укрпошта», розробку та впровадження автоматизованої інформаційної системи управління в інформаційний простір на рівні дирекції підприємства.

**Ключові слова:** автоматизована інформаційна система, відділення поштового зв'язку, автоматизоване робоче місце, база даних, інформаційні технології.

**Шолом П.С., Жигаревич О.К., Рудник А.Ю.** Автоматизированная информационная система управления отделениями почтовой связи. Осуществлен анализ потребностей украинского государственного предприятия почтовой связи «Укрпочта», разработку и внедрение автоматизированной информационной системы управления в информационное пространство на уровне дирекции.

**Ключевые слова:** автоматизированная информационная система, отделение почтовой связи, автоматизированное рабочее место, база данных, информационные технологии.

**Sholom P., Zhyharevych O., Rudnyk O.** Post Offices Automated Information Management System. The analysis of the needs of Ukrainian State Enterprise of Posts «Ukrposhta», development and implementation of an automated information management system in the information space at the directorate level are realized.

**Keywords:** automated information system, post office, workstation, database, information technologies.

**Постановка проблеми.** У сучасному суспільстві інформація стає найбільш важливою цінністю, а індустрія отримання, обробки і трансляції інформації – провідною галуззю діяльності, куди з кожним роком вкладають все більш значні капітали.

В останні десятиліття інформацію розглядають як один з основних ресурсів розвитку суспільства, а інформаційні системи та технології – як засіб підвищення продуктивності праці та ефективності роботи персоналу. Переробка інформації – найважливіша функція, без якої немислима цілеспрямована діяльність будь-якої соціально-економічної, організаційно-виробничої системи (підприємства, організації, території і т.п.). Найбільш широко інформаційні системи та технології використовуються у виробничій, управлінській і фінансовій діяльності. Проте очевидно, що впровадження та активне застосування таких систем і технологій необхідне також для людей, зайнятих в інших сферах [1].

Тому перед кожним підприємством, установою, організацією постає питання ефективного впровадження інформаційних технологій задля забезпечення своєчасного вироблення і прийняття управлінських рішень, вивільнення частини робочого часу працівників, поліпшення якості і оперативності управління, його дієвості тощо.

**Аналіз останніх досліджень і публікацій.** В низці публікацій зосереджується увага на необхідності застосування автоматизованих систем та інформаційних технологій у фінансовому контролі при проведенні ревізій та перевірок, визначаються вимоги до комп'ютерної техніки [2; 3; 4; 5]. Окремі аспекти автоматизації контролю розглядають В.П. Завгородній [5], Є.В. Калюга [2], Р.І. Криницький [3], В.В. Павлюк [4], В.В. Сопко [5] та інші автори. Проблеми автоматизації контролю, принципи побудови та функціонування автоматизованої системи контролю, її інформаційне забезпечення розглядаються в праці М.Т. Білухи [6].

На даний час практично всі підприємства України у своїй діяльності вже застосовують автоматизовані інформаційні системи. Проведений аналіз можливостей таких систем на наявна достатньо глибока теоретична проробка питань створення автоматизованих інформаційних систем [7; 8; 9; 10], свідчать про необхідність і можливість подальшого удосконалення їх структури, змісту, принципів функціонування та організації їх застосування.

**Виділення невирішених раніше частин загальної проблеми.** У кожному підприємстві виникає проблема, пов'язана з отриманням, зберіганням та обробкою інформації. Для вирішення даної проблеми створюються автоматизовані системи для управління та обробки інформації. Комплексна автоматизація інформаційних потоків підприємства, організації, відомства, галузі вимагає створення єдиного інформаційного простору для забезпечення можливості віддаленої

роботи працівників із базами даних, вільного доступу до засобів телекомунікації, збереження цілісності даних у загальній БД, повнотекстового і реквізитного пошуку інформації, належного захисту інформації, налаштування інтерфейсу на задачі користувачів тощо. Із подібною проблемою стикнулася Волинська дирекція УДППЗ «Укрпошта».

**Метою дослідження** є пошук шляхів і засобів підвищення ефективності роботи підприємства та якості надання його послуг через впровадження інформаційних технологій в апаратах управління підприємством та розробка і впровадження в інформаційний простір українського державного підприємства поштового зв'язку «Укрпошта» автоматизованої інформаційної системи управління на рівні дирекції підприємства.

#### **Виклад основного матеріалу дослідження.**

Інформаційна технологія – це методи обробки інформації та організаційно-управлінські концепції її формування і споживання, а також сукупність усіх видів інформаційної техніки; єдність процедур щодо збирання, накопичення, зберігання, обробки та передачі даних із застосуванням вибраного комплексу технічних засобів.

Автоматизована інформаційна система (АІС) – це система, що реалізує інформаційну технологію у сфері управління шляхом спільної роботи управлінського персоналу і комплексу технічних засобів.

Обробку інформації можна здійснити, використовуючи комплекси засобів обчислювальної та інформаційної техніки, за допомогою якої раціонально виконуються чотири групи основних операцій: збір і передача інформації в ПЕОМ, її обробка за заздалегідь розробленими алгоритмами та програмами, видача обробленої інформації користувачам, зберігання й пошук інформації. Залежно від експлуатаційних можливостей і ступеня застосування зазначеної техніки при виконанні цих операцій відбувається часткова чи комплексна механізація операцій, часткова чи повна їх автоматизація [11].

У сучасних умовах, коли потрібно не лише збирати й обробляти інформацію, а й мати змогу забезпечувати нею менеджерів та фахівців різних рівнів, зазначена система дістала подальший розвиток і діє як система автоматизованого збирання та обробки інформації. Така система може створюватися на різних рівнях, у масштабі різних організаційно-управлінських ланок -- цеху чи виробничої одиниці, підприємства чи виробничого об'єднання, галузі, регіону тощо. Зі збільшенням масштабності управлінської ланки значно зростають трудомісткість і складність організації системної обробки інформації.

При системі автоматизованого збирання й обробки інформації, яка діє в ритмі виробництва на об'єкті управління, не лише збирають і обробляють всі первинні дані, що характеризують виробничо-господарську та іншу діяльність, а й забезпечують (через систему інформування) обробленою інформацією всіх менеджерів та фахівців різних рівнів. Завдяки безпаперовій технології користувачі оперативно одержують на екрані ПЕОМ об'єктивно необхідну інформацію згідно з делегованими повноваженнями, якими вони наділені менеджерами вищого рівня, функціональними обов'язками, посадовими правами та мірою відповідальності. Вихідна інформація, як правило, є повністю оброблена, включає результати елементарного аналізу даних і є необхідною для вироблення та прийняття управлінських рішень [12].

Експлуатаційні можливості сучасного комплексу технічних засобів, що використовується в системі автоматизованого збирання й обробки інформації, дають змогу автоматизовано виконувати цілу низку функцій. Стан науково-практичних розробок та технічний рівень згаданого комплексу визначили можливості автоматизованого виконання таких процедур управлінського процесу:

1. У прогнозуванні та плануванні – багатоваріантні розрахунки під час розробки прогнозів, перспективних і поточних економічних та соціальних планів розвитку підприємства, а також оперативно-виробничих планів і планів з технічної підготовки виробництва з метою подальшого визначення оптимальних взаємозв'язаних наборів показників планування в почасовому (година, зміна, тиждень тощо) і в пооб'єктному (робоче місце, ділянка тощо) аспектах.

2. В організації – моделювання організаційних структур управління та імітація процесів виробництва при різних критеріях і параметрах з метою вибору оптимальних.

3. При координації і регулюванні – подання команд на робочі місця (поки що на низовому рівні управління виробництвом) відповідно до плану, технологічного процесу чи інструкції, складених на ті чи інші види робіт або операції.

4. У контролі – спостереження за станом керованого об'єкта за всіма параметрами, а також за своєчасним і повним виконанням керівних команд.

5. В обліку – одноразове збирання (в ритмі виробництва) й системна обробка всієї фактичної (разом з довідковою, плановою, нормативною та іншою) вірогідної інформації про наявність та рух ресурсів, а також про стан, процеси та явища, що мають місце у виробничо-господарській та іншій діяльності підприємства.

6. В аналізі – зіставлення нормативних, планових і фактичних показників, що характеризують ті чи ті операції або процеси виробничо-господарської та іншої діяльності, виявлення відхилень (у кількісних, вартісних, відносних та інших величинах) від заданих параметрів із зазначенням причин і винуватців цих відхилень, оцінка виконання плану в різних аспектах та виявлення факторів, що впливають на ці відхилення.

7. У звітності – автоматичне формування (на основі первинних даних) зведених показників для типових форм установленної бухгалтерської, статистичної та іншої звітностей за допомогою спеціальних перевідних масивів – довідників, – а також одночасне створення машинних носіїв зі зведеними показниками звітності для передавання каналами зв'язку їх до зовнішніх установ (інституцій) вищого рівня [12].

Автоматизована інформаційна система управління відділеннями поштового зв'язку (АІСУ ВПЗ) розроблена з метою контролю роботи працівників відділень поштового зв'язку Волинської дирекції УДППЗ «Укрпошта», а також для контролю знань робочого персоналу підприємства. Автоматизована система є програмним продуктом, розробленим на базі мови програмування С#. Одним з основних завдань АІСУ є управління інформацією баз даних автоматизованого робочого місця відділення зв'язку (АРМ ВЗ).

Основою АІС є платформа, на яку її орієнтовано, тобто види ПК та програмних середовищ, закладених в ІТ-реалізацію процесу управління.

Для розробки системи використано такі програмні продукти:

Microsoft Visual Studio – лінія продуктів компанії Microsoft, які включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів, що дозволяють розробляти консольні додатки, додатки з графічним інтерфейсом, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються ОС Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework і Silverlight. Середовище Visual Studio 2013 має повну підтримку програмних продуктів на ОС Windows, підтримку роботи Firebird SQL, зручний редактор коду та підтримку мови програмування високого рівня С#.

Microsoft .NET – програмна технологія, запропонована фірмою Microsoft як платформа для створення як звичайних програм, так і веб-додатків. Для розробки автоматизованої інформаційної системи управління відділеннями поштового зв'язку використано платформу .Net Framework 4.0. Дана платформа підтримує всі версії ОС Windows (починаючи з Windows XP) і своєю функціональністю повністю задовольняє вимогам до розробки системи.

Firebird (іноді FirebirdSQL) – компактна, крос-платформова, вільна реляційна система керування базами даних, що реалізує більшість функцій ANSI SQL 2003. Вона може запускатись на більшості Unix-систем (в тому числі Linux та FreeBSD) та Windows.

Crystal Reports – бізнес-аналітик додатків по SAP SE. Призначений для розробки і створення звітів з широкого спектру джерел даних. Використовується як додаткове програмне забезпечення для Visual Studio 2013.

IBExpert – GUI-оболонка, призначена для розробки та адміністрування баз даних InterBase та Firebird, а також для вибору і зміни даних, що зберігаються в базах. IBExpert володіє безліччю компонентів, які полегшують роботу: візуальний редактор для всіх об'єктів бази даних, редактор SQL і виконавець скриптів, відладчик для збережених процедур і тригерів, конструктор області, інструмент для імпорту даних з різних джерел, власна скриптова мова, а також дизайнер баз даних і т. д.

У структурі системи (рис. 1) є модулі та бази даних. АІСУ ВПЗ має свою власну базу даних. В ній розміщуються таблиці, які є основою системи. Така структура програмного забезпечення дозволяє будь-якому користувачеві працювати з однією базою даних, що знаходиться на сервері. АІСУ ВПЗ має можливість роботи з базами даних АРМ ВЗ, що знаходяться на кожному автоматизованому відділенні поштового зв'язку.

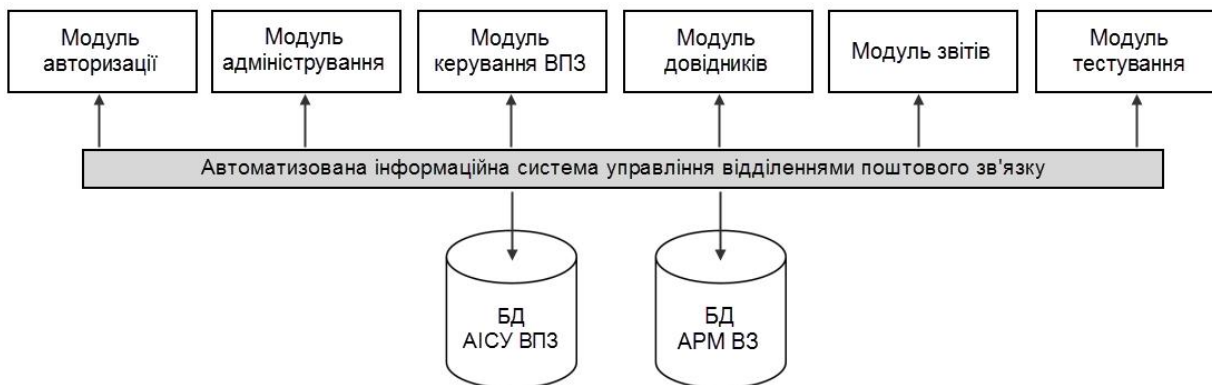


Рис. 1. Структура АІСУ ВПЗ

**Модуль авторизації.** Оскільки автоматизована система має доступ до керування інформацією баз даних АРМ ВЗ (у цій базі зберігаються дані про всі надані послуги відділеннями поштового зв'язку, всі готівкові та безготівкові операції), було б недоцільно давати всім доступ до операцій, які можуть змінювати інформацію у цій базі. Тому в програмі передбачено три ролі для користувачів: адміністратор, користувач та тестування.

При вході в автоматизовану систему користувач повинен ввести свій логін та пароль. Після цього вступає в роботу функція `Authorization()`, яка перевіряє дані, введені користувачем. Якщо дані логіну або паролю не співпадають із заведеними в БД АІСУ ВПЗ адміністратором користувачами, то функція виводить повідомлення про помилку авторизації та блокує вхід до системи. Якщо ж користувач ввів все вірно, тоді відкривається головне меню системи, яке відрізняється в залежності від ролі користувача.

Також в модулі авторизації передбачено збереження логіну та паролю користувачів з метою уникнення повторного їх вводу. Для збереження логіну та паролю користувача використовується `application.ini`. Сама ідея простого запису паролів та логінів користувача в файл конфігурації не є цілком безпечною в плані захисту інформації, тому під час збереження даних користувача в дію вступає функція `SaveUser()`. Вона викликає клас системи `RC4` та кодує отримані дані за допомогою алгоритму кодування `RC4` і тільки після такого процесу записує дані у файл налаштування. Як тільки користувач захоче повторно увійти в програму, виконується функція `LoadUser()`, яка отримує дані про логін та пароль користувача з файлу налаштування та декодує його. Таким методом зменшується вірогідність крадіжки авторизаційних даних користувача.

**Модуль адміністрування.** Цей модуль доступний тільки користувачам з роллю «Адміністратор». В модулі на даний час передбачено лише ведення користувачів автоматизованої системи та призначення їм ролей «Адміністратор» (має повний доступ до усіх модулів автоматизованої системи), «Користувач» (має доступ до модулів: довідників, звітів, тестування) та «Тестування» (має доступ тільки до окремих пунктів модуля тестування).

Таким чином в автоматизованій системі розподілені права доступу з метою зменшення вірогідності стороннього втручання в роботу системи з боку працівників.

Також слід зауважити, що у модулі адміністрування є один системний користувач – адміністратор. Змінити ім'я цього запису чи здійснити його видалення не можливо (є доступ тільки на заміну паролю).

**Модуль керування ВПЗ.** Оскільки відділення поштового зв'язку роблять помилки при роботі в АРМ ВЗ, дуже часто залишок готівки у касі оператора не сходиться з залишком в автоматизованому робочому місці. У цьому випадку начальники таких ВЗ пишуть службову записку з поясненням причин розбіжності в коштах, на підставі якої інженер-програміст повинен виправити залишок готівки на початок дня в АРМ ВЗ. До створення автоматизованої системи для виправлення залишку необхідно було віддалено підключатися на відділення, заходити в його базу даних, шукати таблицю із записами про дані залишки та рядок в таблиці, який необхідно замінити. Це вимагало багато часу і міг це зробити тільки інженер-програміст по супроводу АРМ ВЗ. Тому в модулі керування ВПЗ було створено пункт «Зміна залишку на початок дня». За допомогою



даного пункту користувач з роллю «Адміністратор» має можливість здійснити процедуру зміни залишку за кілька секунд без участі ВЗ та інженерів-програмістів по супроводу АРМ ВЗ.

Для цього користувачу необхідно зайти у відповідний пункт модуля та обрати назву відділення поштового зв'язку, місяць, рік та дату, на яку необхідно замінити залишок. Після виконання цієї процедури система автоматично підключається до БД відділення та виводить саме ті рядки, у яких необхідно замінити суму. Для цього використовується функція `Poshuk()`. Після коригування залишку користувач підтверджує зміни і в дію вступає функція `UpdateSumVpz()`. Особливість даної функції полягає у тому, що в якості розділювача цілої та дробової частини числа обрано крапку. Це вирішує проблему налаштувань розділювача, який на машинах користувача та відділення поштового зв'язку може бути різним, що може призвести до його зникнення (наприклад, замість 100,51 грн. можна отримати 10051 грн.).

**Модуль довідників.** Даний модуль доступний користувачам з ролями «Адміністратор» та «Користувач». Він включає в себе зовнішні довідники, до яких звертається автоматизована система для вирішення певних поставлених перед нею задач. Наприклад, для підключення системи до бази даних АРМ ВЗ їй необхідно знати адресу сервера та шлях, де знаходиться ця база даних. Ці адреси та шляхи можна було б ввести в код самої АІСУ ВПЗ. Але при зміні адреси сервера чи шляху до бази даних, програміст був би зобов'язаний дописати чи змінити програмний код автоматизованої системи та оновлювати програму на усіх робочих місцях. Для вирішення подібних проблем і був створений модуль довідників. Довідники є незалежними від програмного коду автоматизованої системи. Система викликає їх тільки тоді, коли необхідно отримати дані.

В модулі є можливість редагування даних, швидкого пошуку записів, а також імпорт та експорт даних. За допомогою модуля процес оновлення даних, які необхідні для коректної роботи автоматизованої системи, можна виконувати безпосередньо на базі даних АІСУ ВПЗ (без участі робочих місць користувачів), а користувачі автоматично матимуть змогу опрацювати оновлені дані з довідників.

Для зручності експорт та імпорт довідників реалізовано у двох форматах: CSV та XML. При імпорті АІСУ ВПЗ виключає можливість імпорту невірних даних за допомогою функції `ImportXml()` або відповідно `ImportCsv()`. В даних функціях звіряється структура даних файлу зі структурою даних довідника.

**Модуль звітів.** За зверненням відділу технології послуг поштового зв'язку Волинської дирекції УДППЗ «Укрпошта» було розроблено модуль звітів, за допомогою якого проводиться контроль за роботою працівників на відділеннях поштового зв'язку. На даний момент модуль включає в себе три види звітів: реєстр операційних послуг, поштові відправлення, термін зберігання яких закінчився та пошук прийнятих поштових відправлень.

АІСУ ВПЗ дає можливість формування звітів як за обраний період, так і за конкретний день. Таким чином відділ технології послуг бачить якість виконаної роботи операторами відділення. Слід зазначити, що в якому б районі не знаходилась база даних АРМ ВЗ, формування звітів виконується однаково швидко.

Самі звіти формуються на базі даних, отриманої із АРМ ВЗ. В залежності від звіту автоматизована система обирає необхідні дані, групує їх та виводить для перегляду користувачеві. Для виведення звіту використовується модуль `SAP CrystalReport`, який є дуже зручним. За його допомогою можна виконувати пошук по звіті на етапі його попереднього перегляду перед друком та присутня можливість збереження звіту у різних форматах: PDF, xls, doc та інших.

**Модуль тестування.** На прохання відділу технології послуг поштового зв'язку та відділу кадрів підприємства було створено модуль для тестування знань працівників. Повний доступ до нього мають користувачі з роллю «Адміністратор» та «Користувач», а користувач з роллю «Тестування» має доступ тільки для проходження тестування.

Модуль складається з трьох частин:

1. Налаштування. Тут відповідальна за тестування особа заносить питання та варіанти відповіді на них, вказує кількість запитань для проходження тесту та вказує право на перегляд результатів по завершенню тестування.

2. Тестування. Ця частина виконує сам процес тестування працівника. АІСУ ВПЗ автоматично генерує випадкові запитання із власної бази даних. За це відповідає функція `GetRandomQuestions()`. В самому процесі тестування варіанти відповіді автоматично міняють своє положення з метою уникнення конспектування положення вірних відповідей. За дану можливість

відповідають функції `GetAnswerCount()` та `Method()`. При завершенні тестування система перевіряє чи користувач відповів на усі запитання. Якщо ж ні, то автоматично переводить на запитання, на яке не було обрано варіанту відповіді. За це відповідає функція `CheckAnswer()`.

3. Архів. Після завершення тестування його результати попадають в архів. По них відповідальні особи аналізують отриманий бал (за допомогою функції `GetUserMarks()`) та друкують результати пройденого тестування, після чого підшивають їх в особову справу працівника.

Готових модулів тестування в мережі Інтернет є досить багато, але створення власного модуля дає можливість гнучкого налаштування процесу тестування в залежності від вимог організації.

**Обмін даними АІСУ ВПЗ.** При роботі з ВПЗ існує необхідність постійного обміну даними (наприклад відправка електронних відомостей по пенсіям, соціальним виплатам, різних типів оновлень АРМ ВЗ та ін.). Раніше це реалізовувалось за допомогою bat-файлів і обмін даними відбувався за наступною схемою: на ВПЗ надавався загальний доступ до папки, в яку необхідно здійснити переміщення даних; на робочому місці відповідальна за обмін даними особа вносила в bat-файл шлях до папки, в яку необхідно перемістити дані, шлях до файлів, які переміщують, та певні ключі і маски. Після таких операцій відповідальна особа запускала bat-файл, виконуючи таким чином переміщення даних за допомогою протоколу передачі даних FTP.

Недоліки описаної вище схеми наступні:

1. Надання загального доступу до папок на ВПЗ може призвести до зараження та розповсюдження вірусів через мережу на АРМ;

2. При виникненні необхідності зміни шляхів для обміну даними чи інших змін, необхідно здійснювати перезавантаження bat-файлу;

3. При переході дирекції підприємства на доменну систему (через налаштування політики безпеки доменної системи) обмін даними з комп'ютерами, які не входять в цю систему, через протокол FTP став неможливий, оскільки система блокує підключення до «ненадійних» систем. Іншими словами, прямий процес обміну даними через bat-файли став неможливим.

Для вирішення цієї проблеми в АІСУ ВПЗ було створено модуль обміну даними, який дозволяє автоматизувати процес обміну, описаний вище. Модуль складається з двох окремих частин, які відповідають архітектурі «клієнт-сервер», та використовують TCP-протокол обміну даними.

Клієнтом є сама АІС. В конкретних її довідниках відповідальною особою вносяться шляхи, за якими необхідно передати дані, та інші правила, які дозволять відфільтрувати дані, що відправляються. В процесі передачі даних клієнт підключається до сервера за допомогою функції `RunCory()` та відправляє дані за допомогою функції `SendFile()`. Даний процес використовує вбудовану в .NET-клас функцію `TcpClient()`. До відправки дані перетворюються в байти і, в залежності від типу передачі, до них додаються ID-команди, за допомогою яких сервер може визначити місце для переміщення даних.

Сервером є та частина АІСУ ВПЗ, яка відповідає тільки за отримання і обробку даних від клієнта та знаходиться на кожному АРМ ВПЗ. На відміну від клієнта, сервер написаний за допомогою .NET Framework 2.0. При старті комп'ютера сервер запускається автоматично та починає прослуховувати вхідні підключення за допомогою функції `Server()`. Після підключення клієнта до сервера здійснюється передача даних клієнтом за допомогою функції `Receive()`. Вона відповідає за прийом даних від клієнта та здійснює розподіл отриманих файлів в залежності від ID-команди.

Саме ПЗ сервера розроблено таким чином, щоб не заважати працівнику ВПЗ обслуговувати клієнтів. Тобто всі події прийому / обробки файлів є невидимими для працівника. Можливість закрити сервер (окрім завершення процесу) працівнику відділення виключена. Всі маніпуляції з налаштуваннями та завершення роботи сервера захищені паролем. Слід відмітити, що перед передачею даних при можливості доцільно архівувати дані, щоб зменшити їх розмір для передачі. Щоб не здійснювати пошук та розархівування таких даних на АРМ ВЗ в сервері АІСУ ВПЗ передбачений процес розархівування.

Таким чином було автоматизовано процес обміну даними між відповідальною за передачу даних особою та відділеннями поштового зв'язку. Процес передачі є непомітним для працівника ВПЗ і по закінченню отримання даних працівник отримує тільки спливаюче повідомлення про

отримання тих чи інших даних. На стороні клієнта інтерфейс передачі даних реалізований в основному через DataGridView (компонент .NET для перегляду таблиць). Для відображення прогресу передачі даних це компонент було розширено за допомогою класів DataGridViewProgressColumn() та DataGridViewProgressCell().

**Висновки.** Перед українським державним підприємством поштового зв'язку «Укрпошта» постало питання ефективного впровадження інформаційних технологій задля забезпечення своєчасного вироблення і прийняття управлінських рішень, вивільнення частини робочого часу працівників, поліпшення якості і оперативності управління, його дієвості тощо. Було здійснено аналіз потреб підприємства та розробку і впровадження автоматизованої інформаційної системи управління в інформаційний простір на рівні дирекції підприємства. Це дозволило вести контроль за роботою відділень поштового зв'язку, отримувати та відправляти актуальну на даний час інформацію та дані, виправляти помилки в роботі автоматизованого робочого місця відділення зв'язку та вести контроль за знаннями працівників підприємства.

1. Грицунов О. В. Інформаційні системи та технології: навч. посіб. для студентів за напрямом підготовки «Транспортні технології» / О. В. Грицунов; Харк. нац. акад. міськ. госп-ва. – Х.: ХНАМГ, 2010. – 222 с.
2. Калюга Є. В. Фінансово-господарський контроль у системі управління: Монографія. – К.: Ельга, Ніка-Центр, 2002. – 360 с.
3. ЗКриницький Р. И. Контроль и ревизия в условиях автоматизации бухгалтерского учета. – М.: Финансы и статистика, 1990. – 118 с.
4. Павлюк В. В., Сердюк В. М., Акаєв Ш. М. Контроль і ревізія. Навчальний посібник. – К.: Центр навчальної літератури, 2006. – 196 с.
5. Сопко В., Завгородній В. Організація бухгалтерського обліку, економічного контролю та аналізу: Підручник. – К.: КНЕУ, 2000. – 260 с.
6. Білуха М. Т. Курс аудиту: Підручник. – К.: Вища школа, Знання, 1998 – 574 с.
7. Бойко В. В. Проектирование информационной базы автоматизированной системы на основе СУБД / В. В. Бойко, В. М. Савинков – М.: Финансы и статистика, 1982. – 174 с.
8. Гайдамакин Н. А. Автоматизированные информационные системы, базы и банки данных. Вводный курс / Н. А. Гайдамакин: Учебное пособие. – М.: Гелиос АРВ, 2002. – 368 с.
9. Джексон Г. Проектирование реляционных баз данных для использования з микроЭВМ / Г. Джексон: Пер. с англ. – М.: Мир, 1991. – 252 с.
10. Райордан Р. Основы реляционных баз данных / Р. Рейордан. Пер. с англ. – М.: Издательско-торговый дом «Русская редакция», 2001. – 384 с.
11. Зенкевич О., Чанг И. Метод конечных элементов в теории сооружений и в механике сплошных сред. – М.: Недра, 1974. – 240 с.
12. Твердохлеб Н. Г. Информационное обеспечение менеджмента: Учеб. пособие. – М.: Финансы, 2002. – 224 с.

УДК 616.07 (075.8)

Рябіченко А.В.

Східноукраїнський національний університет ім. В. Даля

## ДОСЛІДЖЕННЯ ТЕРМОДИНАМІЧНИХ ПРОЦЕСІВ У ЛЮДСЬКОМУ ОРГАНІЗМІ МЕТОДОМ РЕОЛОГІЧНИХ ПЕРЕХОДІВ

**Рябіченко А. В. Дослідження термодинамічних процесів у людському організмі методом реологічних переходів.** Виконаний аналіз термодинамічних процесів в людському організмі і показана їх складність. Показано, що добові температурні залежності здорової людини змінюються за законами реологічного перенесення теплової енергії і визначаються її віком. Найбільша абсолютна зміна добової температури спостерігається у людей старшого віку, а мінімальна – у дитячому. На основі теорії реологічних перетворень і методу нульового градієнта отримана математична модель термодинамічного процесу здорової людини, яка адекватно описує експериментальні температурні зміни.

**Ключові слова:** температура, людина, процес, метод, здоров'я, організм, реологія, рівняння, енергія, вік.

**Рябиченко А. В. Исследование термодинамических процессов в человеческом организме методом реологических переходов.** Выполнен анализ термодинамических процессов в человеческом организме и показана их сложность. Показано, что суточные температурные зависимости здорового человека изменяются по законам реологического переноса тепловой энергии и определяются его возрастом. Наибольшее абсолютное изменение суточной температуры наблюдается у людей старшего возраста, а минимальная – в детском. На основании теории реологических преобразований и метода нулевого градиента получена математическая модель термодинамического процесса здорового человека, которая адекватно описывает экспериментальные температурные изменения.

**Ключевые слова:** температура, человек, процесс, метод, здоровье, организм, реология, уравнение, энергия, возраст.

**Ryabichenko A. The research of thermodynamic processes in the organism of person by the method of rheological transitions.** The analysis of thermodynamic processes in the organism of person is fulfilled and shows their complexity. It is shown that the diurnal temperature dependences of a healthy person changing by law of the rheological transfer of thermal energy is determined by its age. The largest absolute change in daily temperature is observed in older people, and the minimum - in the children's. Based on the theory of rheological transformation and the method of zero gradient gets the mathematical model of the thermodynamic process of a healthy person, which adequately describes the experimental temperature changes.

**Keywords:** temperature, person, process, method, health, organism, rheology, equation, energy, age.

**Вступ.** Термодинамічні процеси в кожному біологічному організмі, у тому числі й людському, є надзвичайно важливими для діагностування стану їх здоров'я. Прийнято [1], що номінальна середня температура людини рівна 36,6 °С. Для кожної людини номінальна температура тіла є індивідуальною. Як правило, діапазон зміни температур коливається від 36,2 °С до 36,7 °С. Як вказується в науковій літературі [1, 3], номінальна температура тіла людини на протязі доби є нестабільною: вранці вона мінімальна, а ввечері – максимальна. Тому у клінічній практиці, як правило, контролюється температура тіла людини зранку та ввечері. Різниця цих температур для здорової людини є незначною і складає на протязі доби приблизно 1,5 °С. Окрім того, зміна температури в таких же межах може здійснюватися й за рахунок фізичних навантажень, перегріву, вживання алкоголю та калорійних продуктів живлення. Таким чином, для здорової людини амплітуда коливання температури тіла на протязі доби практично є сталою і змінюється в залежності від дії впливових на організм факторів. Основним джерелом теплової енергії кожного біологічного організму є біохімічні процеси, які проходять в шлунково-кишковій системі. Теплова енергія від джерела передається організму як кровоносною так і лімфатичною системою і накопичується в м'язовій, кістковій та інших частинах тіла за рахунок стоку цієї енергії. Такий процес в біологічному організмі відноситься до реологічного, так як має місце рух матеріальних потоків (продукти живлення, вода і кисень), джерело теплової енергії (біохімічний реактор у формі шлунково-кишкової системи) і накопичувач цієї енергії, який складається з кровоносною, лімфатичною, м'язовою, кістковою системи тощо. За рахунок теплової енергії в організмі людини здійснюються масообмінні та енергетичні (теплові) процеси, наприклад перенесення мікроелементів та інших речовин, електричних зарядів для створення електромагнітного поля людини, живлення серцево-судинної системи та системи мозку [3, 4]. Біологічному організму притаманні інші джерела теплової енергії, котрі обумовлені як суцільними, так і локальними запальними захворюваннями. Такі захворювання створюють теплову енергію, котра теж передається кровоносною та лімфатичною системою. Окрім того, при таких захворюваннях досить часто створюються речовини, котрі при взаємодії з кров'ю чи лімфатичною рідиною виділяють теплову енергію, таким чином, ускладнюють процес захворювання. Звідси випливає, що у процесі захворювання організму доцільно контролювати

зміну температури в різних місцях людського організму, особливо там, де передбачається місце захворювання. Особливе значення для діагностики стану здоров'я людини є вивчення термодинамічних змін як здорового, так і хворого організму, характеру термодинамічних процесів при початковому захворюванні, а також частотних характеристик таких процесів, враховуючи те, що частотні зміни можуть визначати характер дії того чи іншого впливового фактора. Таким чином, розробка простих методів діагностування термодинамічного стану людського організму за перехідними процесами зміни температури тіла є важливою науковою задачею.

**Мета статті і постановка досліджень.** Метою статті є дослідження термодинамічних характеристик здорового людського організму на основі реологічних перетворень теплової енергії та методу нульового градієнта. Розрахунок перехідних процесів складних об'єктів описаний в [5]. Метод заснований на розкладанні нелінійного диференціального рівняння методом нульового градієнта на систему простих лінійних диференціальних рівнянь, як правило, першого та другого порядку, котрі розв'язуються сумісно за відомими аналітичними залежностями.

Якщо основне джерело теплової енергії працює стабільно, то процес її перенесення можна описати наступним рівнянням [6-8]:

$$\frac{\partial E_T(\theta, V)}{\partial \theta} = -aLV_{\Pi} \frac{\partial^2 E_T(\theta, V)}{\partial V^2} - \bar{v}S \frac{\partial E_T(\theta, V)}{\partial V} + \gamma_E(t), \quad (1)$$

де  $E_T(\theta, V)$  - кількість теплової енергії, яка передається від джерела в об'єм  $V$  за час  $\theta$ ,  $a = \lambda / c_p \rho$  - температуропровідність біологічного матеріалу;  $L$  - довжина шляху перенесення теплової енергії;  $V_{\Pi}$  - об'єм перебування теплової енергії;  $\bar{v}$  - середня швидкість перенесення конвекційної складової теплової енергії;  $S$  - площа перенесення;  $\gamma_E(t)$  - швидкість стоку теплової енергії за час  $t$ .

У першому наближенні приймемо, що перенесення теплової енергії є двоступеневим, тобто від джерела до стінки, а далі від стінки до рідинного потоку (крові та лімфи). Тоді для швидкості стоку теплової енергії  $E_{TC}(t)$  маємо:

$$\gamma_E(t) = \tau_{2c}^2 \frac{d^3 E_{TC}(t)}{dt^3} + \tau_{1c} \frac{d^2 E_{TC}(t)}{dt^2} + \frac{dE_{TC}(t)}{dt}, \quad (2)$$

де  $\tau_{1c}, \tau_{2c}$  - сталі часу перенесення теплової енергії за рахунок стоку.

Підставивши (2) у рівняння (1), отримуємо:

$$\begin{aligned} \frac{\partial E_T(\theta, V)}{\partial \theta} + aLV_{\Pi} \frac{\partial^2 E_T(\theta, V)}{\partial V^2} + \bar{v}S \frac{\partial E_T(\theta, V)}{\partial V} = \\ = \tau_{2c}^2 \frac{d^3 E_{TC}(t)}{dt^3} + \tau_{1c} \frac{d^2 E_{TC}(t)}{dt^2} + \frac{dE_{TC}(t)}{dt}. \end{aligned} \quad (3)$$

Рівняння (3) є нелінійним і, як вказується в [9], загального аналітичного рішення не має. Задача полягає у дослідженні термодинамічного стану людського організму у патологічно нормальному (здоровому) стані при дії на нього факторів, котрі призводять до суттєвих температурних змін.

**Термодинамічні моделі нормального стану людини.** Для аналітичного розв'язку рівняння (3) використаємо метод нульового градієнта [6], згідно з яким це рівняння при відповідних початкових умовах приводиться до системи звичайних диференціальних рівнянь:

$$\frac{\partial E_T(\theta, V)}{\partial \theta} + aLV_{\Pi} \frac{\partial^2 E_T(\theta, V)}{\partial V^2} + \bar{v}S \frac{\partial E_T(\theta, V)}{\partial V} = 0; \quad (4)$$

$$\tau_{2c}^2(\Phi_i) \frac{d^2 E_T(t)}{dt^2} + \tau_{1c}(\Phi_i) \frac{dE_T(t)}{dt} + E_T(t) = E_T(\theta, V). \quad (5)$$

Важливим для діагностики стану здоров'я людини є характер перенесення теплової енергії від джерела до стоку, тобто кровоносної системи. Якщо швидкість перенесення теплової енергії від джерела є сталою (система працює нормально), то позначивши  $\partial E_T(\theta, V) / \partial \theta = E_T(\theta, V) / \theta_{\Pi}$  у рівнянні (4), маємо:

$$aLV_{\Pi} \frac{\partial^2 E_T(\theta, V)}{\partial V^2} + \bar{v}S \frac{\partial E_T(\theta, V)}{\partial V} + \frac{E_T(\theta, V)}{\theta_{\Pi}} = 0, \quad (6)$$

де  $\theta_{\Pi}$  - час перебування теплової енергії в об'ємі  $V_{\Pi}$ .

Нехай кількість теплової енергії, яка передається до кровоносної системи, дорівнює кількості теплової енергії, котра переноситься судинами. Тоді, позначивши  $\partial V = F_K \partial \theta$ , де  $F_K$  - об'ємна витрата крові, рівняння (6) набуває такого вигляду:

$$\frac{aLV_{\Pi}}{F_K^2} \frac{\partial^2 E_T(\theta, V)}{\partial \theta^2} + \frac{\bar{v}S}{F_K} \frac{\partial E_T(\theta, V)}{\partial \theta} + \frac{1}{\theta_{\Pi}} E_T(\theta, V) = E_{T0}, \quad (7)$$

де  $E_{T0}$  - кількість теплової енергії, яка виділяється в результаті біохімічного реакційного процесу.

Позначивши сталі часу  $\tau_{2K}^2 = aLV_{\Pi}\theta_{\Pi} / F_K^2$  і  $\tau_{1K} = \bar{v}S\theta_{\Pi} / F_K$ , рівняння (7) приймає наступну форму лінійного диференціального рівняння другого порядку:

$$\tau_{2K}^2 \frac{\partial^2 E_T(\theta, V)}{\partial \theta^2} + \tau_{1K} \frac{\partial E_T(\theta, V)}{\partial \theta} + E_T(\theta, V) = E_{T0}. \quad (8)$$

Як видно з (8), характер процесу перенесення теплової енергії від джерела до кровоносної системи залежить від сталих часу  $\tau_{1K}$  і  $\tau_{2K}$ . Для здорової людини швидкість стоку теплової енергії є практично сталою, тобто  $\partial E_T(\theta, V) / \partial \theta = const$ . Тоді рівняння (8) приводиться до такої спрощеної форми:

$$\tau_{1K} \frac{\partial E_T(\theta, V)}{\partial \theta} + E_T(\theta, V) = E_{T0}. \quad (9)$$

Рішенням рівняння (9) при відповідних початкових умовах буде:

$$E_T(\theta, V) = E_{T0}(1 - \exp(-t / \tau_{1K})). \quad (10)$$

Стік теплової енергії здорової людини, як правило, є двостадійним аперіодичним, тобто  $\tau_{1K} / \tau_{2K} > 2$ . У цьому випадку термодинамічний процес описуватиметься таким рівнянням:

$$E_T(t) = E_T(\theta, V) \left[ \frac{P_2}{p_2 - p_1} \exp(p_1 t) + \frac{P_1}{p_2 - p_1} \exp(p_2 t) \right], \quad (11)$$

де  $p_{1,2} = -\frac{\tau_{1K}}{2\tau_{2K}^2} \pm \sqrt{\left(\frac{\tau_{1K}}{2\tau_{2K}^2}\right)^2 - \frac{1}{\tau_{2K}^2}}$  - корені характеристичного рівняння.

Підставивши рівняння (10) у (11), маємо:

$$E_T(t) = kE_{T0} [1 - \exp(tp_3)]^n \left[ \frac{P_2}{p_2 - p_1} \exp(p_1 t) + \frac{P_1}{p_2 - p_1} \exp(p_2 t) \right], \quad (12)$$

де  $k$  - коефіцієнт передачі;  $p_3 = 1 / \tau_{1K}$ ;  $n$  - показник термодинамічного захисту людини.

На рис. 1,а приведені оброблені методом кореляційного аналізу результати експериментальних спостережень за температурою здорового людського організму з різними

віковими групами: крива 1 – діти від 5 до 8 років; крива 2 – діти від 10 до 16 років; крива 3 – дорослі від 25 до 45 років і крива 4 – дорослі старші 45 років. Визначався приріст температури відносно нормальної  $T_H=36,6\text{ }^{\circ}\text{C}$ . Отримані результати потребують, звичайно, ще обґрунтованого вивчення, але перші їх висновки говорять про наступне. По-перше, чим менший вік людського організму, тим більший час початкового зміщення температурних кривих (наприклад, зранку) (точки  $a_1-a_4$ ), при котрому починається підвищення температури. При цьому максимальна амплітуда підвищення температури (точки  $b_1-b_4$ ) з віком людини збільшується і зміщується вліво. Точки  $c_1-c_4$  умовно характеризують температуру в кінці доби. На рис. 1,б приведені такі ж температурні криві, розраховані за формулою (12). Значення коренів  $p_1$ ,  $p_2$  і  $p_3$  визначалися методом підбору за максимальними температурами в точках « $b_1-b_4$ ». Порівнюючи криві рис. 1,а і 1,б, бачимо їх подібність. Це дає підставу стверджувати про можливість описувати термодинамічний стан людського організму математичними залежностями. Викликає зацікавленість наявність показника ступені « $n$ » у рівнянні (12), який назовемо віковим показником термодинамічного режиму людини.

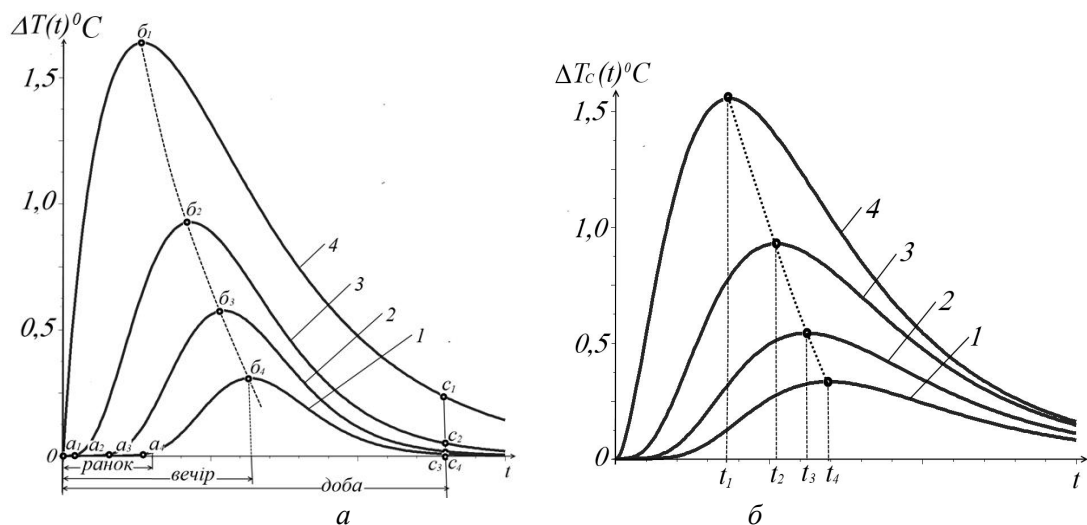


Рис. 1. Термодинамічні криві зміни теплової енергії людини при нормальному стані здоров'я експериментальні (а) і теоретичні (б)

Попередні висновки пов'язують значення цього показника з віком людини. У результаті його визначення встановлено, що для прийнятих при досліджуванні вікових груп показник є близьким до квадратичного. Так, наприклад для людей старшої групи показник  $n_4 \approx 2$ , середньої  $n_3 \approx 4$ , підліткової  $n_2 \approx 6$  і дитячої -  $n_1 \approx 8$ , котрі відповідають кривим 1 – 4 на рис. 1,б. Така залежність потребує ще додаткового аналізу та вивчення. Виходячи зі сказаного, можна зробити попереднє припущення: у людському організмі людини закладені багатократні захисні функції, яких найбільше мають діти. З віком частина цих функцій зменшується, а людський організм стає більш вразливим до захворювань під дією різних внутрішніх і зовнішніх факторів. Подальші дослідження полягають у вивченні дії впливових факторів процесу стоку теплової енергії на характер зміни термодинамічних кривих людини.

**Висновки.** Виконані експериментальні та теоретичні дослідження термодинамічних процесів людського організму та показано, що температура людини на протязі дня зростає нелінійно й у вечірній час досягає максимального значення. Абсолютне збільшення температури залежить від вікового стану людини, який умовно розділений на чотири групи: до першої групи віднесені діти від 5 до 8 років, до другої – (підліткова група) від 10 до 16 років, до третьої – люди середнього віку від 25 до 45 років і 4 група – люди старше 45 років. Показано, що з віком людини її максимальна добова температура може зростати приблизно на  $1,5\text{ }^{\circ}\text{C}$  від нормальної. Окрім того, настання максимуму цієї температури зміщується вліво. Чим менший вік людини, тим пізніше починає зростати ранкова температура. Показано, що характер зміни температури людського організму у першу чергу визначається процесом перенесення теплової енергії від джерела (шлунково-кишкової системи) до кровоносної системи і визначається, в основному, конвекційною

складовою потоку теплової енергії. Установлено, що для здорової людини швидкість перенесення теплової енергії практично є сталою. Теоретичні дослідження процесу перенесення теплової енергії показали, що її стік залежить від реологічного переходу, який можна описати експоненціальною функцією. Показано, що рівняння реологічного перенесення теплової енергії за формою та характером з достатньою точністю описують експериментальні криві термодинамічного режиму здорової людини, що дозволяє зробити висновок про можливість їх використання для клінічних досліджень з метою діагностики стану її здоров'я. Потрібно відмітити, що виконані дослідження є попередніми, котрі бажано було би продовжити з метою вдосконалення критерію оцінки стану здоров'я людини. Не зовсім вивченим є процес квадратичної зміни процесу перенесення теплової енергії у залежності від віку людини, що приводить до підвищення її температури та зміщення термодинамічної кривої вліво.

1. Василенко В. Х. Пропедевтика внутренних болезней / В. Х. Василенко, А. Л. Гребенева, Н. Д. Михайлова и др. – М.: Медицина, 1974. – 528 с.
2. Ремизов А. Н. Медицинская и биологическая физика / А. Н. Ремизов. – М.: Высшая школа, 1987. – 638 с.
3. Орлов В. Н. Руководство по электрокардиографии / В. Н. Орлов. – М.: Медицина, 1984. – 526 с.
4. Виноградов А.В. Дифференциальный диагноз внутренних болезней. / А. В. Виноградов. – М.: Медицина, 1980. – 836 с.
5. Стенцель Й. І. Автоматизація технологічних процесів хімічних виробництв / Й. І. Стенцель. – К.: ІСДО, 1995. – 360 с.
6. Стенцель Й.І. Математичне моделювання хімічних процесів на основі теорії реологічних переходів / Й. І. Стенцель // Вісник СНУ імені В.Даля. Науковий збірник. – №5 (111). – Ч.2. – 2007. – С. 91-97.
7. Лыков А.В. Теория теплопроводности / А. В. Лыков. – М.: Высш. шк., 1967. – 599 с.
8. Эккерт Э. Р. Теория тепло-и массообмена / Э. Р. Эккерт, Р. М. Дрейк. - М.: Госэнергоиздат, 1962. – 562 с.
9. Вайнберг А. М. Математическое моделирование процессов переноса. Решение нелинейных краевых задач / А. М. Вайнберг. – Москва-Иерусалим, 2009. – 210 с.



УДК 004.942 : 519.876

Сафонова С.А., к.т.н.

Восточноукраинский национальный университет им. В. Даля

## МЕТОД ОПРЕДЕЛЕНИЯ БЕЗОПАСНОГО ИСПОЛЬЗОВАНИЯ ПРЕДОХРАНИТЕЛЬНЫХ КЛАПАНОВ

**Сафонова С.О. Метод визначення безпечного використання запобіжних клапанів.** У статті пропонується метод визначення наслідків аварійних викидів для підтримки прийняття рішень щодо конструкційних особливостей і компоновки запобіжних пружинних клапанів, заснований на моделюванні витікання газової фази з клапана і одночасного розсіювання домішки в атмосфері з урахуванням найгірших погодних умов.

**Ключові слова:** розсіювання, витікання, зона ураження, технологічний процес, моделювання, запобіжне обладнання.

**Сафонова С.А. Метод определения безопасного использования предохранительных клапанов.** В статье предлагается метод определения последствий аварийных выбросов для поддержки принятия решений относительно конструкционных особенностей и компоновки предохранительных пружинных клапанов, основанный на моделировании истечения газовой фазы из клапана и одновременного рассеяния примеси в атмосфере с учетом наихудших погодных условий.

**Ключевые слова:** рассеяние, истечение, зона поражения, технологический процесс, моделирование, предохранительное оборудование.

### **Safonova S.A. A method for determination of safe use of the safety valves.**

The article proposes a method determining the consequences of emergency emissions for decision-support regarding design features and layout of the safety valve springs, based on the simulation of the gas phase of the expiration valve and simultaneous dispersion of contaminants in the atmosphere with a glance of the worst weather conditions.

**Keywords:** dispersion, flow, affected area, process, modeling, safety equipment.

**Постановка проблеми.** В соответствии с ГОСТ 12.2.085-2002 "Сосуды, работающие под давлением. Клапаны предохранительные. Требования безопасности" оборудование, находящееся под давлением свыше 0,07 Мпа, должно эксплуатироваться с установленными предохранительными клапанами различных конструкций и характеристик в зависимости от условий и сред технологических процессов.

В основном такие клапана ставятся непосредственно на емкости или арматуру систем высокого давления и должны соответствовать требованиям нормативной документации, расчетам пропускной способности и конструктивным требованиям, позволяющим обеспечить надежность, прочность и работоспособность. Однако, такие требования приводят к тому, что сброс парогазовой фазы (ПГФ) для снижения избыточного давления происходит непосредственно в атмосферу вблизи выпускного отверстия клапана. Практически никогда не рассматривается вопрос последствий такого выброса, если сбрасываемая ПГФ является чрезвычайно токсичной или взрывоопасной. Так, на предприятии Miro (Германия) произошел сброс значительной части HF через предохранительный клапан, установленный на емкости на высоте 10 метров над уровнем земли, что привело к тяжелым отравлениям двух человек на расстоянии до 170 метров от места выброса. В Дзержинске (РФ) произошел сброс окиси этилена, приведший к отравлению сотрудника предприятия и вспышки в области колонны синтеза в результате экзотермической реакции разложения.

Изменение конструкции клапана, расположения выходного отверстия, выбор условий пропускной способности позволяют предупредить негативные последствия аварийных сбросов.

В статье предлагается метод определения последствий таких выбросов для поддержки принятия решений относительно конструкционных особенностей и компоновки предохранительных пружинных клапанов (ППК), основанный на моделировании истечения газовой фазы из клапана и одновременного рассеяния примеси в атмосфере с учетом наихудших погодных условий.

**Целью** разработки является математическое обеспечение процесса поддержки решения и выработки рекомендаций относительно безопасной эксплуатации предохранительных клапанов.

**Изложение основного материала и обоснование полученных результатов исследования.**

Основные результаты исследования заключаются в разработке модели истечения газовой фазы из сложного оборудования, а также методики, позволяющей определить зоны поражений токсичной примесью или распространения горючей взрывоопасной примеси в атмосфере.

Для формалізації *моделі истечения газовой фазы из сложного оборудования* приняты следующие предположения:

- истечение газовой фазы – изотермическое;
- известны - ограниченный общий объем емкости  $V_0$ , начальное давление  $P$ , температура  $T$ , состав смеси (постоянен в течение времени истечения);
- истечение возможно через систему трубопроводов, запорной арматуры и другие местные и гидравлические сопротивления;
- изменение давления в емкости происходит из-за уменьшения массы газовой фазы, выбрасываемой из емкости через отверстие истечения площадью  $F_2$ , и увеличения массы газовой фазы за счет прихода через отверстие подпитки площадью  $F_1$ .
- начальное давление газовой фазы в емкости соответствует давлению источника  $P_s$ , которое не меняется в течение всего времени существования этого источника (до перекрытия). Внешнее давление равно  $P_a$ .

Для расчета расхода истечения газа предлагается преобразовать стационарное уравнение (1) в динамическое с учетом изменения массы газовой фазы и считать массовый расход по зависимости (2) [1]:

$$q = \mu \cdot F \cdot \psi \sqrt{\frac{P}{v}}, \quad (1)$$

- где  $\mu$  - коэффициент расхода;  
 $F$  - площадь плоскости отверстия выброса ( $\text{м}^2$ );  
 $P$  - давление внутри емкости (Па);  
 $v$  - удельный объем емкости ( $\text{м}^3/\text{кг}$ );  
 $\psi$  - коэффициент, который определяется в соответствии с [2].

$$\frac{dM(t)}{dt} = \mu \cdot F \cdot \psi(t) \sqrt{\frac{P(t)}{v(t)}}. \quad (2)$$

Представив массовый расход из емкости в конечных разностях, получаем зависимость (3):

$$M_\tau = M_{\tau-1} - \mu_2 \cdot F_2 \cdot \psi_{\tau-1} \sqrt{\frac{P_{\tau-1}}{v_{\tau-1}}} \cdot \Delta\tau, \quad (3)$$

- где  $M_{\tau-1}$  - масса газа в емкости в предыдущий момент времени (кг);  
 $M_\tau$  - масса газа в емкости в расчетный момент времени (кг);

Учитывая что:

$$v_{\tau-1} = \frac{R \cdot T}{M \cdot P_{\tau-1}}, \quad (4)$$

- где  $R$  - универсальная газовая постоянная (Дж/моль К);  
 $M$  - молекулярная масса выбрасываемого газа (кг/моль);  
 $P_{\tau-1}$  - давление в емкости в предыдущий момент времени (Па),

получаем для расчета массового расхода из емкости выражение (5):

$$M_\tau = M_{\tau-1} - \mu_2 \cdot F_2 \cdot \psi_{\tau-1} \cdot P_{\tau-1} \cdot \sqrt{\frac{M}{R \cdot T}} \cdot \Delta\tau. \quad (5)$$

Увеличение массы газовой фазы в емкости за счет прихода через отверстие подпитки будет рассчитано как:

$$\Delta M_\tau = \mu_1 \cdot F_1 \cdot \psi_{1\tau} \cdot P_s \cdot \sqrt{\frac{M}{R \cdot T}} \cdot \Delta\tau. \quad (6)$$

Результирующая масса вещества в емкости определяется как:

$$M_\tau = M_{\tau-1} + \Delta M_\tau. \quad (7)$$

Давление в емкости перерасчитывается с учетом дополнительной массы:

$$P_{\tau} = \frac{M_{\tau} \cdot R \cdot T}{M \cdot V}. \quad (8)$$

Расчет производится до указанного экспертом физического времени окончания процесса или ручной остановки.

Расчет истечения газа через систему местных и гидравлических сопротивлений основан на представлении о перепаде давлений на гидравлическом сопротивлении [3-5]:

$$P_s - P_a = \rho \cdot g \cdot h. \quad (9)$$

$h$  может быть выражена через скорость потока и сопротивление:

$$h = \frac{\bar{w}^2}{2g} \cdot \left( \sum_{i=1}^m \lambda_i \cdot \frac{l_i}{d_i} + \sum_{j=1}^n \zeta_j \right). \quad (10)$$

Для расчета необходимо знать значения набора переменных элементов в системе сопротивления потоку.

Среднюю скорость  $\bar{w}$  (м/с) можно вычислить, зная перепад давлений между емкостью и средой истечения, из уравнения:

$$\left( \frac{P_s - P_a}{P_s + P_a} \right) \cdot \frac{4RT}{M} = \bar{w}^2 \cdot \left( 0.11 \cdot \sum_{i=1}^m \left( \frac{\Delta_i \cdot 10^{-3}}{d_i} + \frac{68 \cdot \nu}{\bar{w} \cdot d_i} \right)^{0.25} \cdot \frac{l_i}{d_i} + \sum_{j=1}^n \zeta_j \right), \quad (11)$$

где

$$\bar{\rho} = \frac{P_s + P_a}{2} \cdot \frac{M}{RT} - \text{средняя плотность потока (кг/м}^3\text{);}$$

$$\lambda = 0.11 \left( \frac{\Delta \cdot 10^{-3}}{d} + \frac{68}{\text{Re}} \right)^{0.25} - \text{гидравлический коэффициент трения (формула}$$

Альтштуля);

$$\text{Re} = \frac{\bar{w} \cdot d}{\nu} - \text{число Рейнольдса;}$$

$l_i$  - длина  $i$ -го участка трубопровода (м),  $d_i$  - диаметр  $i$ -го участка трубопровода (м),  $\Delta$  - эквивалентная абсолютная шероховатость трубы (мм) [3],  $\nu$  - кинематическая вязкость газа ( $\text{м}^2/\text{с}$ ),  $\zeta_j$  - коэффициент местного сопротивления [4],  $T$  - температура газа внутри оборудования (К).

Массовый расход системы равен:

$$\frac{dM}{dt} = \bar{w} \cdot \frac{\pi \cdot \left( \sum_{i=1}^m d_i \right)^2}{8} \cdot \frac{(P_s + P_a)M}{RT}. \quad (12)$$

Потеря массы из емкости в единицу времени рассчитывается, как и в предыдущем примере.

Данная модель предназначена для определения количества парогазовой фазы, выбрасываемой из оборудования при аварии. Эти данные являются входными для моделирования рассеяния ОХВ в атмосфере.

**Методики, позволяющие определить зоны поражений токсичной примесью или распространения горючей взрывоопасной примеси в атмосфере**, можно разделить на численные модели класса CFD [6-8], модели класса аналитических оценок [9-14], оценочные эмпирические [15].

Численное моделирование требует много времени для получения динамических характеристик распространения примеси ОХВ в атмосфере. Поэтому, применение моделей класса CFD исключается. Оценочные эмпирические модели устарели и дают результат, не отвечающий требованиям к качеству прогноза, что не позволяет эффективно ликвидировать последствия выброса.

Поступление парогазовой фазы в атмосферу сопровождается процессом смешения и рассеяния примеси под воздействием ветра и атмосферных течений. Предлагается метод определения параметров распространения опасных химических веществ на базе гауссовой модели Ван Ульдена и Холстлага с учетом действительного состояния атмосферы на момент аварии, возможных колебаний направления и скорости ветра [13, 14].

На основе обработки экспериментальных данных, представленных в работах Берлянда и Бызиной, посвященных атмосферной турбулентности, предлагаются зависимости колебания направлении распространения облака опасных веществ от скорости ветра и времени усреднения. Используя аппроксимацию при обработке данных, ставим в соответствие им функцию отклонения направления ветра от скорости (13) (коэффициенты представлены в табл.1):

$$u(v) = a_i^1 \cdot \exp(a_i^2 \cdot v) + a_i^3, \quad (13)$$

где  $v$  - скорость ветра на высоте 1 метр над поверхностью земли, м/с. Рассчитывается согласно профилю скоростей [16].

**Таблица 1. Коэффициенты функции отклонения направления ветра в зависимости от скорости и состояния атмосферы**

$i$	Режим устойчивости атмосферы	Коэффициент		
		$a_i^1$	$a_i^2$	$a_i^3$
1	Конвекция (неустойчивое состояние)	16,63	-0,26	0,345
2	Изотермия (равновесно состояние)	6,858	-0,29	1,22
3	Инверсия (устойчивое состояние)	5,519	-0,38	0,8

Отклонение угла направления ветра в зависимости от времени усреднения предлагается определять линейной функцией (14) (коэффициенты представлены в табл.2):

$$u(T) = b_j^1 \cdot T + b_j^2, \quad (14)$$

где  $T$  - время усреднения в минутах.

**Таблица 2. Коэффициенты функции отклонения угла направления ветра в зависимости от времени усреднения и состояния атмосферы**

$j$	Режим устойчивости атмосферы	Коэффициент	
		$b_j^1$	$b_j^2$
1	Конвекция (неустойчивое состояние)	0,118	8,118
2	Изотермия (равновесно состояние)	0,125	2,5
3	Инверсия (устойчивое состояние)	0	0

Общее отклонение направления ветра при заданном времени усреднения и скорости ветра составляет (в градусах):

$$u(v, T) = u(v) + u(T). \quad (15)$$

При расчетах по методам, приведенным в [17], поле заражения местности по пороговой дозе выглядит приближено как эллипс. Половина угла раскрытия сектора составляет 11 градусов.

Учитывая общее отклонение направления ветра при заданном времени усреднения и скорости ветра, полный угол раскрытия сектора определяется как:

$$S(v, T) = (11 + u(v) + u(T)) \cdot 2. \quad (16)$$

Зона возможного поражения определяется наложением сектора на карту с заданным средним направлением ветра.

Скорость, направление ветра, температуру окружающей среды предлагается определять с помощью автоматической цифровой метеостанции

Для полного набора данных необходимо еще знать время суток (день, ночь) и наличие облачности. По умолчанию можно предполагать существование облачности, т.к. этот параметр вносит в прогноз незначительный вклад.

Рассеяние считать для скорости ветра 1 м/с, а время усреднения выбирать как отношение скорости ветра к глубине распространения токсичной или взрывоопасной границы примеси.

Моделирование с использованием изложенных методов позволяет определить пространственные характеристики зон заражения и загазованности взрывоопасной примесью и выработать такие конструктивные решения, которые позволяют избежать отравлений персонала установок и возможных аварий, связанных с воспламенением примеси в атмосфере.

**Выводы и перспективы дальнейших исследований.** Таким образом, используя предложенные модели, возможно провести анализ технических решений эксплуатации предохранительного оборудования с учетом реальных условий технологического процесса и внешних условий окружающей среды, а также с учетом расположения обслуживающего персонала и близлежащего оборудования.

Моделирование наихудших условий возможных нештатных ситуаций позволяет оценить параметры и характеристики негативных последствий срабатывания предохранительного оборудования и выработать ограничительные требования к его эксплуатации. На основании таких ограничений возможна разработка технических предложений, позволяющих избежать жертв и разрушений при реализации нештатных ситуаций.

1. Крутов В. И. Техническая термодинамика под ред. В. И. Крутова / Учебник для вузов. – М.: «Высш. школа». – 1971. – 176 с.
2. Дейч М. Е. Техническая газодинамика / М. Е. Дейч. – М.: Энергия. – 1974. – 453 с.
3. Григорьев В. А., Зорин В. М. Теоретические основы теплотехники. Теплотехнический эксперимент.: Справочник / Под общ. Ред. чл.-корр. АН СССР В. А. Григорьева, В. М. Зорина // 2-е изд. – М.: Энергоатомиздат. – 1988. – 560 с.
4. Альшгтуль А. Д. Гидравлическое сопротивление / А. Д. Альшгтуль // – М.: Стройиздат. – 1973. – 287 с.
5. Идельчик И. Е. Справочник по гидравлическим сопротивлениям / И. Е. Идельчик // – М.: Машиностроение. – 1975. – 237 с.
6. Fairhurst S., Turner R. M. Toxicological assessments in relation to major hazards. / S. Fairhurst, R. M. Turner // Journal of Hazardous Materials 33. – 1993, – p. 215-227.
7. Granovskiy E. A., Lifar V. A., Skob Yu. A., Ugryumov M. L. Computational modeling of pressure effects from hydrogen explosions / E. A. Granovskiy., V.A. Lifar, Yu.A. Skob, M.L. Ugryumov // Abstracts Book and CD-ROM Proceedings of the 2 International Conference on Hydrogen Safety. – San Sebastian (Spain). – 2007. – 15 p. (ICHS Paper No. 13).
8. Skob Yu. A., Ugryumov M. L., Korobchynskiy K. P., Shentsov V. V., Granovskiy E. A., Lifar V. A. Numerical Modeling of Hydrogen Deflagration Dynamics in Enclosed Space / Yu. A. Skob, M. L. Ugryumov, K. P. Korobchynskiy, V. V. Shentsov, E. A. Granovskiy, V. A. Lyfar // Abstracts Book and CD-ROM Proceedings of the 3-rd International Conference on Hydrogen Safety. – Ajaccio (France). – 2009. – 12 p. (ICHS Paper No. 182).
9. Scire J. S., Strimaitis D. G., Yamartino R. J. A User's Guide for the CALPUFF Dispersion Model (V. 5). / J. S. Scire, D. G. Strimaitis, R. J. Yamartino. // – Earth Tech, Inc. – 2000. – P. 521.
10. Granovsky E. A., Lyfar V. A., Vasilyuk E. V. Industrial accident modeling: consequences and risk / E. A. Granovskiy, V. A. Lyfar, E. V. Vasilyuk // Prevention of Hazardous Fires and Explosions. – Kluwer Academic Publishers (Netherlands). – 1999. – P.183–197.
11. Turren P. B. Workbook of Atmospheric Dispersion Estimates, Public. / P. B. Turren // Health Service Publication N 999-AP-26, U.S. Department of Health, Education and Welfare. – 1969.
12. U.S. Environmental Protection Agency, Guideline for Air Quality Models, OAQPS Guideline Series, Research Triangle Park, NC. – 1980.
13. Берлянд М. Е. Прогноз и регулирование загрязнения атмосферы / М. Е. Берлянд // – Л.: Гидрометеиздат. – 1985. – 272 с.
14. Суттон О. Г. Микрометеорология: исследование физических процессов в нижних слоях атмосферы / О. Г. Суттон // Л.: Гидрометеиздат. – 1958. – с.356
15. Руководящий документ. Методика прогнозирования масштабов заражения сильнодействующими ядовитыми веществами при авариях (разрушениях) на химически опасных объектах и транспорте. РД 52.04.253-90 от 21.05.90 г. – Л.: Гидрометеиздат. – 1991. – 23 с.
16. Ньюстад Ф. Т. М., Ван Допа Х. Атмосферная турбулентность и моделирование распространения примесей / Под ред. Ф. Т. М. Ньюстада и Х. Ван Допа. – Л.: Гидрометеиздат. – 1985. – 351 с.
17. Методика оценки последствий химических аварий (Методика «Токси-2.2», утв. НТЦ "Промышленная безопасность", согл. Госгортехнадзором России) в сборнике «Методики оценки последствий аварий на опасных производственных объектах»: Сборник документов. Серия 27. Выпуск 2 / Колл. авт. – 2-е изд., испр. и доп. – М.:ГУП НТЦ «Промышленная безопасность» . – 2002. – 208 с.

УДК 004.056 : 519.2

Фауре Э.В., Щерба А.И., Лавданский А.А

Черкасский государственный технологический университет

## ОЦЕНКА СТАТИСТИЧЕСКИХ ХАРАКТЕРИСТИК ПОСЛЕДОВАТЕЛЬНОСТИ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ, ПОРОЖДЕННОЙ КОМБИНАЦИОННЫМ ГЕНЕРАТОРОМ

**Фауре Э.В., Щерба А.И., Лавданский А.А. Оценка статистических характеристик последовательности псевдослучайных чисел, порожденной комбинационным генератором** Генераторы псевдослучайных последовательностей находят применение для решения широкого круга практических задач. Однако последовательности псевдослучайных чисел не могут быть использованы без детального исследования статистических свойств и сравнительных количественных оценок. В работе рассмотрен комбинационный метод генерации псевдослучайных чисел. Рассмотрены способы построения исходных таблиц для комбинационного генератора с помощью существующих генераторов псевдослучайных и случайных чисел. Проведен анализ статистических характеристик последовательности чисел на выходе комбинационного генератора с помощью специализированного пакета тестирования NIST. Произведена оценка исследуемых последовательностей с помощью непараметрических критериев, таких как критерий знаков и критерий серий. Результаты, полученные в работе, позволяют использовать комбинационный генератор в задачах, требующих высокого качества последовательностей псевдослучайных чисел.

**Ключевые слова:** последовательность псевдослучайных чисел, комбинационный генератор, тестирование, непараметрический критерий.

**Фауре Е.В., Щерба А.І., Лавданський А.О. Оцінка статистичних характеристик послідовності псевдовипадкових чисел, що породжена комбінаційним генератором** Генератори псевдовипадкових послідовностей знаходять застосування для вирішення широкого кола практичних задач. Однак послідовності псевдовипадкових чисел не можуть бути використані без детального дослідження статистичних властивостей і порівняльних кількісних оцінок. У роботі розглянуто комбінаційний метод генерації псевдовипадкових чисел. Розглянуто способи побудови вихідних таблиць для комбінаційного генератора за допомогою існуючих генераторів псевдовипадкових і випадкових чисел. Проведено аналіз статистичних характеристик послідовності чисел на виході комбінаційного генератора за допомогою спеціалізованого пакета тестування NIST. Проведена оцінка досліджуваних послідовностей за допомогою непараметричних критеріїв, таких як критерій знаків і критерій серий. Результати, отримані в роботі, дозволяють використовувати комбінаційний генератор в задачах, що потребують високої якості послідовностей псевдовипадкових чисел.

**Ключові слова:** послідовність псевдовипадкових чисел, комбінаційний генератор, тестування, непараметричний критерій.

**E. Faure, A. Shcherba, A. Lavdanskiy Cherkasy state technological university Estimation of statistical characteristics for pseudorandom numbers sequence generated by combinational generator** Generators of pseudorandom sequences are used for a wide range of practical problems. However, the sequence of pseudorandom numbers can not be used without detailed study of the statistical properties of the comparative and quantitative assessments. In this paper we consider the combinational method of pseudorandom numbers generating. Methods of constructing the source tables for combinational generator with existing generators of pseudorandom and random numbers are reviewed. The analysis of the statistical characteristics of the output numbers sequence of combinational generator using specialized test suite NIST is performed. The estimation of the test sequences using nonparametric tests such as the sign test and runs test is performed. The results obtained in this work allow the use of the combinational generator in tasks requiring high quality of pseudorandom numbers.

**Keywords:** pseudorandom numbers sequence, combinational generator, testing, nonparametric test.

### Введение

Генераторы псевдослучайных чисел (ПСЧ) широко используются для решения большого круга практических задач, таких как задачи защиты информации, имитационного моделирования и т.д. При этом результат решения задачи значительно зависит от качества используемого генератора. Качественный генератор ПСЧ должен производить последовательность чисел, сравнимую по своим статистическим характеристикам с последовательностью случайных чисел, порождаемых естественными (природными) источниками, и при этом быть воспроизводимым. Существующие генераторы ПСЧ обладают многими характеристиками естественных источников случайных чисел, но на данный момент времени не существует генератора ПСЧ, неотличимого по своим статистическим характеристикам от естественных генераторов случайных чисел (ГСЧ). Поэтому задача улучшения качества генераторов ПСЧ и поиска новых алгоритмов формирования ПСЧ является актуальной и стимулирует исследования путей улучшения статистических свойств псевдослучайных последовательностей чисел.

### Постановка проблемы

Различные методы генерации псевдослучайных чисел не могут быть применены для решения практических задач без исследования их статистических свойств и сравнительных количественных оценок. Для исследования статистических свойств используют различные статистические тесты, объединенные в пакеты тестирования. Прохождение статистических тестов позволяет с высокой вероятностью говорить о высоком качестве исследуемой псевдослучайной последовательности чисел.

Целью настоящей работы является оценка качества последовательности псевдослучайных чисел, порожденных комбинационным генератором, с помощью статистических пакетов тестирования, а также некоторых непараметрических критериев.

#### **Постановка задачи**

Комбинационные генераторы, подробно рассмотренные в [1, с. 45-49] и [2, с. 283-290], основываются на комбинации нескольких исходных генераторов. Такие генераторы позволяют улучшить статистические свойства последовательностей на выходе существующих генераторов псевдослучайных чисел. Кроме того, комбинация генераторов позволяет увеличить период повторения последовательности. В данной работе будем рассматривать комбинационный генератор с различным количеством исходных генераторов (от 2 до 8), представляющих собой циклические сдвиговые регистры (таблицы) с записанными в них перестановками, сформированными с помощью линейного конгруэнтного метода [2, с. 275-277], аддитивного генератора [3] либо генератора случайных чисел (в данной работе используется квантовый ГСЧ [4]). Мощность алфавита  $M$  выбрана равной 256, что позволяет без дополнительных преобразований формировать бинарный файл из слов на выходе комбинационного генератора, пригодный для последующего использования статистическими пакетами тестирования.

Комбинирующей функцией рассматриваемого генератора является функция "сумма по модулю  $M$ ", применяемая к значениям последовательностей, порожденных группой независимых первичных генераторов. Задачей работы является исследование статистических характеристик последовательностей, порождаемых комбинационным генератором с комбинирующей функцией суммы по модулю  $M$  в зависимости от количества исходных таблиц перестановок и их заполнения.

#### **Решение задачи**

Алгоритм работы рассматриваемого в данной работе комбинационного генератора изложен в [5]. Его принцип работы состоит в следующем. Группа исходных генераторов работает синхронно (слова на выходе каждого из генераторов появляются одновременно). В исходные генераторы (таблицы перестановок) записаны перестановки на множествах с мощностями алфавитов  $M_i$  (последовательности чисел, которые равномерно распределены на отрезке  $[0, M_i - 1]$  с нулевой ошибкой воспроизведения [6]). Результаты работы каждого из генераторов одновременно подаются на вход сумматора по модулю  $M$ , выход которого и является результатом работы комбинационного генератора.

Предположим, что  $X_1, X_2, \dots, X_n$  – дискретные равномерно распределенные случайные величины в диапазоне  $[0..M-1]$ . Широко известно, что случайная величина  $Y = X_1 + X_2 + \dots + X_n/M$  – также равномерно распределена в диапазоне  $[0..M-1]$ . Такое утверждение справедливо также для бесконечных последовательностей значений случайных величин  $X_1, X_2, \dots, X_n$ .

Если исходные последовательности  $X_1, X_2, \dots, X_n$  представляют собой циклически повторяющиеся перестановки на множествах с мощностями алфавитов  $M_i$ , следует воспользоваться результатами исследования, изложенного в [7]. Пусть каждый из двух исходных генераторов циклически формирует некоторую перестановку на множествах целых чисел из диапазонов  $[0, M_1 - 1]$  и  $[0, M_2 - 1]$  для первого и второго генератора, соответственно. Тогда, как указано в [7], для равномерного распределения дискретной случайной величины на множестве целых чисел мощности  $M$  на выходе комбинационного генератора, выполняющего операцию суммирования по модулю  $M$  слов от двух исходных генераторов, достаточно, чтобы  $M_1$  и  $M_2$  были взаимно просты и одно из значений  $M_1$  или  $M_2$  было кратно  $M$ .

Указанное утверждение с помощью метода индукции можно расширить для комбинационного генератора, состоящего из  $n$  исходных генераторов. В результате получим утверждение: для равномерного распределения дискретной случайной величины на множестве целых чисел мощности  $M$  на выходе комбинационного генератора, выполняющего операцию суммирования по модулю  $M$  слов от  $n$  исходных генераторов, где  $i$ -ый исходный генератор

циклически формирует некоторую перестановку на множестве целых чисел из диапазона  $[0, M_i - 1]$ ,  $i \in [1, n]$ , достаточно, чтобы все значения  $M_i$  были попарно взаимно просты  $\left( \text{НОД}(M_i, M_j) \Big|_{i \neq j} = 1 \right)$  и одно из значений  $M_i$  было кратно  $M$ .

Следует учитывать, что период комбинационного генератора будет равен наименьшему общему кратному периодов исходных генераторов. При использовании исходных генераторов (таблиц перестановок) с взаимно простыми периодами повторения можно достичь максимального периода последовательности на выходе комбинационного генератора  $T_{\max} = \prod_i M_i$ . Комбинационный генератор позволяет использовать существующие методы формирования псевдослучайных чисел, такие как линейный конгруэнтный метод или метод, основанный на использовании регистра сдвига с обратными связями, без их дополнительной модернизации.

В целях ускорения работы и упрощения конструкции рассматриваемого комбинационного генератора будем рассматривать предварительно подготовленные таблицы перестановок (исходные таблицы), сформированные с помощью генераторов псевдослучайных либо случайных последовательностей.

#### **Заполнение исходных таблиц перестановок комбинационного генератора**

Существуют два принципиально разных типа генераторов случайных и псевдослучайных чисел. Для пояснения рассмотрим аналогию с лототроном, в который засыпаны пронумерованные от 0 до  $M-1$  шары.

Если последовательно извлекать перемешанные шары из урны, не возвращая шар в урну, получим генератор типа "без возврата". Вероятность выпадения каждого шара для этого типа генераторов зависит от порядкового номера вытянутого шара и равна  $p_i = \frac{1}{M-i}$ , где  $i$  – порядковый номер шара,  $i \in [0, M-1]$ . После опустошения урна снова засыпается шарами, они перемешиваются, а приведенный алгоритм повторяется.

Тип генератора "с возвратом" отличается от описанного выше тем, что извлеченный шар после считывания его численного значения, опускается в урну и шары в урне повторно перемешиваются. Вероятность выпадения шара в таком случае не зависит от порядкового номера шара и равна  $p_i = \frac{1}{M}$ . Для формирования исходных таблиц комбинационного генератора при использовании генераторов случайных чисел возникает проблема повторения слов в таблицах.

Поскольку генератор случайных чисел работает по схеме "с возвратом", а  $p_i = \frac{1}{M}$ , его использование в исходном виде для заполнения таблиц в комбинационном генераторе не представляется возможным. Рассмотрим алгоритм построения исходных таблиц для комбинационного генератора с помощью генераторов (псевдо)случайных чисел, функционирующих по схеме "с возвратом", т. е. алгоритм преобразования генератора "с возвратом" в генератор "без возврата".

Сформируем две пустые таблицы  $A$  и  $B$  размером  $M$ , где  $M$  – мощность алфавита требуемой таблицы. Таблица  $A$  – временная таблица, таблица  $B$  – таблица-результат. Заполним таблицу  $A$  последовательно значениями от 0 до  $M-1$ . Генератор типа "с возвратом" (далее генератор) настроим на формирование числа в диапазоне от 0 до  $M-1$ . Значение, полученное от генератора, является указателем на ячейку таблицы  $A$ . Значение этой ячейки записывается в таблицу  $B$  на нулевую позицию. Из таблицы  $A$  ячейка удаляется со сдвигом всех последующих значений вверх и уменьшением размера таблицы на единицу. Далее генератор настраивается на формирование числа в диапазоне от 0 до  $M-2$  (где  $M-2$  фактически является новым размером таблицы  $A$ ). Полученное значение аналогично используется как указатель в таблице  $A$  с последующим перемещением ячейки таблицы  $A$  на следующую позицию таблицы  $B$ . После  $M$  итераций описанного алгоритма в таблице  $B$  находятся значения от 0 до  $M-1$  без повторов и пропусков, перемешанные соответственно использованному генератору.

#### **Статистические свойства последовательности, порожденной комбинационным генератором**



Для определения статистических свойств случайных и псевдослучайных последовательностей чисел применяются различные тесты, из которых можно выделить два класса: статистические и графические [8]. Статистические тесты позволяют получить численную оценку качества исследуемой последовательности. Результат графического теста представляется в виде графика и не имеет численного значения, является субъективным и не может быть объективно оценен. Существует множество пакетов статистического тестирования последовательностей чисел, такие как: тесты NIST, DIEHARD, TEST-U01, CRYPT-X, The pLab Project, Dieharder, ENT и др. В данной работе используется пакет тестирования NIST [9] как наиболее подробно отражающий статистические характеристики исследуемых последовательностей.

Дополнительно проведем оценку последовательности на выходе исследуемого генератора с помощью непараметрических критериев, таких как критерий знаков [10, с. 254-260, 11, с. 89-91] и критерий серий [11, с.91-93]. Заметим, что непараметрические критерии используют не численные значения выборки, а ее структурные свойства, что позволяет провести оценку последовательности независимо от предполагаемого закона распределения. Так, критерий знаков позволяет определить однородность двух рассматриваемых выборок. Однородными называются выборки, имеющие равные функции распределения. В данной работе будем рассматривать сравнение различных реализаций комбинационного генератора со случайной последовательностью чисел. Критерий серий позволяет определить, что слова на выходе исследуемого генератора являются случайными и независимыми.

### Результаты анализа последовательности с помощью пакета тестирования NIST

Для определения статистических характеристик использовалась выборка размером  $2^{25}$  байт. Каждый из 188 тестов пакета производится  $N$  раз над последовательностью длиной  $V$  бит. В данной работе, в соответствии с рекомендациями NIST [9],  $N=100$ ,  $V=1000000$ . В качестве нулевой гипотезы  $H_0$  принималась гипотеза о том, что проверяемая последовательность является случайной. Результатом работы каждого теста является  $N$  значений, называемых  $p$ -value.  $P$ -value есть вероятность ошибки при отклонении нулевой гипотезы о случайности последовательности (ошибка первого рода).  $N$  результатов тестов разделяются по значению на 10 групп:  $[0;0,1)$ ,  $[0,1;0,2)$ , ...,  $[0,9;1]$ . Распределение по этим группам должно быть равномерным. Подсчитанное количество значений в каждой из групп оцениваются с помощью критерия хи-квадрат (теоретическое значение  $N/10$ ). Полученная оценка и является результатом каждого из тестов пакета. Результаты тестирования по этому параметру приведены в таблице 1.

Таблица 1

Количество тестов, не удовлетворяющих доверительной вероятности для теста NIST

Заполнение таблиц	Количество исходных таблиц						
	2	3	4	5	6	7	8
Линейный конгруэнтный метод	148/188	1/188	0/188	0/188	0/188	0/188	0/188
Аддитивный генератор	147/188	0/188	0/188	0/188	0/188	0/188	0/188
Квантовый генератор случайных чисел	149/188	0/188	0/188	0/188	0/188	0/188	0/188

Дополнительно производится подсчет полученных значений  $p$ -value, не удовлетворяющих доверительной вероятности. В данной работе доверительная вероятность равна  $\alpha=0,01$ . Отношение количества значений  $p$ -value, которые больше доверительной вероятности, к общему количеству результатов теста отображается в процентном соотношении. В соответствии с рекомендациями NIST [9], в данной работе используется ограничение в 96% – тесты с меньшим значением считаются не пройденными. Результаты тестирования по этому параметру приведены в таблице 2.

Таблица 2

Количество тестов, в которых тестирование прошло меньше 96% последовательностей для теста NIST

Заполнение таблиц	Количество исходных таблиц						
	2	3	4	5	6	7	8

Линейный конгруэнтный метод	11/188	1/188	2/188	1/188	3/188	1/188	3/188
Аддитивный генератор	17/188	3/188	2/188	0/188	0/188	0/188	0/188
Квантовый генератор случайных чисел	20/188	1/188	1/188	0/188	0/188	0/188	0/188

Рассмотрим результаты статистического исследования последовательностей с помощью пакета NIST. Как следует из таблиц 1 и 2, при использовании двух исходных таблиц в независимости от метода их заполнения видно, что большая часть тестов не удовлетворяет доверительной вероятности (таблица 1), а также значительная часть тестов не преодолевает порог пройденных тестов в 96% (таблица 2). Причины таких результатов следующие. При использовании двух исходных таблиц в данной работе использовались таблицы размерами  $M_1 = 251$  и  $M_2 = 241$  для всех вариантов заполнения. При такой конфигурации комбинационного генератора период формируемой им последовательности равен  $T = M_1 * M_2 = 251 * 241 = 60491$  слов или 483928 бит. При этом, согласно рекомендациям, изложенным в [9], исследования проводились для последовательностей длиной 1000000 бит. В случае с двумя исходными таблицами происходит заикливание генератора, которое ведет к повторению последовательности после 483928 бит. Естественно такая последовательность не может считаться случайной, что и определяет данный статистический пакет тестирования. Поэтому следует всегда учитывать период повторения последовательности на выходе комбинационного генератора при его использовании.

При использовании трех и более исходных таблиц все варианты заполнения комбинационного генератора удовлетворяют требованиям доверительной вероятности (таблица 1). Однако дополнительная оценка по проценту последовательностей, прошедших тесты (таблица 2), показывает незначительные отклонения для последовательностей при заполнении исходных таблиц с помощью линейного конгруэнтного метода. При таких результатах тестирования не рекомендуется использовать подобную конструкцию комбинационного генератора в задачах, требующих высокого качества псевдослучайных последовательностей. Для других же вариантов заполнения исходных таблиц (при количестве таблиц 5 и более) последовательности полностью проходят все тесты пакета тестирования NIST, что свидетельствует о высоком качестве последовательности псевдослучайных чисел, получаемой на выходе комбинационного генератора.

#### **Результаты анализа последовательности с помощью непараметрических критериев знаков и серий**

Рассмотрим результаты исследования последовательностей, порожденных комбинационным генератором, с помощью критерия знаков. В качестве эталонной последовательности использована случайная последовательность чисел, полученная с помощью [12]. В качестве основной гипотезы  $H_0$  принимается утверждение, что исследуемые последовательности являются однородными, т.е. вероятности отклонения разности между словами исследуемой и случайной последовательностей в ту либо иную сторону равны между собой. Тогда  $H_0: p = 1/2$ . В качестве альтернативных выдвигаются гипотезы:

- гипотеза  $H_1: p > 1/2$  – вероятность отклонения разности между словами исследуемой и случайной последовательностей в положительную сторону больше, чем в отрицательную;
- гипотеза  $H_2: p < 1/2$  – вероятность отклонения разности между словами исследуемой и случайной последовательностей в отрицательную сторону больше, чем в положительную.

Если результат  $F_v$  расчетного значения статистики критерия знаков меньше критического значения F-распределения Фишера ( $F_v < F_{(1-\alpha)}(k_1, k_2)$ ) – тест считается пройденным. Для данного критерия в работе используется уровень значимости  $\alpha = 0,05$ . В соответствии с рекомендациями, изложенными в [10, с. 257-258], число испытаний для применения критерия знаков должно быть достаточно велико для значений  $p$ , близких к  $1/2$  и конкурирующих с  $p = 1/2$ . Так, например, для того, чтобы критерий знаков в 95% случаев отбрасывал гипотезу  $H_0: p = 1/2$ , когда на самом деле  $p = 0,45$  с уровнем значимости  $q = 5\%$ , необходимо произвести не менее 1297 наблюдений [10, с. 258]. Минимальный требуемый объем выборки монотонно увеличивается при приближении значения  $p$  к  $1/2$  и при уменьшении уровня значимости  $q$ . Исходя из этого, в качестве исходных данных для применения критерия знаков в настоящей работе рассматривались первые 4096 слов каждой из последовательностей комбинационного генератора в зависимости от различного количества исходных таблиц и различного их заполнения. Результаты исследования приведены в таблицах 3, 4, 5.

Таблица 3

Результаты теста критерия знаков для заполнения исходных таблиц с помощью аддитивного генератора

Количество исходных таблиц	$H_1 : p > 1/2$				$H_2 : p < 1/2$			
	$k_1$	$k_2$	$F_B$	$F_{(1-\alpha)}(k_1, k_2)$	$k_1$	$k_2$	$F_B$	$F_{(1-\alpha)}(k_1, k_2)$
2	4004	4158	1,03846	1,052845	4160	4002	0,96202	1,052877
3	4120	4030	0,97816	1,052903	4032	4118	1,02133	1,052885
4	4048	4118	1,01729	1,052833	4120	4046	0,98204	1,052847
5	4064	4092	1,00689	1,052868	4094	4062	0,99218	1,052874
6	4086	4072	0,99657	1,052866	4074	4084	1,00245	1,052863
7	4150	4006	0,9653	1,052894	4008	4148	1,03493	1,052865
8	4170	4004	0,96019	1,052839	4006	4168	1,04044	1,052806

Таблица 4

Результаты теста критерия знаков для заполнения исходных таблиц с помощью линейного конгруэнтного метода

Количество исходных таблиц	$H_1 : p > 1/2$				$H_2 : p < 1/2$			
	$k_1$	$k_2$	$F_B$	$F_{(1-\alpha)}(k_1, k_2)$	$k_1$	$k_2$	$F_B$	$F_{(1-\alpha)}(k_1, k_2)$
2	4032	4126	1,02331	1,052858	4128	4030	0,97626	1,052878
3	4144	4012	0,96815	1,052891	4014	4142	1,03189	1,052865
4	4042	4110	1,01682	1,052879	4112	4040	0,98249	1,052893
5	4104	4058	0,98879	1,052856	4060	4102	1,01034	1,052847
6	4106	4050	0,98636	1,052878	4052	4104	1,01283	1,052867
7	4136	4018	0,97147	1,052895	4020	4134	1,02836	1,052871
8	4136	4032	0,97485	1,052846	4034	4134	1,02479	1,052825

Таблица 5

Результаты теста критерия знаков для заполнения исходных таблиц с помощью квантового генератора случайных чисел

Количество исходных таблиц	$H_1 : p > 1/2$				$H_2 : p < 1/2$			
	$k_1$	$k_2$	$F_B$	$F_{(1-\alpha)}(k_1, k_2)$	$k_1$	$k_2$	$F_B$	$F_{(1-\alpha)}(k_1, k_2)$
2	4178	3990	0,955	1,052864	3992	4176	1,04609	1,052826
3	4112	4060	0,98735	1,052824	4062	4110	1,01182	1,052814
4	4030	4108	1,01935	1,052925	4110	4028	0,98005	1,052942
5	4144	4010	0,96766	1,052898	4012	4142	1,0324	1,052871
6	4120	4044	0,98155	1,052854	4046	4118	1,0178	1,052839
7	4150	4006	0,9653	1,052894	4008	4148	1,03493	1,052865
8	4000	4166	1,0415	1,052832	4168	3998	0,95921	1,052866

Результаты исследование с помощью критерия знаков показывают однородность последовательности, полученной с помощью комбинационного генератора (при исследованных вариантах заполнения исходных таблиц), и последовательности случайных чисел. Из этого можно сделать вывод, что функции распределения исследуемых случайных величин равны. Результаты исследования последовательностей, сформированных комбинационным генератором, с помощью критерия серий приведены в таблице 6. Исследование проводилось для различного количества и заполнения исходных таблиц генератора. Рассматривались первые 256 слов каждой из последовательностей. Успешным прохождением теста является нахождение результата расчета статистики критерия ( $z_B$ ) в пределах критических значений. Примем уровень значимости  $\alpha = 0,01$ . Для такого  $\alpha$  критические значения  $z_B$ :  $-2,576 < z_B < 2,576$ .

Таблица 6

Результаты теста критерия серий

Количество исходных таблиц	$z_B$	$z_B$	$z_B$
	(заполнение исходных таблиц с помощью аддитивного)	(заполнение исходных таблиц с помощью линейного)	(заполнение исходных таблиц с помощью квантового)

	генератора)	конгруэнтного метода)	генератора случайных чисел)
2	-0,43836	-0,56361	-0,56266
3	0,688852	2,191802	-2,0657
4	0,698043	0,815126	-0,05872
5	-2,06656	0,188859	-0,68042
6	0,313115	-0,8141	-0,06262
7	-0,93934	-0,1869	1,565573
8	-1,31418	-0,1869	-0,68885

Результаты применения критерия серий подтверждают гипотезу о случайности слов на выходе комбинационного генератора для всех вариантов количества и заполнения исходных таблиц.

### Выводы

Результаты проведенного в работе исследования позволяют сформулировать следующие выводы:

- для последовательностей псевдослучайных чисел, порождаемых комбинационным генератором, успешное прохождение пакета тестирования NIST наблюдается для следующих конфигураций генератора: количество исходных генераторов (таблиц перестановок) – 5 и более; заполнение исходных таблиц – аддитивный генератор, квантовый ГСЧ;
- использование линейного конгруэнтного метода для заполнения исходных таблиц перестановок комбинационного генератора не рекомендуется для задач, требующих высокого качества последовательностей псевдослучайных чисел;
- успешный результат применения непараметрических критериев, таких как критерий знаков и критерий серий, подтверждает однородность исследуемых последовательностей со случайной последовательностью чисел (критерий знаков) и случайность слов на выходе комбинационного генератора (критерий серий).

1. Кнут Д. Э. Искусство программирования. Том 2. Получисленные алгоритмы / Дональд Э. Кнут. – М.: Вильямс, 2007. – 832с.
2. Шнайфер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си; [пер. с англ. под ред. Семьянова П.В.]. – [2-е изд.]. – М.: Триумф, 2002. – 816 с.
3. Random Class [Электронный ресурс]. – Режим доступа: <http://msdn.microsoft.com/library/system.random%28v=vs.110%29.aspx>.
4. QRNG Service [Электронный ресурс] – Режим доступа: <http://qrng.physik.hu-berlin.de/>.
5. Лавданский А.А. Комбинационный метод формирования последовательности псевдослучайных чисел / А.А. Лавданский, Э.В. Фауре // Системний аналіз та інформаційні технології: матеріали 16-ї Міжнародної науково-технічної конференції SAIT-2014, Київ, 26-30 травня 2014р. / ННК «ІПСА» НТУУ «КПІ». – К.: ННК «ІПСА» НТУУ «КПІ», 2014. – С. 403-404.
6. Фауре Э.В. Оценка точности воспроизведения закона распределения дискретной случайной величины при ее преобразовании / Э.В. Фауре, А.С. Береза, Е.А. Ярославская // Вісник Хмельницького національного університету. – 2012. – №5. – С. 176-182.
7. Фауре Э.В. Закон распределения дискретной случайной величины на выходе комбинационного генератора / Э.В. Фауре // Безпека інформації. – 2014. – №2. – С. 153-158. [Электронный ресурс] – Режим доступа: <http://jrn1.nau.edu.ua/index.php/Infosecurity/article/view/7301/8195>.
8. Иванов М.А. Теория, применение и оценка качества генераторов псевдослучайных последовательностей / М.А. Иванов, И.В. Чугунков – М.: КУДИЦ-ОБРАЗ, 2003. – 240с.
9. Random Number Generation and Testing [Электронный ресурс]. – Режим доступа: <http://csrc.nist.gov/groups/ST/toolkit/rng/index.html>.
10. Смирнов Н. В. Курс теории вероятностей и математической статистики для технических приложений / Н. В. Смирнов, И. В. Дунин-Барковский – М.: Наука. Главная редакция физико-математической литературы, 1969. – 512 с.
11. Большев Л. Н. Таблицы математической статистики / Л. Н. Большев, Н. В. Смирнов – [3-е изд.] – М.: Наука. Главная редакция физико-математической литературы, 1983. – 416 с.
12. True Random Number Service [Электронный ресурс] – Режим доступа: <http://random.org/>.

УДК 681.515.8

Zdolbitska N.V., Zdolbitsky A.P., Chmil O.M.  
Lutsk national technical university

## LIBRARY DEVELOPMENT FOR CHARACTER REAL-TIME ANIMATION

**Zdolbitska N.V., Zdolbitsky A.P., Chmil O.M. Library development for character real-time animation.** Increasing quantity and quality of games and other applications made popular using the detailed characters. The article describes skeleton and library for character that has realistic looking and skeleton structure. Most of the elements that are included to library are found in modern character animation libraries. To make unique characteristic of the character additionally was made new methods of animation and control. Moreover, animation technique of importing one model format to another is described.

**Key words:** skeleton, library, animation, effect, rigid bones, model.

**Figures 5, Bibliography 11**

**Здолбіцька Н.В., Здолбіцький А.П., Чміль О.М. Розробка бібліотеки реального часу для анімації персонажа.** Збільшення кількості та якості ігор та інших програм популяризувало використання деталізованих персонажів. Стаття описує скелет та бібліотеку, що створені для персонажу, який має реалістичний вигляд та скелетну структуру. Більшість елементів бібліотеки знайдено у сучасних анімаційних бібліотеках. Додатково були розроблені нові методи контролю та анімації для додавання унікальних рис персонажу. Також описано техніку імпортування одного формату моделі в інший.

**Ключові слова:** скелет, бібліотека, анімація, ефект, жорсткі кістки, модель.

**Здолбицкая Н.В., Здолбицкий А.П., Чмиль О.М. Разработка библиотеки реального времени для анимации персонажа.** Увеличение количества и качества игр и других программ популяризировало использование детализированных персонажей. Статья описывает скелет и библиотеку, созданных для персонажа, который имеет реалистичный вид и скелетную структуру. Большинство элементов библиотеки найдено в современных анимационных библиотеках. Дополнительно были разработаны новые методы контроля и анимации для добавления уникальных черт персонажа. Также описана техника импортирования одного формата модели в другой.

**Ключевые слова:** скелет, библиотека, анимация, эффект, жесткие кости, модель.

**Research Analysis/ previous works.** There are open and commercial systems that implement some form of character animation. Open systems include few libraries that have functions close to the developed one. Cal3d is only one stand-alone library and Nebula Device 2 and Irrlich are most popular through game developers.

Cal3d is an open source character animation library that is written in C++. It is skeletal based 3D character animation library that built in platform-graphic API-independent way. The library does not belong to a certain engine. Character basic concepts of animation are included in this library, but advanced features are not available. It uses an original file format and runs others format that are converted.

The Irricht engine is fast and supports skeleton and mesh based animation. There is no possibility to use advanced effects in it. The main focus is on character features, but not animation.

Advanced graphic engine Nebula Device 2 has good animation system and a high performance. It supports most character animation features. To perform skinning it uses palette-skinning vertex shader.

The main focus of new systems is on skeleton animation. Unfortunately, they do not put attention to mesh animation. Flexible library need to have methods that can control and extend skeletal animation, these include inverse kinematics and rigid bone attachment. Exactly this direction is chosen to make Skeleton Library.

**Development of the Skeleton Library.** Developed library includes features that have to be in the modern animation library, there are also some improvements in it. Furthermore this library is independent, so it can be added to any engine. It includes the following features:

- Model's control skeleton animation
- Support of existing and new formats
- Weighted skinning
- Seamless animation blending
- Advanced function controlled effects for any part of the model
- Skeleton fitting gives a possibility to use animation from other model formats
- Possibility to add library in any project

**Approach to design.** Library design is originated from BVH format. This format lacks a mesh structure. But library that has been developed took it to account. This is resulted in library including structures for a skeleton, mesh, keyframe animation and blender. Here the example of the listing:

```

HIERARCHY
ROOT Hips
{
  OFFSET -0.347901 96.7718 1.79791
  CHANNELS 6 Xposition Yposition Zposition Zrotation Xrotation Yrotation
  JOINT LeftUpLeg
  {
    OFFSET 8.91 -6.27 -2.145
    CHANNELS 3 Zrotation Xrotation Yrotation
    JOINT LeftUpLegRoll
    {
      OFFSET 0 -22.7323 0
      CHANNELS 3 Zrotation Xrotation Yrotation
    }
  }
  ...
}
    
```

Supporting of new formats is gained by parsers that are integrated to developed library. So adding a new format became really easy. Popular feature extensions like inverse kinematics and rigid bone attachment can be done through using the function controlled effects.

The BVH is the format that stores motion data. It is modeling format that has hierarchical structure of joints and effectors.

**Library features**

*Model* contains skeleton structure and mesh that defines the character's look and poses. It includes a number of animations and effects based on function. They determine movements that character is capable of. Another important thing is that model must have skeleton structure; the reason for it mesh animation is not supported. The model structure is depicted in Figure 1. All improvements are presented in Figure 2.

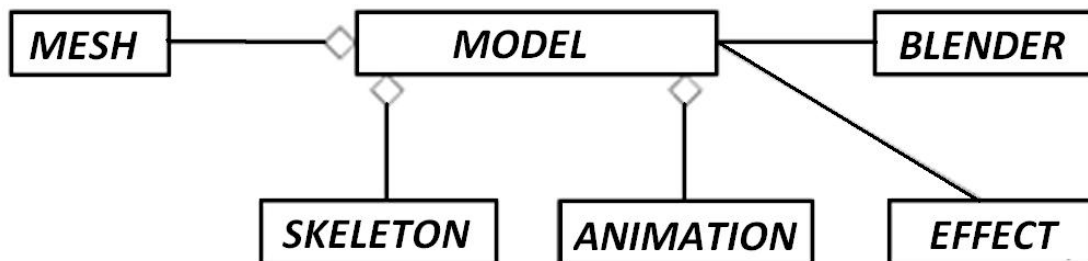


Fig. 1 – Model structure

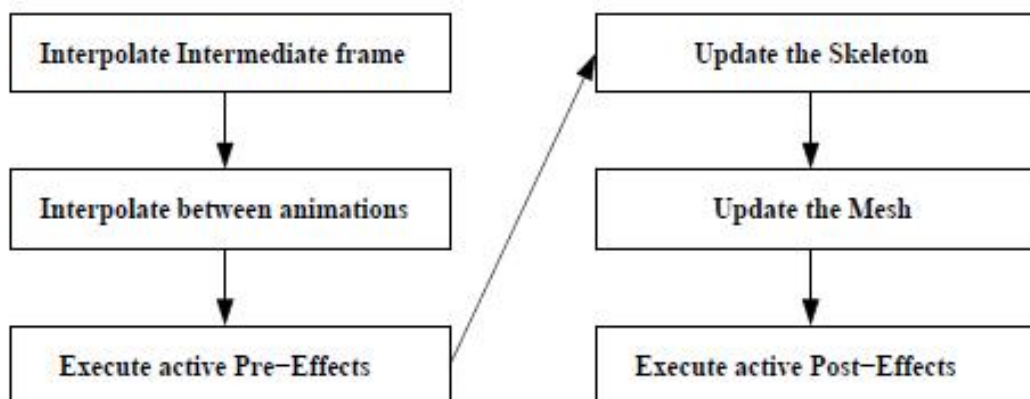


Fig. 2 – Module improvements

*Skeleton structure* consists of a number of joints, each one connected with parent hanks to lengths offset. Also joints have a rotation that is used for character movements. In addition the end effectors are used. An end effector is a rigid extension from the joint which cannot rotate. This gives an opportunity to use external tools to the body. The basic relative rotation to its parent is kept by the end effectors. The main root joint is the base of structure. Its offset is pointing the translation of the model. Next Figure describes the structure of the skeleton.

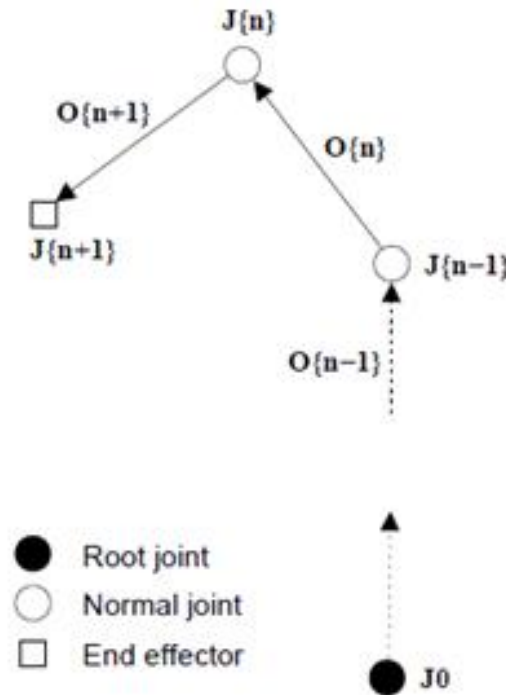


Fig. 3 – Skeleton structure

**Mesh.** The mesh possesses all main properties that include vertex normals, material properties, faces and texture coordinates. Weighted vertices are also supported. Each vertex connects to several bones and that is used to make the transitions between submeshes. There are can be any number of weights per vertex, but in some cases more than four weights are demanded. The mesh is divided to submeshes.

**Animation.** Here is used keyframe animation that means an animation includes few keyframes each consist of one rotation matrix and one offset vector for one bone. Every animation has a timer that helps to track current keyframe position. The animation also can be looped that means that character repeats one movement. There is also a possibility to apply many animations at the same time (Blender can be used for it).

**Blender and Progressive Skeleton Fitting.** Bender takes care of two main steps in the animation process: interpolation between keyframes of one animation and interpolation of all active animations by interpolating these together to one picture. To do so Blender uses quaternions and use following to spherical interpolation. Quaternions are weighted by user. It need to be remembered that only whole model can be blended.

OSL\_PARSER is performing skeleton fitting. It takes two models, source and target, and joint map to define which joints have to be matched. Thanks to this function fitted animation can be added. Nowadays, each set of model format demands separate map.

**Effects are controlled by functions.** All the effects implemented in subclass OSL\_Effect. Inside the subclass there is a function execute (OSL\_Model\*) that adds all updates of the model. This function is integrated with a model and has access to all private members. Pre-effect and Post-effect can be added to the model. They are made in the same way, but using Post-Effects demands extra care. The reason is that the Skeleton and Mesh are already done and all updates should be done correctly.

The effects will typically execute every update cycle of the model either prior to update or as a post-effect. The use of the post effects has the advantage in calculating the positions for multiple models only needing to do one update run through the entire model and adding the individual effects afterwards, skipping a lot of unnecessary recalculations. Of course it must be taken into consideration that changing a bone in a skeleton structure after its update means that a recalculation of its children must take place. The Pre-Effects are simply easier to create since no concern for specific update issues must be addressed.

In the case of Post-Effects, care must be taken to make sure that everything is updated correctly. For instance if the Effect influences a bone, it must make the effect of this alternation propagate to its children (if this is desired).

Additional improvements:

- Inverse Kinematics
- Animation bits - blending control
- Level-on-detail
- Vertex shader driven skinning
- Integrated physics

**Skeleton Fitting.** The following example illustrates the use of skeleton fitted animations. The application shows the typical usage where a very good animation for a skeleton with few bones exists but the animation is to be used on an advanced model with a detailed skeleton with a mesh. In this demo, the MD5 model fatty from Doom 3 is chosen as target model for animation. The source animation is taken from the the BVH model format, used for motion capture data. Target and source model in this scenario consists of 80 and 19 joints, respectively. The two models can be seen in Figure 4.

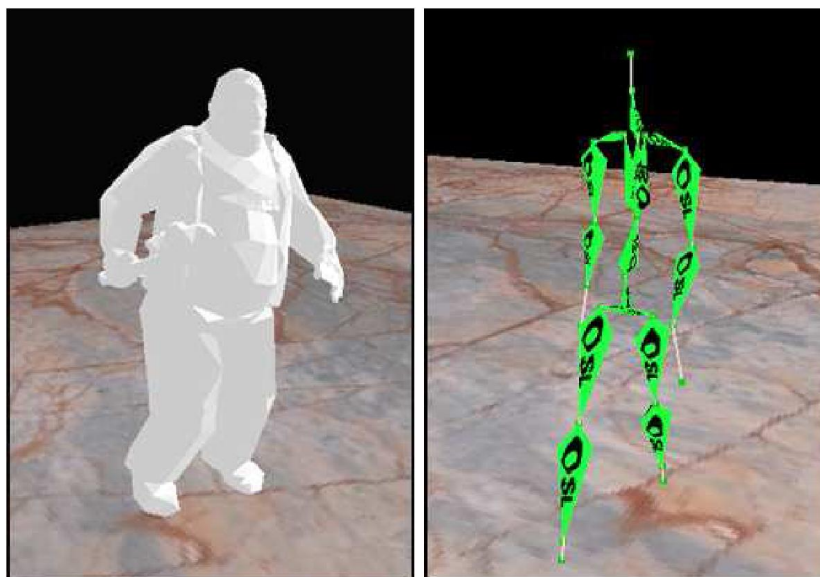


Fig. 4 – Target and source model

The process of transforming the animation from the source to the target starts with loading the different models with their corresponding models:

```
OSL_Parser_MD5 md5_p;  
OSL_Parser_BVH bvh_p;  
OSL_Model md5Model = md5_p.parseModel(“zfat/ zfat.md5mesh”);  
OSL_Model bvhModel = bvh_p.parseModel(“bvh/ ballet.bvh”);
```

To match the bones correctly, a map must be required. It contains an index at every position pointing to the matching joint index of the source model. If no matching joint is available, the default value will be -1.

```
std::vector<int> map;  
for (int m = 0; m<md5.getSkeleton() ->getNbrOfBones(); m++)  
{  
std::string name =  
md5model . getSkeleton() ->getBone (m) ->getName());
```



```
int bvhbone;  
if (name == "Body") bvhbone = BVH_HIPS;  
else if (name == "Lupleg") bvhbone = BVH_LEFTHIP;  
else if (name == "Rupleg") bvhbone = BVH_RIGHTHIP  
.br/>.br/>.br/>else bvhbone = -1;  
}  
map.pus_back(bvhbone);
```

This map is then used to apply the animation to the target model.

```
OSL_Parser::fitAnimation(md5model, bvhModel, 0, map);
```

This line of code adds the animation starting at index 0 from the bvhModel to md5model. The fitting of the bindpose is made by matching the target's bindpose to the source's bindpose which can be seen in Figure 5, from left to right, the default bindpose of target, the source's bindpose and finally the target's fitted bindpose. The fitting is now complete and the animation is loaded and usable by the md5model.

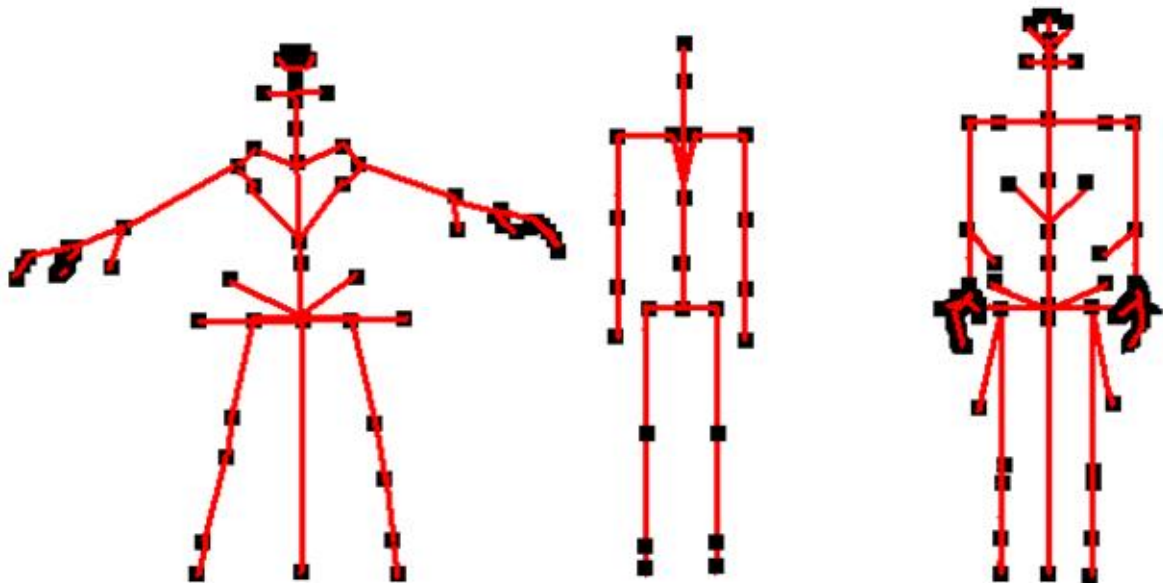


Fig. 5 – The steps of skeleton fitting

**Rigid Bone Attachment.** To illustrate how effective the function controlled effects are, they have been used to implement rigid bone attachments for OSL. The attachment is implemented through a Post-Effect, taking an OSL\_Model as the attachment, and connecting it to a designated bone. It is then added to a target model where its execute function replaces the attached model's root offset and rotation found at the target's designated bone. Since it is a Post-Effect it will execute after the target model has performed its updates, thus, putting it in the right place. The trick is to communicate with the model update through the use of its matrix and vector stores.

**Conclusions and Results.** The Open Library has a sound design capable of including many features. All implemented animation techniques in the library are smoothly integrated with each other and can be used in a smooth way. The performance of the library can be improved by implementing vertex shader driven skinning.

The new features, skeleton fitting and function controlled effects, were successfully implemented. Through the Effects the same model can be used repeatedly with individual characteristics, preventing the appearance of the cloning effect often seen in games. The Effects can also be used to easily add new arbitrary methods of control. The skeleton fitting is an excellent tool for applying animations from one model to another. This is especially useful when motion capture data, which is obviously done for a human skeleton, should be applied to a non-human model.

Although not complete the Open Skeleton library has the potential to be one of the leading open source character animation libraries. A clear design and extensible structure makes it ideal as a modular extension in, for instance, a game engine.

1. Здолбіцька Н.В., Здолбіцький А.П., Яручик В.Л. Крокуюча система на Bluetooth управлінні // Міжвузівський збірник "Комп'ютерно-інтегровані технології: освіта, наука, виробництво" – Луцьк: Видавництво ЛНТУ. – Вип. 16-17. – 2014. – С. 27–31.
2. Здолбіцька Н.В., Савлук А.В. Моделювання поведінки живих організмів за допомогою аніматів на базі роботизованих платформ // Міжвузівський збірник "Комп'ютерно-інтегровані технології: освіта, наука, виробництво" – Луцьк: Видавництво ЛНТУ. – Вип. 12. – 2013. – С. 27-30.
3. Шолом П.С., Здолбіцька Н.В. Пропорційно-інтегрально-диференціальне управління стабільністю руху мобільного агента Зрі по заданій траєкторії / П.С. Шолом, Н.В. Здолбіцька // Міжвузівський збірник "Комп'ютерно-інтегровані технології: освіта, наука, виробництво" – Луцьк: Видавництво ЛНТУ. – Вип. 8. – 2012. – С. 125–129.
4. Bio Vision Incorporated <http://www.biovision.com>
5. Blundell, Barry G. & Schwarz, Adam J. "Creative 3-D display and interaction interfaces: a transdisciplinary approach". Hoboken. Wiley-Interscience, 2006
6. Cheng, Irene... [et al.], (eds.). "3D online multimedia & games: processing, transmission and visualization". Hackensack, N.J. World Scientific Publishing, 2008
7. Mikio L. Braun. Accurate error bounds for the eigenvalues of the kernel matrix. J. Mach. Learn. Res., 7:2303–2328, 2006.
8. Pablo de Heras Циєчомські Parametric Dynamics - A method for addition of dynamic motion to computer animated human characters Lund University of Technology 2001.
9. Pauli, David B., Programming Dynamic Character Animation, 2002 Charles River Media, inc, pages 164-165.
10. Sebastien Domine Mesh Skinning nVidia 2005.
11. Watt, Alan and Policarpo Fabio, 3D Games - Animation and Advanced Real-time Rendering Vol. 2, 2003 Pearson Educational Lint, pages 358-362.

УДК 519.718: 371.261:336:713 (045)

Яремко С.А.

Вінницький торговельно-економічний інститут

Бевз С.В.

Вінницький національний технічний університет

## ПОБУДОВА МОДЕЛІ РЕЙТИНГОВОГО ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ ДІЯЛЬНОСТІ КОМЕРЦІЙНИХ БАНКІВ

**Яремко С.А., Бевз С.В. Побудова моделі рейтингового оцінювання результатів діяльності комерційних банків.** У статті проведено аналітичний огляд методів та моделей, які використовуються для проведення рейтингового оцінювання комерційних банків. Це дало змогу здійснити їх класифікацію, провести порівняльний аналіз параметрів та визначити шляхи удосконалення рейтингового оцінювання комерційних банків на основі використання розробленої регресійної моделі.

**Ключові слова:** рейтинг, модель рейтингового оцінювання банківської діяльності, параметри моделі, інтегральний показник, регресійний аналіз.

**Форм. 13. Табл. 1. Рис. 1. Літ. 14**

**Яремко С.А., Бевз С.В. Построение модели рейтингового оценивания результатов деятельности коммерческих банков.** В статье проведен аналитический обзор методов и моделей, которые используются для проведения рейтингового оценивания коммерческих банков. Это дало возможность осуществить их классификацию, провести сравнительный анализ параметров и определить пути совершенствования рейтингового оценивания коммерческих банков на основании использования разработанной регрессионной модели.

**Ключевые слова:** рейтинг, модель рейтингового оценивания банковской деятельности, параметры модели, интегральный показатель, регрессионный анализ.

**Форм. 13. Табл. 1. Рис. 1. Лит. 14**

**Yaremko S.A., Bevs S.V. Construction of model rating evaluation of results activity of commercial banks.** The state-of-the-art review of methods and models which are used for the leadthrough of rating evaluation of commercial banks is conducted in the article. It enabled to carry out their classification, conduct the comparative analysis of parameters and define the ways of improvement rating evaluation of commercial banks on the basis of the use of the developed regressive model.

**Keywords:** rating, model of rating evaluation of bank activity, model parameters, integral index, regressive analysis.

**Form. 13. Tabl. 1. Pict. 1. Lit. 14**

**Постановка проблеми.** В умовах ускладнення економічного та політичного становища в Україні особливої актуальності для громадян набуває питання вибору надійного банку, що має значний запас стійкості та в змозі вчасно розраховуватись по своїм зобов'язанням. Це обумовлює необхідність проведення оцінки фінансової діяльності банків як для клієнтів, що користуються їх послугами, так і для самих банків щоб дослідити зовнішнє середовище та свою конкурентоспроможність. Тому доцільним є проведення не тільки внутрішнього аналізу діяльності банку, але й порівняння його фінансових показників з такими ж параметрами інших банків на основі використання сучасних методів та моделей рейтингового оцінювання. Згідно загальноприйнятого розуміння, рейтинг є інструментом, що призначений для порівняльного аналізу динамічно розвинених систем. Рейтинг банків, зокрема, є певною системою оцінки їх діяльності, яка базується на фінансових показниках роботи та балансових даних банку [1, С.149].

Оцінювання банків здійснюється за певними ознаками, що дозволяє групувати їх в певній послідовності за ступенем зменшення (або збільшення) цих показників. Ознаки за якими оцінюють банки можуть відображати окремі напрями їх діяльності (якість активів, платоспроможність, рентабельність, ліквідність тощо) або ж діяльність банків в цілому (менеджмент, обсяг операцій, показники зростання, надійність). Сукупність оцінок банку за певними ознаками дозволяє сформувати узагальнюючий рейтинговий показник, який дає інтегровану оцінку установи [2, С.49]. Тому щоб забезпечити конкурентоспроможність перед керівництвом комерційних банків на даному етапі ставляться принципово нові вимоги, спрямовані на обґрунтоване та професійне управління, що базуватиметься на результатах відповідних аналітичних досліджень та розрахунків щодо рейтингового оцінювання поточного й майбутнього стану банку, його потенційних можливостей, сильних і слабких сторін тощо.

**Аналіз останніх досліджень і публікацій.** Методам та моделям рейтингового оцінювання показників діяльності комерційних банків присвячені наукові публікації таких вітчизняних та зарубіжних вчених як Домбровський В.С. [6], Каширіна О. [11], Кириченко О. [3],

Кромонов В.С. [7], Майба В.В. [2], Матвієнко В.П. [10], Пшик Б.І. [1], Турило А.М. [4], Фалюта А.В. [12], Ширинська Е.Б. [9], проте даний напрямок потребує подальшого розвитку з метою удосконалення оцінки стану фінансових установ відповідно до поточної економічної ситуації.

**Метою роботи** є здійснення аналітичного огляду сучасних методів та моделей, які використовуються для проведення рейтингового оцінювання комерційних банків; проведення порівняльного аналізу їх параметрів та визначення шляхів удосконалення рейтингового оцінювання комерційних банків.

**Виклад основного матеріалу дослідження.** Використання моделей рейтингового оцінювання насамперед обумовлене бажанням учасників фінансового ринку знати результати діяльності інших учасників – партнерів чи конкурентів аби орієнтуватися на них і визначати перспективні напрями свого розвитку. В Україні за допомогою рейтингів як правило проводиться оцінка надійності й ефективності банківських установ, а також оцінка їхньої кредитоспроможності. Особливо важливе місце належить рейтингу надійності банків. Оцінювання цього фінансового показника може визначити як відомство нагляду за діяльністю банків (в Україні – Національний банк України), так і окремі рейтингові агентства. Найбільш відомими рейтинговими агентствами в світі є компанії Moody's Investor Servis, Standart Poor's Rating Services, IBCA Ltd, Thomson Bank Watch [3]. Необхідно зазначити, що проведення оцінювання рейтингу банку міжнародними рейтинговими агентствами, в сучасних умовах є необхідною складовою для виходу фінансової установи на зовнішні ринки. Для проведення рейтингового оцінювання у вітчизняній та закордонній практиці застосовують різноманітні методи та моделі, залежно від мети, яку ставлять перед собою дослідники. Зрозуміло, що єдиної універсальної методики немає, оскільки кожний банк має свої специфічні особливості: функціональне призначення; стиль управління; комплекс операцій та послуг; склад клієнтів тощо. Усю сукупність методів та моделей оцінювання діяльності банків можна умовно класифікувати як на рис.1.



Рис. 1. Класифікація методів та моделей рейтингового оцінювання

Найбільш поширеними є номерні, бальні, рейтингові, індексні методи, проте поряд із ними має місце застосування економіко-математичних та інших методів і моделей. Бальні методи дають змогу отримати інтегральну оцінку фінансового стану банків у балах, що відображає сумарну сукупність оцінок за кожним обраним фінансовим показником. Під час використання бального методу кожному показнику присвоюється певний бал у шкалі, визначеній експертами. Загальна сума балів, отриманих банком за кожним із показників, дає підставу зараховувати банк до певної групи [4, С.261].

Класичним прикладом бального методу є рейтингова система CAMELS, яка була розроблена групою експертів США. Ця методика широко використовується у світовій практиці та в Україні. Абревіатура CAMELS походить від перших букв компонент, що перевіряються [5]:

- С (Capital adequacy) – достатність капіталу;
- А (Asset quality) – якість активів;
- М (Management) – якість управління;
- Е (Earnings) – прибутковність;
- L (Liquidity) – ліквідність;
- S (Sensitivity to risk) – чутливість до ризику.

Для реалізації методу CAMELS передбачається оцінювання кожного банку за зазначеними вище параметрами у балах на основі документів, які надходять до агентства банківського нагляду. Загальна оцінка розраховується як середнє арифметичне оцінок за кожною компонентою, яка перевіряється. Найкраща оцінка – 1, найгірша – 5.

Також за цим методом всі активи за ступенем ризику поділяються на чотири групи: I група – особливо згадані активи (ступінь ризику 1 %), II – нестандартні активи (ступінь ризику 20 %), III – сумнівні активи (ступінь ризику 50%) і IV – втрати (ступінь ризику 100%). Віднесення активів банку до вказаних груп ризику на основі проведеного аналізу дозволяє визначити рівень сукупного ризику активів банку. Слід зазначити, що рейтинг банку, визначений за системою CAMELS, є конфіденційною інформацією, призначеною тільки для внутрішнього використання Національним банком України та не підлягає опублікуванню у засобах масової інформації, тому доступ до цієї інформації інших учасників фінансового ринку є закритим. Для адаптації методики CAMELS до вітчизняної банківської системи Національний банк України додатково враховує такі параметри, як місткість фінансового ринку, рівень розвитку інфраструктури, ступінь стабільності банківської системи. Таким чином бальні методи характеризуються нескладною процедурою аналізу, наочністю результатів, але вимагають наявності усіх необхідних для оцінювання документів та не завжди є доступними для ознайомлення іншим учасникам фінансового ринку.

Індексний метод передбачає використання вагових коефіцієнтів для кожного з показників. Як правило, вибір вагових коефіцієнтів проводиться експертами на основі власного досвіду роботи, інтуїції та професійних знань [7].

На використанні індексів базується метод В.С. Кромонава, що дозволяє досить повно оцінити фінансово-економічні показники діяльності банків. Для аналізу використовуються дані балансу, на основі яких розраховуються абсолютні параметри капіталу, зобов'язань, активів. За допомогою співставлення вказаних параметрів визначаються параметричні коефіцієнти: генеральний коефіцієнт надійності  $K_1$ , коефіцієнт миттєвої ліквідності  $K_2$ , крос-коефіцієнт як співвідношення сукупних зобов'язань та ризикових активів  $K_3$ , генеральний коефіцієнт ліквідності  $K_4$ , коефіцієнт захищеності капіталу  $K_5$ , коефіцієнт фондової капіталізації прибутку як відношення власного капіталу до сформованого статутного фонду  $K_6$ . Далі розраховується поточний індекс надійності, для чого отримані коефіцієнти нормуються евристичним методом, перемножуються на емпіричну вагу і підсумовуються. Узагальнена формула для обчислення поточного індексу надійності за методом В.С. Кромонава має такий вигляд (1):

$$N = 45\% \times K_1 + 20\% \times K_2 + 10\% \times K_3 / 3 + 15\% \times K_4 + 5\% \times K_5 + 5\% \times K_6 / 3 \quad (1)$$

Усі показники у формулі (1) підібрані таким чином, щоб якомога повніше відобразити фінансово-економічну діяльність комерційного банку, а сама формула має адитивний вигляд, тобто чим більше значення має кожен з показників, тим більшим у підсумку виявляється рівень фінансової стійкості. При цьому кожен з показників фінансової стійкості нормується евристичним методом і множить на емпіричну вагу. Евристичний метод нормування в даному випадку полягає в тому, що значення показників, які характеризують діяльність банку, ділять на відповідні показники «ідеального банку», що має оптимальне співвідношення між надійністю і прибутковістю. Згідно з методом В.С. Кромонава, «ідеальним» вважається банк, який має такі показники:  $K_1=1$ ;  $K_2=1$ ;  $K_3=3$ ;  $K_4=1$ ;  $K_5=1$ ;  $K_6=3$ . Емпіричну вагу кожного показника визначено в результаті експертного дослідження його значущості ( $K_1=45\%$ ;  $K_2=20\%$ ;  $K_3=10\%$ ;  $K_4=15\%$ ;  $K_5=5\%$ ;  $K_6=5\%$ ) і підібрано так, щоб значення індексу надійності  $N$  коливалося в межах від 0% (абсолютно нестійкий банк) до 100% (ідеально стійкий банк).

Отже, з доступністю вхідної інформації та результатів проведення аналізу в цьому методі проблем немає, але є не менш важливий аспект – всебічність та об'єктивність рейтингової оцінки. З використовуваних для аналізу параметричних коефіцієнтів можна зробити висновок, що методика робить акцент на стійкості банківської установи, що не є об'єктивним універсальним критерієм конкурентоспроможності банків. А відсутність показників прибутковості та якості кредитного портфеля є найбільшим недоліком методики В. Кромонава.

Подібним за своєю суттю є метод EUROMONEY. Він був розроблений Асоціацією комерційних банків Латвії на базі методики розрахунку рейтингу ефективності банків

Euromoney із заміною якісних факторів на показники достатності капіталу, величини балансового капіталу та активів. Інтегральний показник для цього методу розраховується за виразом (2):

$$N = 15 \times ROE + 10 \times ROA + 10 \times CIN + 10 \times MIN + 15 \times NI + \\ + 15 \times CA + 10 \times ASSETS + 15 \times CAPITAL, \quad (2)$$

де ROE – прибутковість капіталу; ROA – прибутковість активів; CIN – співвідношення доходів та витрат; NIM – чиста процентна маржа; NI – чистий прибуток; CA – показник достатності капіталу; ASSETS – активи банку (валюта активу балансу); CAPITAL – балансовий капітал.

Наведені у формулі показники відображають залежність результатів, отриманих за методикою Euromoney від таких показників, як, зокрема, розмір капіталу банку та загальний розмір активів, вагові значення яких відповідно складають 15 та 10. А зважаючи на досить незначну кількість показників у даній моделі можна зробити висновок, що чим вище буде значення даних показників у оцінюваного банку, тим вищий у кінцевому результаті він отримає рейтинг. Така ситуація у будь-якому випадку може свідчити про залежність тільки від однієї групи параметрів та неврахування багатьох інших важливих факторів в такій рейтинговій методиці.

Слід також зазначити, що загальною проблемою ранжування банків за ступенем надійності є те, що в більшості випадків два банки неможливо порівняти в цілому, оскільки за деякими показниками один банк кращий за інший, в той же час порівняння за іншими показниками може виявити зворотну тенденцію. Ще одним фактором, що вказує на неможливість такого порівняння є і те, що по різних початкових характеристиках «якнайкращими» і «якнайгіршими» є різні банки [8, С.148]. Певним удосконаленням наведених вище методів за рахунок конструктивних змін є методика О.Б. Ширинської, що передбачає використання у системі рейтингових показників також і коефіцієнтів рентабельності (ROA, ROE) [9]. А для відсікання суб'єктів оцінки, окрім інших показників, запропоновано застосовувати частку прострочених позик у кредитному портфелі банку, що є суттєвим нововведенням. Так, банк, у кредитному портфелі якого частка простроченої заборгованості перевищує 3 %, вилучається з рейтингу. Крім того, система рейтингової оцінки за методикою О.Б. Ширинської характеризується дворазовим збільшенням кількості застосовуваних коефіцієнтів і ранжуванням їх в окремі групи, при цьому присвоєння коригувальних ваг здійснюється не лише коефіцієнтам, а й групам, до яких вони віднесені. Після присвоєння коригувальних ваг для кожної групи обчислюються узагальнюючі коефіцієнти шляхом перемноження наведених вище коефіцієнтів на встановлені для них емпіричні ваги та подальшого їх підсумовування. Таким чином запропоновані в методі О.Б. Ширинської нововведення дозволяють врахувати кредитні ризики та інші впливові фактори, а присвоєння коригувальних ваг сприяє підвищенню точності рейтингового оцінювання. Проте недостатньо розкритим залишається питання обґрунтованого вибору показників, які всебічно характеризують діяльність банку.

Новим підходом до рейтингового оцінювання характеризується також методика, запропонована вітчизняними банківськими експертами [4, С.261]. Вона передбачає врахування таких параметрів як рівень загальних активів, проблемних кредитів, коефіцієнт миттєвої ліквідності, рівень левериджу, коефіцієнт відкритої валютної позиції. В цій методиці загальну кількість показників зведено до мінімуму, відвівши основну роль проблемним кредитам. До них автори пропонують відносити 100% прострочених і 45,8% пролонгованих кредитів. Найменше ж уваги приділяється коефіцієнту відкритої валютної позиції у зв'язку з пильним контролем за нею з боку НБУ. У вітчизняній методиці аналіз базується на порівнянні показників кожного банку із середніми значеннями досліджуваної сукупності. Після проведення нормування і перемноження отриманих значень на відповідну питому вагу добутки підсумовуються. Також автори методики надають важливого значення оптимальності показника миттєвої ліквідності (розрахованого як різниця між його середнім значенням та середньолінійним відхиленням цього показника за щомісячними даними банку), адже його надмірне зростання призводить до істотного зниження рівня прибутковості банку [4, С.262 ].

Отже, аналізуючи дану методику слід відмітити, що її сильною стороною є розрахунок показника миттєвої ліквідності, причому врахування не лише загальної динаміки його змін, а й амплітуди коливань. Також позитивним моментом є проведення нормування коефіцієнтів шляхом розрахунку їхніх середніх значень та середньоквадратичних відхилень. Проте при проведенні аналізу до складу індикативної групи поряд з малими та середніми не були включені деякі великі банки, що знижує достовірність розрахунків оптимальних показників.

Здійснити дистанційний моніторинг фінансової стабільності банківської системи шляхом розрахунку рейтингової оцінки надійності банківських установ на базі економічних нормативів НБУ дозволяє метод В.П. Матвієнко [10, С.27]. Для реалізації методу здійснюється обґрунтований вибір показників, які найповніше характеризують діяльність банку. По кожному з них задається граничне (нормативне) значення. Показники мають бути невід'ємними і однонаправленими, тобто їх зростання означає покращання стану банку і навпаки. Цей метод дозволяє визначити найкращі та найгірші банки з точки зору надійності, проте у ньому немає можливості нормування коефіцієнтів для співставлення багатьох параметрів різних банків.

Одним з відомих методів подолання незрівнянності багато-параметричних систем є метод зведених показників (МЗП) [11]. Згідно цього методу на основі показників рейтингового оцінювання за методом В.П. Матвієнко отримують рейтинги фінансових установ, що відповідають висунутим замовникам вимогам. Розрахунок фінансових коефіцієнтів здійснюється на базі даних, що публікуються НБУ. Таким чином, метод зведених показників найбільш повно використовує відкриту для користувача інформацію та об'єктивно і всебічно дозволяє провести рейтингову оцінку банківської установи. Заслуговує на увагу також метод рейтингової оцінки банків, який застосовують фахівці журналу «Експерт» [12]. В якості вхідних даних у цьому методі використовуються баланси банків за рахунками другого порядку, а також оборотно-сальдові відомості. На основі цих даних розраховується дуже широкий спектр показників, які характеризують різні сфери діяльності банків. До таких показників можна віднести: активи-нетто, власний капітал, коефіцієнт ділової активності, достатність капіталу, генеральний коефіцієнт рентабельності, коефіцієнт іммобілізації капіталу, коефіцієнт поточної ліквідності. Для розрахунку коефіцієнтів попередньо проводять класифікацію банків за розміром і характером діяльності фінансових установ. Класифікація банків у групах проводиться на основі узагальненого значення коефіцієнта фінансової стійкості, який розраховується на основі 22 власних коефіцієнтів фінансової стійкості, які розраховуються для окремо обраного банку. В результаті аналізу формули розрахунку коефіцієнта фінансової стійкості можна казати, що методика журналу «Експерт» відає перевагу в оцінюванні розміру власного капіталу банку, тобто чим більший розмір власного капіталу банку, тим вище його фінансова стійкість. Загалом методика є детальною та доступною для розрахунку, але її об'єктивність є недостатньою для визначення інших впливових факторів на діяльність фінансових установ. Отже, проведений аналітичний огляд сучасних методів рейтингового оцінювання комерційних банків дозволив визначити їх основні характеристики, переваги та недоліки, які для зручності проведення порівняльного аналізу були зведені у табл. 1

Таблиця 2 Порівняльна характеристика моделей рейтингової оцінки

Групи показників	Рейтингові моделі						МЗП	Вітчизняна методик
	CAMELS	В.С. Кромонова	EUROMONEY	П.В. Матвієнко	О.Б. Ширинської	«Експерт»		
Капітал	+	+	+	+	+	+	+	+
Якість активів	+	-	-	-		+	-	+
Рентабельність	-	-	-	+	+	+	+	-
Прибутковість	+	+	+	-	+	-	+	-
Ліквідність	+	+	+	+	+	+	+	+
Надійність	-	+	-	-		-	-	-
Захищеність	-	+	-	-	x	-	-	-
Ризики	+	-	-	+		-	+	-
Менеджмент	+	-	-	-	x	-	-	-

Стратегія розвитку	+	-	-	-	-	-	-
--------------------	---	---	---	---	---	---	---

Аналіз наведених вище характеристик методів рейтингової оцінки дозволив встановити, що найбільш повно враховує усі показники діяльності банківської установи та дозволяє здійснити їх детальну оцінку метод CAMELS, проте результати рейтингового оцінювання за даним методом є конфіденційною інформацією і можуть використовуватися тільки Національним банком України.

Серед інших розглянутих методів можна виділити як кращі – методику журналу «Експерт», метод зведених показників та метод запропонований вітчизняними експертами. Однак, ці методи теж мають недоліки, серед яких найбільш суттєвими є необґрунтованість вибору параметрів для аналізу та неможливість співставлення показників діяльності різних банківських установ.

Для усунення цих недоліків можна запропонувати використання регресійного аналізу для проведення рейтингового оцінювання діяльності комерційного банку. Цей метод передбачає визначення взаємозв'язку між результатами банківської діяльності і факторами, які їх визначають. Для цього спочатку із сукупності показників діяльності фінансової установи визначаються ті, які не корелюють між собою. Далі проводиться побудова моделі результуючого інтегрального показника діяльності банку на основі вагомих факторів впливу серед сукупності обраних для моделювання показників. Використаємо математичний апарат регресійного аналізу [13, 14], завданням якого є знаходження формульної залежності, що пов'язує значення функції відгуку з вагомими факторами впливу. Основною метою математичного моделювання є отримання лінійної регресійної моделі, яка в загальному вигляді за [13] описується виразом (1):

$$f\{Y\} = \beta_0 + \sum_{i=1}^n \beta_i x_i + \sum_{\substack{i:l=1 \\ i<l}}^n \beta_{il} x_i x_l, \quad (3)$$

де  $f\{Y\}$  – цільова функція, що характеризує інтегральний показник діяльності банку,  $\beta_i, i = 1, 2, \dots, l$  – невідомі коефіцієнти впливу,  $x_i, i = 1, 2, \dots, l$  – фактори впливу,  $n$  – загальна кількість досліджень факторів впливу.

Оцінювати теоретичні коефіцієнти  $\beta_i, \beta_{il}$  будемо на основі вибірових коефіцієнтів впливу  $b_i, b_{il}$ . Для цього спочатку здійснимо вибір інтервалів варіації за кожним з факторів. Під інтервалом варіації згідно теорії статистичного моделювання [13] розуміємо число (характерне для кожного фактора), яке в результаті додавання до нульового рівня, дає верхній, а в результаті віднімання – нижній рівні фактору. На першому етапі (при отриманні лінійної моделі) фактори завжди змінюють тільки на двох рівнях. Інтервал варіації не може бути меншим від похибки, з якою фіксується рівень фактора, інакше верхній та нижній рівні будуть однаковими. Крім того, нижній та верхній рівні повинні, знаходитися в межах області визначення факторів [13, 14].

Якщо кількість факторів відома, тоді для варіювання факторів на двох рівнях у лінійній моделі кількість спостережень факторів можна визначити за формулою (4):

$$n = 2^k, \quad (4)$$

де  $n$  – кількість досліджень факторів,  $k$  – кількість факторів.

Для спрощення планування спостережень і оброблення результатів рекомендується перетворення розмірених керованих факторів у безрозмірні нормовані  $z_i = (x_i - x_{i0}) / \Delta x_i$ , що дає змогу спростити запис матриці планування, оскільки верхні та нижні рівні варіювання  $z_{ie}$  та  $z_{in}$  у відносних одиницях дорівнюють +1 та -1 незалежно від фізичної природи факторів та значень основних інтервалів варіювання факторів  $\Delta x_i$ . [13].

Процес знаходження математичної моделі за [13, 14] має таку послідовність: планування та проведення досліджень факторів; перевірку відтворення (однорідності вибірових дисперсій  $s_j^2$ ); отримання математичної моделі об'єкта досліджень з перевіркою статистичної значимості вибірових коефіцієнтів регресії та перевірку адекватності математичного описання.

Математичне планування досліджень передбачає побудову матриці згідно наступним вимогам [13, 14]:

1. Кожний  $i$ -тий рядок матриці повинен містити набір координат  $z_{ig}$  точки, в якій проводиться дослідження  $g$ -того фактора.



2. Оскільки змінні  $z_i$  приймають значення лише +1 та -1, усі взаємодії  $z_i, z_l (i; l = 1, 2, 3; i \neq l)$  можуть приймати тільки такі ж значення.

3. У першому рядку ( $i=1$ ) всі керовані фактори вибираються на нижньому рівні, тобто  $z_i = -1$ . Наступні  $i$ -ті варіанти варіювання при складанні матриці планування вибираються так, щоб при порядковому переборі всіх варіантів частота зміни знаку факторів для кожного наступного фактора  $z_{i+1}$  була вдвічі менша, ніж для попереднього  $z_i$ .

Після проведення досліджень необхідно перевірити гіпотезу про рівність генеральних дисперсій  $\sigma^2 \{Y_1\} = \sigma^2 \{Y_2\} = \dots = \sigma^2 \{Y_n\}$  за формулою (5):

$$s^2_j = s^2_{ВДТ} = \frac{1}{l-1} \sum_{g=1}^k (Y_{gj} - \bar{Y}_j)^2. \quad (5)$$

Так як всі оцінки дисперсій отримані за вибірками однакової кількості паралельних досліджень, кількість ступенів свободи для всіх складе:  $v_{ВДТ} = m - 1$ .

Для перевірки гіпотези про однорідність оцінок  $s^2_j$  дисперсій необхідно використати критерій Кохрена [13], який оснований на законі розподілу відношення максимальної оцінки дисперсії до суми всіх зіставлених оцінок дисперсій як відображено у виразі (6):

$$G = \frac{\max\{s_j^2\}}{\sum_{g=1}^k s_j^2 \{Y\}} \quad (6)$$

Для отримання математичного опису функції відгуку необхідно визначити незалежні оцінки  $b_0, b_i, b_{il}$  відповідних коефіцієнтів  $\beta_0, \beta_i, \beta_{il}$ , тобто  $b_0 \rightarrow \beta_0, b_1 \rightarrow \beta_1, b_2 \rightarrow \beta_2, b_3 \rightarrow \beta_3$ . Ці оцінки знаходяться за виразами (7-9) [14]:

$$b_0 = \frac{1}{n} \sum_{i=1}^n z_0 \bar{Y}_j. \quad (7)$$

$$b_i = \sum_{i=1}^n \bar{Y}_j z_{ij} / n, j = 0, 1, 2, \dots, m, i = 1, 2, \dots, n. \quad (8)$$

$$b_{il} = \frac{1}{n} \sum_{i=1}^n z_{ij} z_{lj} \bar{Y}_j, (i; l = 1, 2, \dots, n; i \neq l) \quad (9)$$

Після визначення оцінок  $b$  коефіцієнтів регресії, необхідно перевірити гіпотези про їх значущість, тобто перевірити відповідні нуль-гіпотези  $\beta = 0$ . Перевірка гіпотез про значущість коефіцієнтів виконується за допомогою критерію Стьюдента (10, 11) [14]:

$$t = |b| / s\{b\}, \quad (10)$$

$$s^2\{b\} = \frac{1}{nm} s^2_{ВДТ} \{Y\}, \quad (11)$$

де  $s^2\{b\}$ - дисперсія оцінки коефіцієнта  $b$ ;  $n$  – кількість спостережень,  $m$ – кількість паралельних спостережень.

За вимогами значущості, якщо обчислена величина  $t$  для відповідного коефіцієнта  $b$  перевищує значення  $t_{кр}$ , знайдене із таблиці  $t$ -розподілу Стьюдента, то нуль-гіпотезу  $H_0: \beta = 0$  відкидають і відповідну оцінку коефіцієнта визнають значущою. В іншому випадку нуль-гіпотезу не відкидають і оцінку  $b_i$  вважають статистично незначущою, тобто  $\beta = 0$ .

Перевірка адекватності математичного опису функцій відгуку полягає в оцінці відхилення передбачуваної за отриманим рівнянням регресії величини відгуку  $\hat{Y}_j$  від результатів досліджень  $\bar{Y}_j$  в одних і тих же точках факторного простору. Розсіювання результатів спостережень поблизу

рівняння регресії, що оцінює істинну функцію відгуку, визначається за допомогою дисперсії адекватності (12) [13,14]:

$$s_{AD}^2 = \frac{m}{n-d} \sum_{i=1}^n (\bar{Y}_j - \hat{Y}_j)^2, \quad (12)$$

де  $d$  – кількість членів апроксимуючого поліному.

Для того, щоб перевірити адекватність моделі необхідно розрахувати співвідношення між дисперсією адекватності  $s_{AD}^2$  та оцінкою дисперсії відтворюваності відгуку  $s_{ВДТ}^2\{Y\}$ . Якщо ці оцінки дисперсії однорідні, то математичний опис адекватно відтворює результати досліджень, у іншому випадку опис вважається неадекватним. Використання  $F$ -критерію Фішера дозволяє перевірити гіпотезу про однорідність двох вибірових дисперсій  $s_{AD}^2$  і  $s_{ВДТ}^2\{Y\}$ . У випадку, коли  $s_{AD}^2 > s_{ВДТ}^2\{Y\}$ ,  $F$ -критерій характеризується співвідношенням [12] (13):

$$F = s_{AD}^2 / s_{ВДТ}^2\{Y\}. \quad (13)$$

Далі із таблиці  $F$ -розподілення Фішера з рівнем значущості  $q$  для ступенів свободи  $v_{1AD} = n - d, v_{2AD} = n(m - 1)$  знаходиться значення  $F_{кр}$ . Якщо розраховане за (13) значення  $F$ -критерію Фішера є меншим від  $F_{кр}$ , гіпотезу про адекватність не відкидають і математичний опис визнається адекватним. В іншому випадку отриманий математичний опис вважається неадекватним і вимагає проведення додаткових досліджень.

Таким чином, побудована математична модель дає можливість дослідити та визначити коефіцієнт впливу кожного із попередньо визначених впливових факторів на вид цільової функції, що є інтегральним показником діяльності фінансової установи та застосувати відповідні заходи для оптимізації цього впливу.

**Висновки.** Таким чином, проведений аналітичний огляд та порівняльний аналіз характеристик сучасних методів та моделей рейтингового оцінювання дозволив встановити, що на сьогодні рейтинги банків є індикаторами їх надійності та ефективності для потенційних вкладників, інвесторів, банків-партнерів та державного банківського нагляду. Проведені дослідження стали підґрунтям для побудови регресійної моделі для рейтингового оцінювання діяльності комерційних банків, очевидною перевагою якої є отримання об'єктивної оцінки, заснованої на аналізі реальних взаємозв'язків між факторами впливу і функцією відгуку, що характеризує результати діяльності банківської установи.

В цілому, для підвищення ефективності рейтингового оцінювання можна запропонувати такі напрямки:

– розвиток національної мережі рейтингових агентств для можливості порівняння результатів рейтингів сформованих різними компаніями;

– удосконалення методики рейтингового оцінювання шляхом розробки системи якісних показників, що дозволять сформувані інтегральний показник рейтингу на підставі аналізу кількісних і якісних критеріїв;

– вдосконалення нормативно-правової бази, що дасть можливість мінімізувати подання неправдивої інформації банками та уникнути фальсифікованих рейтингів.

1. Пшик Б.І. Актуальні проблеми рейтингового оцінювання комерційних банків в Україні / Б.І. Пшик // Вісник університету банківської справи Національного банку України – 2014.– №2(20). – С.149 – 152
2. Майба В.В. Рейтингове оцінювання фінансової стійкості комерційних банків / В. В. Майба // Ефективна економіка. – 2010. – № 5. – С. 48 – 65.
3. Кириченко О. Банківський менеджмент. – К.: Основи. – 2011. – 311 с.
4. Турило А. М. Рейтингова система оцінки фінансового стану комерційного банку / А. М. Турило, І. С. Вчєрашня // Вісник КТУ. – 2011. – № 28. – С. 260 – 263.
5. Смірнов А.В. Аналіз фінансового стану комерційних банків: моногр. /А.В. Смірнов // Міжнародний банківський клуб «Аналітика без меж»: [Електронний ресурс]. –Режим доступу:<http://www.mbka.ru/item59/>.
6. ТОП-20 самых надежных банков Украины: [Електронний ресурс]. – Режим доступу: [http://finance.bigmir.net/kredit\\_depozit/51741-V-TOP-20-samyh-nadezhnyh-bankov-Ukrainy-tol-ko-vosem-ukrainskih](http://finance.bigmir.net/kredit_depozit/51741-V-TOP-20-samyh-nadezhnyh-bankov-Ukrainy-tol-ko-vosem-ukrainskih)
7. Кромонов В. С. Методика составления рейтинга надежности банков // Профиль. – 1998. – №20.

8. Домбровський В.С. Розробка автоматизованої моделі рейтингової оцінки банківських установ / В.С. Домбровський, Д.М. Григор'єв // Ефективна економіка – № 2 – 2012 р.– С. 145 – 153.
9. Ширинська Е.Б. Операции коммерческих банков / Е.Б. Ширинська.– М.: Финансы и статистика. – 2001. – 628 с.
10. Матвієнко П.В. Узагальнююча оцінка розвитку регіонів / Матвієнко П.В. // Економіка України. – 2007. – № 5. – С. 26 – 35.
11. Каширіна О. Рейтингова система оцінки фінансової стабільності банків з іноземним капіталом як інструмент підвищення надійності банківської системи / О. Каширіна, В. Фурсова // Вісник НБУ. – 2011. – № 9. – С. 33 – 41.
12. Фалюта А. В. Удосконалення рейтингового оцінювання банків / А. В. Фалюта // Міжнародна банківська конкуренція: теорія і практика : зб. тез доп. VII Міжнародної науково-практичної конференції (24 – 25 травня 2012 р.). – ДВНЗ «УАБС НБУ». – Суми, 2012. – Т. 1. – С. 102–104.
13. Іванова В.М. Математическая статистика. Учебник / [В.М. Іванова, В.Н. Калініна, Л.А. Нешумова та ін.]. – [2-е вид.]. – М.:Вища школа, 1981. – 371 с.
14. Адлер Ю.П. Планирование эксперимента при поиске оптимальных условий / Ю.П.Адлер, Е.В.Маркова, Ю.В.Грановский. – Москва.:Наука, 1976. – 276с.

УДК 001(09)

Лісковець С.М.

Луцький національний технічний університет

## МЕТОДИ ДОСЛІДЖЕННЯ ОКРЕМИХ ПЛОСКИХ ТРАНСЦЕНДЕНТНИХ КРИВИХ

**Лісковець С.М. Методи дослідження окремих плоских трансцендентних кривих.** Представлені узагальнені результати досліджень маловідомих праць обчислювача та математика Я.П. Кулика (1793 - 1863), які присвячені вивченню властивостей деяких плоских трансцендентних кривих, зокрема – однаково напружених ланцюгових ліній, ліній «ланцюгового моста», тощо. Приділена увага механізму виведення основних формул, аналізу характеристик даних ліній.

**Ключові слова:** однаково напружена ланцюгова лінія, лінія «ланцюгового моста», характеристики ліній.

**Рис. 2., Літ. 4.**

**Лісковець С.М. Методы исследования некоторых плоских трансцендентных кривых.** Представлены обобщенные результаты исследования малоизвестных работ вычислителя и математика Я.Ф. Кулика (1793 - 1863), которые посвящены изучению свойств некоторых плоских трансцендентных кривых, а именно, одинаково напряженных цепных линий, линий «цепного моста». Уделено внимание механизму вывода основных формул, анализу характеристик данных линий.

**Ключевые слова:** одинаково напряженная цепная линия, линия «цепного моста», характеристики линий.

**Liskovets S.M. The methods of research of some plane transcendental curves.** The generalized results of researches of lesser – known works by mathematician and numerator I.F.Kulyk are submitted, which are dedicated of studying on properties of some plane transcendental curves, especially – equally intense catenary «chain bridge», etc. Attention is paid on the mechanism of withdrawal of basic formulas, signature analysis of data lines.

**Keywords:** chain equally hard line, «chain bridge», characteristics of lines.

**Постановка проблеми.** Питаннями створення математичних моделей плоских кривих в цілому, та кривих, що мають широке практичне використання зокрема, займався багато вчених в різні історичні часи. Проблема пошуку оптимальної та вдосконаленої моделі дослідження кривих практичного застосування є актуальною в будь-який історичний період. Для отримання узагальнених найбільш раціональних напрацювань неабияку роль відіграє ґрунтовне вивчення шляхів вирішення проблем вченими попередніх поколінь.

**Мета дослідження** полягає в узагальненні розробок та напрацювань математика ХІХ століття Я.П. Кулика стосовно вивчення питань, що пов'язані з деякими плоскими трансцендентними кривими, зокрема з кривими практичного використання – однаково напруженими ланцюговими лініями, лініями «ланцюгового моста». Роботи вченого відповідної тематики практично не досліджувалися істориками математики, тому мета статті носить також інформативно-пізнавальний характер.

### **Виклад основного матеріалу дослідження.**

Згідно з історичними довідками, форму кривої провисання, яку пізніше назвали „ланцюговою лінією“, вперше досліджував Галілео Галілей (1564–1642), розробляючи кінематичні методи для знаходження дотичних та нормалей до кривих. Даний метод дав початок дослідженню траєкторій і складних рухів, визначенню дотичних в будь-якій точці траєкторії. Потрібно зазначити, що Галілей криву провисання вважав за параболу і, відповідно, всі основні підрахунки були зроблені для траєкторій, які мали вигляд параболи. Параметром такої параболи Галілей визначив висоту падіння точки, що збільшена в чотири рази, і яка потрібна для того, щоб надавати точці швидкість рівну початковій горизонтальній швидкості [1, с. 165–166].

Ланцюгові лінії використовувалися, в першу чергу, для розрахунків пов'язаних з провисанням дротів, тобто мали практичний інтерес, а отже, потребували детального вивчення. Крім того, поверхня, яка утворюється обертанням дуги ланцюгової лінії навколо осі  $OX$ , що називається катеноїдом, є мінімальною поверхнею і представляє також практичний інтерес. Вперше катеноїд описав Л. Ейлер, який досліджував властивості плоских кривих. Він розробив методи знаходження кривих та сімейств кривих, що задовольняють спеціальним властивостям. Ейлер поставив перед вченими-послідовниками окремі задачі, що мали відношення до детального дослідження властивостей як плоских, так і просторових кривих та поверхонь, що утворюють криві в результаті обертання навколо координатних осей [2, с. 116–117].

Дослідженням плоских кривих ліній Я.П. Кулик захопився ще на початку своєї наукової діяльності. В 1832 році в Празі з'явилася праця „Theorie und Tafeln der Kettenlinie“ („Теорія і

таблиці ланцюгових ліній“) [3]. Автор зауважував, що серед плоских кривих ланцюгова лінія виділяється багатьма цікавими властивостями, не дивлячись на те, що вона є найпростішою серед трансцендентних кривих та повністю визначається своїми параметрами. Він писав про криву: „її можна точно виміряти, центр її ваги розташований під кривими, які тісно пов'язані кінцевими точками, а поверхня того тіла, яка утворюється завдяки повороту ланцюгової лінії навколо горизонтальної осі, є найбільшою чи найменшою, залежно від осі обертання [3, с. 2]“. Крім того, дуже важливим та широким було застосування цих кривих в будівництві, зазначав Кулик. Вони є лінійною опорою так званих дротових та ланцюгових мостів, склепів, куполів, збудовані випуклості на кшталт кривих є лише тоді довговічними, якщо у них дотримується достатня кількість ліній опори [3, с. 2].

Вчений дав таке пояснення кривої, яку досліджував: „Якщо нитку, яка є одночасно важкою, але повністю гнучкою, підвісити тонкими кінцями за два нерухомих пункти  $A$  та  $B$ , то у стані рівноваги утворюється крива  $AMB$ , яку називають ланцюговою лінією. Точка  $M$  - довільна точка кривої [3, с. 3]“.

В роботі виведені всі основні формули, що відносяться до ланцюгових ліній. Слід відмітити, що математичний апарат представлений автором дуже детально.

В першу чергу, представлено рівняння ланцюгової лінії:

$$y = \frac{1}{2} p \left( e^{\frac{x}{p}} + e^{-\frac{x}{p}} \right),$$

або

$$x = p \lambda \frac{y + \sqrt{y^2 + p^2}}{p},$$

де  $p$  - параметр ланцюгової лінії. Знаком  $\lambda$  Кулик позначав натуральний логарифм.

В статті „Теорія та таблиці ланцюгової лінії“ багато уваги приділено знаходженню формул для обчислення основних характеристик кривої: радіуса кривизни, амплітуди, довжини дуги. При цьому запропоновані різні варіанти формул в залежності від початкових даних. Автор зазначав, що одним із завдань роботи є створення математичної бази для знаходження характеристик ланцюгової лінії, маючи кут положення точки кривої. Кулик не лише вивів формули, але й побудував таблицю, за допомогою якої, знаючи кут  $\nu$  положення точки  $M_0$ , можна відшукати координати точки  $(x, y)$ , довжину дуги  $S$  від точки до вершини кривої, радіус кривизни  $R$ , нормаль  $N$ , параметр  $t$ , яким автор позначив заглиблення точки. Таблиця побудована для всіх кутів  $\nu$  від  $1^0$  до  $85^0$ .

Продовжив дослідження про ланцюгові лінії взагалі та про лінії ланцюгового моста Кулик в роботі „Untersuchungen über die Kettenbrückenlinie“ („Дослідження про лінії ланцюгового моста“) [4], яка з'явилася в Празі у 1838 році. За словами автора, основною метою роботи є точне пояснення якостей тих кривих ліній, які беруть на себе роль підвісних ланцюгів ланцюгового моста, в той час, коли вони крім своєї власної ваги, мусять витримувати разом з проїзною частиною ще вагу підпираючих жердин [4, с. 3]. Вчений саме такі лінії називав лініями ланцюгових мостів. Термін „лінії ланцюгового моста“ вперше ввів Кулик (про цей факт зазначав сам автор у вступі до своєї роботи).

Автор зауважував, що коли розглядати лінію ланцюгового моста як параболу, еліпс і т.д., тоді довжину підпираючих жердин можна розрахувати заздалегідь, але тоді виникає необхідність експериментальним шляхом скоротити, в залежності від обставин, підпираючі жердини, або ж замінити їх на довші, поки приблизно не буде досягнуто передбачуваної лінійності проїзної частини. В іншому випадку, що найчастіше і траплялося, проїзна частина була не горизонтальною, а з вгнутою серединою.

В роботі запропонований, порівняно нескладний, метод викривлення конічних перерізів, який через свою простоту міг би викликати, за словами автора, інтерес у конструкторів та інженерів. Крім того, таблиці для необхідних підрахунків, що наведені в роботі, істотно відрізнялися від всіх відомих на той час високою точністю.

В першій частині праці „Однаково напружена ланцюгова лінія“ автор розглядав загальну ланцюгову лінію та відхилену від неї криву лінію, яка утворюється внаслідок зміни напруження

вздовж лінії ланцюгового моста. Вчений зазначав, що загальна ланцюгова лінія утворюється тоді, коли гнучкий та рівномірно товстий ланцюг буде прикріплений на кінцях таким чином, щоб усі елементи вільно висіли. Навантаження, яке існує в окремих точках, стабільно зростає з середини ланцюгової лінії, тобто з її вищої точки, і в точках підвішування воно саме найбільше [4, с.4]. Продовжуючи дослідження, вчений запропонував розглядати нерівномірно товстий ланцюг, який має таку властивість: міцність ланцюга зростає від середини до двох кінців в такому ж відношенні як збільшується напруження. В такому випадку можна стверджувати, що ланцюг із загальної ланцюгової лінії утворює відхилену криву лінію, яка може через однакову міцність, або однакове напруження всіх її частин називатися однаково напруженою ланцюговою лінією [4, с. 5].

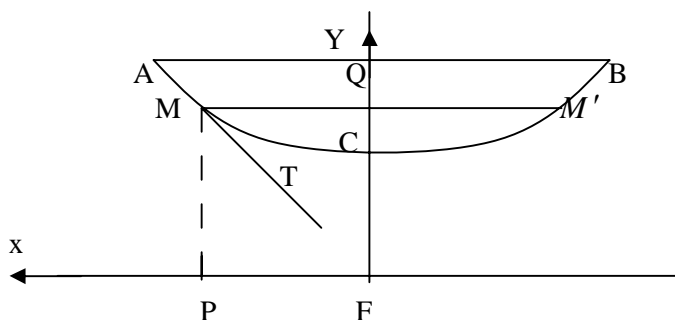


Рис.1. Ланцюгова лінія

Нехай точки А, В, С представляють загальну ланцюгову лінію (рис.1), де А, В – точки підвішування, а С – її вершина. Через будь-яку точку М заданої лінії проведена горизонталь MQ та дотична MT. Кут QMT називається кутом положення точки М. Якщо цей кут позначити  $\nu$ , а довжину дуги CM від вершини  $S$ , тоді справедлива формула:

$$S = p \cdot \operatorname{tg} \nu,$$

Величина  $p$  – параметр ланцюгової лінії, це довжина дуги точки, кут положення якої становить  $45^\circ$ , тобто для  $\nu = 45^\circ$ ,  $S=p$ .

Якщо через С провести вертикальну пряму CY та до CY провести перпендикуляр PF, який називається директрисою ланцюгової лінії, тоді можна визначити закони, за якими змінюються координати будь-якої точки М ланцюгової лінії (абсциса  $x=FP$ , ордината  $y=PM$ ):

$$x = p \cdot \lambda \operatorname{tg} \left( 45^\circ + \frac{1}{2} \nu \right),$$

$$y = p \cdot \sec \nu.$$

Довжина дуги обчислюється за формулою:

$$S = \sqrt{y^2 - p^2}.$$

Більш детально в роботі описане виведення формул, що відносяться до однаково напруженої ланцюгової лінії. Такий тип ліній був вперше введений саме Куликом. Підхід до виведення формул для обчислення довжини дуги однаково напруженої ланцюгової лінії аналогічний як для загальної ланцюгової лінії, але враховується напруження в будь-якій точці, вагу одиниці довжини лінії та поперечні перерізи ланцюга у вершинах дуги.

Довжина дуги однаково напруженої лінії обчислюється за формулою:

$$S = p \cdot \lambda (\operatorname{tg} \nu + \sec \nu),$$

або

$$S = p \cdot \lambda \left( \operatorname{tg} 45^\circ + \frac{1}{2} \nu \right).$$

Розраховані таблиці для визначення дуги однаково напруженої ланцюгової лінії [4, с. 7]. Значення в таблиці залежать від кута положення точки лінії. Наведені дані для всіх кутів, що збільшуються на шість мінут.

Наприклад: для  $\nu = 17^\circ 48'$   $S = 31,5790$ ;  
 $\nu = 17^\circ 42'$   $S = 31,3957$ .

Дані в таблиці наводяться для параметра  $p=100$ , але автором розроблений математичний механізм для користування таблицями з іншим параметром  $p'$ .

Для полегшення роботи з однакою напруженою ланцюговою лінією будуються окремі таблиці, зокрема:

– „Координати, довжини дуг, напруження та кути положення однакою напруженої ланцюгової лінії“;

– „Кути положення точок підвішування“.

Стаття наповнена поетапними математичними виведеннями. Вчений запропонував в кількох варіантах формули, що обчислюють ті, чи інші характеристики за різними вихідними даними, обґрунтовує спосіб підрахунку необхідних логарифмів, які забезпечують більшу точність.

Так, для однакою напруженої ланцюгової лінії виведені такі формули для знаходження координат точок:

$$\begin{aligned}x &= p\nu, \\y &= p(\lambda \sec \nu + 1)\end{aligned}$$

де  $p$  – параметр кривої (директриса),  $\nu$  – кут положення точки, що обчислюється в радіанній мірі, параметр  $\lambda$  позначає натуральний логарифм.

Напруження  $T$  точки кривої обчислення за формулою:

$$T = ph \sec \frac{x}{p},$$

де  $h$  – відстань від точки до осі ординат.

Вивівши формули для обчислення характеристик однакою напруженої ланцюгової лінії, Кулик зробив такі висновки [4, с. 9–10]:

Однаково напружена лінія:

– є трансцендентною кривою;

– складається з двох нескінченних ліній, оскільки для  $\nu = 90^\circ$ ,  $x$  переходить в  $\frac{1}{2}\rho\pi$ , а  $y$  стає нескінченним;

– оскільки для  $\nu = 0$ ,  $x = 0$  і  $y = p$ , то всі точки ланцюгової лінії лежить на відстані  $y=p$ ;

– оскільки для одночасно додатної і від'ємної величини  $x$ , величина  $y$  залишається незмінною, то вісь координат поділяє криву на дві рівні частини;

– для абсцис більших за  $\frac{1}{2}\rho\pi$  немає ніяких точок кривої;

– всі однакою натягнуті ланцюгові лінії є подібними, отже, відповідні плоскі фігури, що обмежені лініями – подібні.

Значна частина досліджень Кулика присвячена ланцюговим лініям практичного використання, а саме лініям ланцюгового моста. Автор визначив рівняння таких ліній при умові, що вага несучих ланцюгів разом з вагою несучих жердин та проїзної частини приєднуються до одного горизонтального навантаження.

Нехай  $A, B$  – точки підвішування,  $C$  – вершина кривої  $AMB$ , кут  $\angle PMT$  ( $\nu$ ) – це кут положення точки  $M$ ,  $CP=x$ ,  $MP=y$ ,  $AE=d$  – напіврозмах,  $CE=t$  – поглиблення всередині лінії,  $A'B'$  – проїзна частина,  $CD=p$  – параметр кривої (рис. 2).

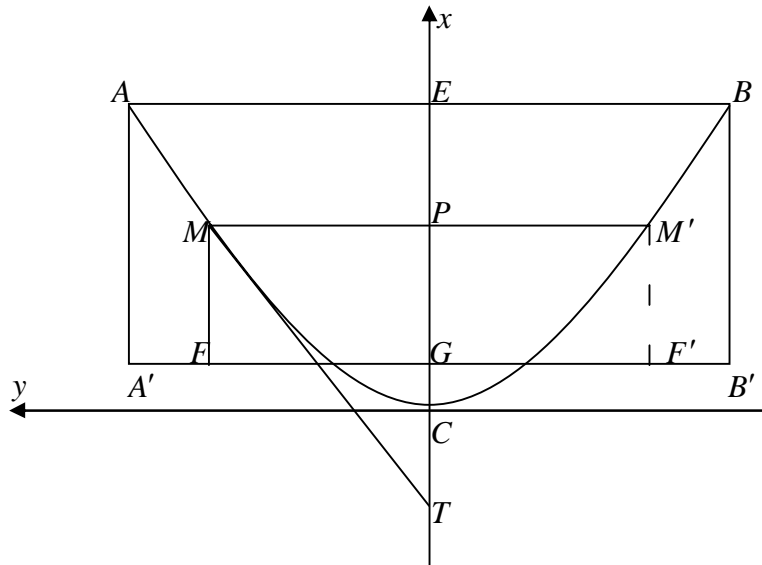


Рис. 2. Лінія ланцюгового моста

Через  $h$  вчений позначав вагу одиниці довжини, наприклад, поєднується вага одиниці довжини основи ланцюгів та проїзної частини з навантаженням. Для напруження  $T$  в точці  $M$  справедливі співвідношення

$$T \sin \nu = hy, \quad T \cos \nu = ph,$$

тому, що вага ланцюгової дуги  $MC$ , розташованої в тій самій частині  $FG$  проїзної частини, розглядається як єдине горизонтальне навантаження.

Кулик довів, що справедлива формула:

$$T = h\sqrt{p^2 + y^2}.$$

Згідно з підрахунками автора, рівняння лінії ланцюгового моста має вигляд:

$$y^2 = 2px, \quad \text{або} \quad y^2 = \frac{d^2x}{t} \quad (d^2 = 2pt).$$

Рівняння задають лінію параболічного типу з відповідним параметром.

В полярній системі координат рівняння лінії ланцюгового моста має вигляд:

$$r \approx \frac{\frac{1}{2}p}{1 + \cos a},$$

$a$  – кут нахилу точки кривої.

Вчений стверджував, що отримати найбільшу точність довжини дуги такої лінії можна за формулою:

$$S = \frac{1}{4}p \left( \operatorname{tg} \nu \cdot \sec \nu + \lambda \operatorname{tg} \left( 45^\circ + \frac{1}{2} \nu \right) \right). \quad [4, \text{с. } 20].$$

В роботі подається поетапне виведення даної формули з детальним поясненням.

Дослідник розрахував „Таблицю координат лінії ланцюгового моста“ та відповідних логарифмів, які суттєво зменшують громіздкі математичні підрахунки.

Я.П. Кулик зауважував, що у ланцюгових мостів навантаження ланцюгів є набагато важливішим ніж разом взяті вага підпираючих жердин (несучих стержнів), вага проїзної частини та вага випадкового навантаження моста. Своїми підрахунками вчений довів, що, у випадку об'єднання останнього навантаження з першим, можна отримати однаково напружену ланцюгову лінію, яка є ближчою до лінії ланцюгового моста, ніж до параболи. Із ще більшою точністю можна визначити ланцюгову лінію моста, якщо кожне навантаження виділити у відповідності до свого напрямку та розрахувати окремо. Зроблені відповідні підрахунки для випадку, коли навантаження ланцюгів діє в напрямку дотичних окремих точок лінії, а вага проїзної частини, несучих стержнів та вага випадкового навантаження моста розглядається як горизонтальне навантаження ланцюгів.



В розділі „Про лінії ланцюгового моста“ виводяться формули для характеристик даної лінії, ввів нові поняття, такі як модуль лінії  $k$ , кут кривої  $\omega$ , які, як стверджує автор, більш точно описують саму лінію. Так, координати точок лінії можна обчислити за формулами:

$$x = \frac{P}{\sqrt{k}} \cdot \omega, \quad y = \frac{P}{k} \lambda \sec \omega + p.$$

Важливими повинні бути розрахунки відповідних навантажень, які б максимально впливали на міцність та безпеку ланцюгових мостів.

#### **Висновки**

Дослідження, математичний апарат та розрахунки, описані в роботах [3,4], є підґрунтям для теоретичних напрацювань при вивченні деяких плоских трансцендентних кривих. Достатньо обґрунтованим є представлений аналіз однаково напруженої ланцюгової лінії та лінії «ланцюгового моста», а також їх практичне використання. Підхід до створення відповідної математичної бази кривих, методи розрахунків та обчислень характеристик ланцюгових ліній можуть розширити та доповнити напрацьований матеріал даної тематики.

1. Рыбников К.А. История математики / К.А. Рыбников. – М.: Из-во МГУ, 1974. – 455 с.
2. Развитие идей Леонарда Эйлера и современная наука. [Сборник статей под редак. Боголюбова Н.Н., Михайлова Г.К., Юшкевича А.П.] – М.:Наука, 1988. – 518 с.
3. Kulik J.P. Theorie und Tafeln der Kettenlinie / J.P. Kulik. – Prag, 1832. – 50 s.
4. Kulik J.P. Untersuchungen über die Kettenbrückenlinie / J.P. Kulik – Prag, 1838. – 38 s.

УДК 378.147

Губаль Г.М.

Луцький національний технічний університет

## ЗАСТОСУВАННЯ КЕЙС-МЕТОДУ У ВИКЛАДАННІ ВИЩОЇ МАТЕМАТИКИ ТА ПРИКЛАД НАУКОВО-ДОСЛІДНОГО КЕЙСУ

Губаль Г. М. Застосування кейс-методу у викладанні вищої математики та приклад науково-дослідного кейсу. У статті розглядається застосування кейс-методу у викладанні вищої математики та приклад науково-дослідного кейсу. Обґрунтовано важливість застосування кейс-методу для засвоєння знань. Показано шляхи підвищення якості та ефективності процесу навчання з вищої математики застосуванням кейс-методу у викладанні вищої математики. Розроблено приклад науково-дослідного кейсу.

**Ключові слова:** кейс-метод, ситуація, кейс-завдання, науково-дослідний кейс, диференціальне рівняння.

Літ. 6.

Губаль Г. Н. Применение кейс-метода в преподавании высшей математики и пример научно-исследовательского кейса. В статье рассматривается применение кейс-метода в преподавании высшей математики и пример научно-исследовательского кейса. Обосновано важность применения кейс-метода для усвоения знаний. Показано пути повышения качества и эффективности процесса обучения высшей математике использованием кейс-метода в преподавании высшей математики. Разработано пример научно-исследовательского кейса.

**Ключевые слова:** кейс-метод, ситуация, кейс-задание, научно-исследовательский кейс, дифференциальное уравнение.

Лит. 6.

Hubal H. M. Application of Case study in teaching of higher mathematics and an example of a research Case study. Application of Case study in teaching of higher mathematics and an example of a research Case study are considered in the article. The importance of Case study to the assimilation of knowledge is proved. Ways to improve the quality and effectiveness of learning higher mathematics using Case study in teaching of higher mathematics are presented. An example of a research Case study is presented.

**Keywords:** Case study, situation, Case assignment, research Case study, differential equation.

Bibl. 6.

**Вступ.** Для досягнення високих результатів у навчанні необхідно використовувати в навчальному процесі активні й інтерактивні методи навчання.

Серед інтерактивних методів навчання в даний час все більшу актуальність набуває кейс-метод (Case study) – метод аналізу ситуацій [1-6]. Суть методу полягає в тому, що студентам пропонують обдумати реальну життєву ситуацію, опис якої одночасно відображає не тільки яку-небудь практичну проблему, але й актуалізує певний комплекс знань, який необхідно засвоїти при розв'язанні даної проблеми. При цьому сама проблема не має однозначних рішень.

Кейс-метод базується на наданні студентам інформаційних освітніх ресурсів у вигляді спеціальних наборів (кейсів) навчально-методичних матеріалів, призначених для вивчення. Завдання викладача полягає у виборі реального матеріалу, а студенти повинні розв'язати поставлену проблему і одержати оцінку своїх дій від інших студентів і викладача. При цьому можливі різноманітні варіанти розв'язання проблеми. Тому викладач повинен допомогти студентам розмірковувати, дискутувати, сперечатись, а не нав'язувати їм свою думку. Викладач направляє бесіду або дискусію, наприклад, за допомогою проблемних питань у втягування студентів у процес аналізу кейса і пошуку варіантів розв'язання. При цьому викладач може нагадувати теоретичний матеріал, пояснювати, узагальнювати.

**Основна частина.** Кейс-метод найчастіше застосовується при вивченні гуманітарних дисциплін, однак він може бути використаний і при вивченні вищої математики.

Зауважимо, що кейс-метод почав застосовуватись ще на початку ХХ-го століття в області права і медицини, як метод активного навчання, найбільш наближений до практики.

Перевагами кейс-методу є: створення проблемної ситуації на основі фактів з реального життя, колективний характер пізнавальної діяльності, творчий підхід до пізнання, поєднання теоретичних знань і практичних навиків.

Типи кейсів:

- практичний;
- навчальний;
- науково-дослідний.

*Практичний кейс* містить життєві ситуації, в яких можливо застосування знань з вищої математики. При цьому їх навчальне призначення зводиться до тренінгу студентів, закріпленню знань, умінь і навиків з можливим включенням альтернативних ситуацій, з яких необхідно вибрати оптимальний варіант. Будується алгоритм дій з перевіркою проміжних відповідей.

*Навчальний кейс* містить навчальні (умовні) ситуації в предметній області «Вища математика». При цьому формується змістовна модель кейс-завдання. Наводяться взаємопов'язані підзадачі, розв'язання яких повинно привести до розв'язання поставленої задачі.

*Науково-дослідний кейс* містить математичну модель для одержання нових знань про ситуацію і поведінку в ній. При цьому допускається побудова декількох математичних моделей, які можуть використовувати різні розділи вищої математики, що приводять до розв'язання кейс-завдання. Навчання зводиться до науково-дослідних робіт.

Кейс-метод розвиває навчально-інформаційні компетенції, комунікативні компетенції, дозволяє встановити оптимальне поєднання теорії і практики.

Використання кейс-методу при вивченні вищої математики дає можливість:

- формувати представлення про вищу математику, використання якої дозволяє пізнавати дійсність, описуючи і вивчаючи реальні процеси і явища;
- розвивати вміння працювати з навчально-методичними матеріалами і науковими роботами з вищої математики, точно і грамотно математично виражати свої думки, використовувати логічні обґрунтування, доведення;
- розвивати вміння будувати і досліджувати математичні моделі реальних ситуацій;
- розвивати вміння застосовувати вивчений математичний апарат для розв'язування практичних задач і задач із суміжних дисциплін.

При розв'язуванні кейс-завдань з вищої математики необхідно дотримуватись наступних етапів:

- знайомство з ситуацією, її особливостями. Аналіз ситуації і визначення проблеми;
- визначення можливих методів розв'язання проблеми;
- прийняття рішень по вибору методу і теоретичного апарату;
- побудова математичної моделі;
- розв'язання проблеми;
- перевірка розв'язання.

Кейс-завдання з вищої математики можуть надаватись як для індивідуальної, так і для групової роботи студентів.

Науково-дослідні кейс-завдання являються завданнями більш високого рівня складності.

Для наведеного нижче прикладу науково-дослідного кейсу необхідно застосувати знання з вищої математики та інших наук (екологія, біологія, ботаніка, зоологія, хімія, фізика).

*Кейс «Застосування диференціальних рівнянь та систем диференціальних рівнянь».*

Тип кейса: науково-дослідний.

Зміст кейса:

Навести приклади коливальних процесів у природі. Побудувати та розв'язати диференціальне рівняння для хімічної реакції в гомогенному (однорідному) середовищі, яка відображає важливі особливості періодичних хімічних реакцій (хімічних реакцій, які протікають у коливальному режимі) у біосистемах. Показати, що мають місце затухаючі коливання.

*Наведемо розв'язання даного кейс-завдання.*

Множина явищ навколишнього життя показує, що в тварин і рослин, у тому числі навіть в одноклітинних наявні біологічні годинники. Про це свідчать ритмічні скорочення серця, закриття віночків багатьох квітів з настанням темноти, періодична зміна інтенсивності фотосинтезу в рослин, коливання розмірів ядер у клітинах та інше.

Усе живе існує в зовнішніх умовах, що періодично змінюються: день змінюється на ніч, прилив – відливом, періодично чергуються пори року і т.д.

Очевидно, що для найкращого пристосування до періодичних зовнішніх умов необхідно мати свій годинник, щоб знати заздалегідь, коли, наприклад, наступить ніч, і встигнути до цього приготуватись.

Крім того, коливальні системи мають ряд важливих особливостей. Одна з них – здатність до взаємної синхронізації, завдяки чому годинник може налаштуватись вірно, і з множини слабо зв'язаних коливальних процесів виникає гармонія періодичного явища. Наприклад, на енцефалограмі, яка реєструє біоструми мозку, видно періодично повторювані піки, що слідує один за одним з частотою біля 9 *гц*. Це так званий  $\alpha$ -ритм мозку, який керує багатьма нервовими процесами. Зокрема, стробування зображення оком відбувається з частотою 9–10 *гц*.  $\alpha$ -ритм формується в результаті взаємної синхронізації множини елементарних генераторів біострумів, розміщених у мозку, які мають біохімічну природу.

Результат зовнішнього впливу на коливальний процес також залежить від того, у який момент часу цей вплив відбувся. Впливи, однакові за силою і характером, можуть привести до протилежних результатів, якщо вони здійснились у різні моменти, або, як кажуть, у різні фази. Інколи навіть слабка, але повторювана дія може здійснити сильний вплив, якщо вона діє в

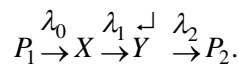
потрібний момент. Очевидно, що цю обставину природа використовувала і використовує при створенні досконалих витворів – живих істот.

Таким чином, живій природі необхідні коливальні процеси, які синхронізовані з часом доби.

Розглянемо побудову диференціального рівняння для хімічної реакції в гомогенному середовищі.

Нехай у деякому об'ємі знаходиться в надлишку речовина  $P_1$ , тобто в процесі реакції витрати речовини  $P_1$  майже непомітні. Молекули речовини  $P_1$  з деякою постійною швидкістю  $\lambda_0$  перетворюються в молекули речовини  $X$  (тут ми маємо справу з реакцією нульового порядку). Речовина  $X$  може перетворюватись у речовину  $Y$ . Це вже реакція другого порядку, оскільки швидкість її тим більша, чим більша концентрація речовини  $Y$ . У наведеній нижче кінетичній схемі на цю залежність указує обернена стрілка над символом  $Y$ . Молекули речовини  $Y$  у свою чергу незворотно розпадаються, у результаті утворюється речовина  $P_2$  (реакція першого порядку).

Кінетична схема цієї реакції виглядає наступним чином:



Побудуємо математичну модель цієї реакції, позначивши для спрощення символами  $X$ ,  $Y$  і  $P_2$  концентрації відповідних речовин:

$$\begin{cases} \frac{dX}{dt} = \lambda_0 - \lambda_1 XY, \\ \frac{dY}{dt} = \lambda_1 XY - \lambda_2 Y, \\ \frac{dP_2}{dt} = \lambda_2 Y. \end{cases} \quad (1)$$

Оскільки перші два диференціальні рівняння не залежать від  $P_2$ , то їх можна розглядати окремо. З'ясуємо спочатку, чи може ця реакція протікати так, щоб швидкість утворення речовини  $P_2$  залишалась постійною. Це буде в тому випадку, коли концентрації  $X$  і  $Y$  не змінюються з часом, тобто:

$$\frac{dX}{dt} = 0 \quad \text{і} \quad \frac{dY}{dt} = 0.$$

З цих умов згідно з (1) одержимо два алгебраїчні рівняння, які пов'язують рівноважні концентрації  $\bar{X}$  і  $\bar{Y}$ :

$$\begin{cases} \lambda_0 - \lambda_1 \bar{X} \bar{Y} = 0, \\ \lambda_1 \bar{X} \bar{Y} - \lambda_2 \bar{Y} = 0. \end{cases} \quad (2)$$

Розв'язком системи (2) є:

$$\bar{X} = \frac{\lambda_2}{\lambda_1}, \quad \bar{Y} = \frac{\lambda_0}{\lambda_2}.$$

Задамо для концентрацій  $X$  і  $Y$  малі відхилення  $x(t)$  і  $y(t)$  від рівноважних концентрацій  $\bar{X}$  і  $\bar{Y}$ , тобто:

$$X(t) = \bar{X} + x(t), \quad Y(t) = \bar{Y} + y(t).$$

Підставляючи ці вирази в рівняння (1) з урахуванням розв'язку системи (2) і що  $\bar{X}$  і  $\bar{Y}$  – величини сталі, одержимо систему диференціальних рівнянь для відхилень  $x(t)$  і  $y(t)$ :

$$\begin{cases} \frac{dx}{dt} = -\lambda_2 y - \lambda_1 x y - \frac{\lambda_1 \lambda_0}{\lambda_2} x, \\ \frac{dy}{dt} = \frac{\lambda_1 \lambda_0}{\lambda_2} x + \lambda_1 x y. \end{cases}$$

Нехтуючи членами, що містять величини другого порядку мализни  $x$  і  $y$ , одержимо лінеаризовану систему для відхилень:

$$\begin{cases} \frac{dx}{dt} = -\lambda_2 y - \frac{\lambda_1 \lambda_0}{\lambda_2} x, \\ \frac{dy}{dt} = \frac{\lambda_1 \lambda_0}{\lambda_2} x. \end{cases} \quad (3)$$

Зауважимо, що в системі (3) на відміну від системи (1) величини  $x$  і  $y$  можуть змінювати знак, тоді як вихідні змінні  $X$  і  $Y$ , які являються концентраціями, можуть бути тільки додатними.

Позначимо  $\frac{\lambda_1 \lambda_0}{\lambda_2} = 2\delta$ ,  $\lambda_1 \lambda_0 = \delta_1^2$  і перейдемо до одного диференціального рівняння другого порядку для  $x(t)$ . Тоді, запишемо систему (3) у вигляді:

$$\begin{cases} \frac{dx}{dt} = -\frac{\delta_1^2}{2\delta} y - 2\delta x, \\ \frac{dy}{dt} = 2\delta x. \end{cases}$$

Продиференціювавши перше з одержаних рівнянь по  $t$ , одержимо:

$$\frac{d^2 x}{dt^2} = -\frac{\delta_1^2}{2\delta} \frac{dy}{dt} - 2\delta \frac{dx}{dt}.$$

Підставляючи в одержане диференціальне рівняння значення  $\frac{dy}{dt}$  з другого рівняння системи, одержимо диференціальне рівняння:

$$\frac{d^2 x}{dt^2} + 2\delta \frac{dx}{dt} + \delta_1^2 x = 0. \quad (4)$$

Характеристичне рівняння цього лінійного однорідного диференціального рівняння зі сталими коефіцієнтами має вигляд:

$$k^2 + 2\delta k + \delta_1^2 = 0.$$

Його корені  $k_{1,2} = -\delta \pm \sqrt{\delta^2 - \delta_1^2}$ .

Якщо  $\delta^2 - \delta_1^2 < 0$  або  $\lambda_2^2 > \frac{\lambda_1 \lambda_0}{4}$  (корені характеристичного рівняння комплексно-спряжені), то, поклавши  $\delta^2 - \delta_1^2 = -\omega^2$  (тоді  $k_1 = -\delta + \omega i$ ,  $k_2 = -\delta - \omega i$ ), загальний розв'язок рівняння (4) можна записати у вигляді:

$$x = e^{-\delta t} (C_1 \cos \omega t + C_2 \sin \omega t). \quad (5)$$

Зауважимо, що нерівність  $\lambda_2^2 > \frac{\lambda_1 \lambda_0}{4}$  одержується з нерівності  $\delta^2 - \delta_1^2 < 0$  наступним чином:

$$\delta^2 - \delta_1^2 = \left( \frac{\lambda_1 \lambda_0}{2\lambda_2} \right)^2 - \lambda_1 \lambda_0 = \lambda_1 \lambda_0 \left( \frac{\lambda_1 \lambda_0}{4\lambda_2^2} - 1 \right).$$

Тоді  $\lambda_1 \lambda_0 \left( \frac{\lambda_1 \lambda_0}{4\lambda_2^2} - 1 \right) < 0$ . Оскільки  $\lambda_1$  і  $\lambda_0$  можуть бути тільки додатними, то  $\frac{\lambda_1 \lambda_0}{4\lambda_2^2} - 1 < 0$ ,

звідки  $\lambda_2^2 > \frac{\lambda_1 \lambda_0}{4}$ .

Помноживши і поділивши праву частину рівняння (5) на  $\sqrt{C_1^2 + C_2^2}$  і позначивши

$$\sqrt{C_1^2 + C_2^2} = A, \quad \frac{C_1}{\sqrt{C_1^2 + C_2^2}} = \sin \alpha, \quad \frac{C_2}{\sqrt{C_1^2 + C_2^2}} = \cos \alpha$$

(враховуючи, що  $\sin^2 \alpha + \cos^2 \alpha = 1$ ), одержимо:

$$x = Ae^{-\delta t} (\sin \alpha \cos \omega t + \cos \alpha \sin \omega t).$$

Тоді розв'язок (5) запишемо у вигляді:

$$x = Ae^{-\delta t} \sin(\omega t + \alpha). \quad (6)$$

Вираз (6) вказує на те, що мають місце затухаючі коливання, де  $\omega$  – частота коливань,  $Ae^{-\delta t}$  – амплітуда коливань, яка в початковий момент часу ( $t = 0$ ) дорівнює  $A$ , а з плином часу зменшується.

Зауважимо, що однією з самих досконалих коливальних систем, створених природою, є серце. Правильність роботи серця визначається синхронною роботою груп м'язів, які забезпечують позмінне скорочення шлуночків і передсердь. Синхронізацію роботи серця здійснює синусовий вузол серця, що виробляє з певною частотою синхронізуючі імпульси електричної напруги. Якщо синхронний режим скорочення серцевих м'язів порушується, то можуть настати так звані фібриляції – хаотичні спазми окремих волокон серцевого м'язу, які, якщо не прийняти екстрених мір, приводять до загибелі організму. Термінові міри полягають в насильній синхронізації серця за допомогою спеціального масажу або електричних імпульсів від лабораторного генератора. Іноді мініатюрний електронний генератор синхронізуючих імпульсів навіть вживлюють в організм. Синхронізація коливань серцевого м'язу проходить на фізіологічному рівні, в ній головну роль відіграє нервова система.

У житті окремих клітин і колективів клітин коливання і синхронізація цих коливань також є. Різні органели клітини – ядро, рибосоми, мітохондрії – коливаються, змінюючи свою форму і об'єм. Коливання ядра, очевидно, сприяє більш інтенсивному обміну речовинами між ядром і цитоплазмою клітини, зокрема більш інтенсивному обміну молекулами, які несуть генетичну інформацію.

**Висновки.** Таким чином, у статті розглянуто застосування кейс-методу у викладанні вищої математики та показано, що кейс-метод сприяє розвитку вмінь аналізувати ситуацію, оцінювати альтернативи, вибирати оптимальний варіант і складати план його здійснення, формує навички самостійно будувати алгоритми розв'язання задач, виробляє стійкий навик розв'язання практичних задач, формує цікавість і позитивну мотивацію до навчання, підвищує якість і ефективність процесу навчання.

Розроблено приклад науково-дослідного кейсу.

1. Еремін А.С. Обеспечение учебной работы с использованием кейс-метода / А.С. Еремін // Инновации в образовании. – 2010. – № 4. – С. 77–90.
2. Жигилей И.М. Формирование профессиональных компетенций с помощью кейс-метода в высшем образовании / И.М. Жигилей // Преподаватель XXI век. – 2012. – № 1. – С. 29–36.
3. Кейс-метод. Окно в мир ситуационной методики обучения (Case study) [Электронный ресурс] / Управление образовательных и культурных программ Государственного Департамента США, 2007. – Режим доступа: <http://www.casemethod.ru>, своб. – Загл. с экрана.
4. Погребельная Н.И. Кейс-метод как условие формирования исследовательских способностей студентов вуза / Н.И. Погребельная // Наука и школа. – 2008. – № 1. – С. 73–76.
5. Cameron A. The live teaching case: a new IS method and its application / A. Cameron, M. Trudel, R. Titah, P. Leger, P. Blakey // Journal of information Technology Education. – 2012, – Vol. 11. – P. 27–42.
6. Herreid C. Case study teaching / C. Herreid // New Directions for Teaching and Learning. – 2011. – Vol. 2011, No. 128. – P. 31–40.

УДК 658.631.3

Сіваковська О.М., аспірант

Луцький національний технічний університет

## СИСТЕМНІ ЗАСАДИ СТВОРЕННЯ ОРГАНІЗАЦІЙНО-ТЕХНІЧНИХ СИСТЕМ АВТОМАТИЗОВАНОГО УПРАВЛІННЯ ПРОЕКТАМИ ТА ПРОГРАМАМИ

**Сіваковська О.М.** Системні засади створення організаційно-технічних систем автоматизованого управління проектами та програмами. Дано визначення технологічної та організаційно-технічної систем, розкрито їх структурну подібність. Означено системні складові технологічних систем та з'ясовано сутність управління ними на основі команд (розпоряджень), що виробляються організаційно-технічними системами. Концептуально розкрито задачі системного аналізу та синтезу цих систем, основні підстави, що формують вимоги до методів розв'язання управлінських задач.

**Ключові слова:** система, технологічна, організаційно-технічна, управління, проекти, задачі, методи, параметри, конфігурація.

**Сиваковская Е.Н.** Системные основания создания организационно-технических систем автоматизированного управления проектами и программами. Определена технологическая и организационно-техническая системы, раскрыто их структурное сходство. Обозначены системные составляющие технологических систем и выяснена сущность управления ими на основании команд (распоряжений), которые вырабатываются организационно-техническими системами. Концептуально раскрыты задачи системного анализа и синтеза этих систем, основания, которые формируют требования к методам решения управленческих задач.

**Ключевые слова:** система, технологическая, организационно-техническая, управление, проекты, задачи, методы, параметры, конфигурация.

**Sivakovska O.** System bases of creating organising technical systems for automation control projects and programs. There have been proposed a definition of a technological system and an organizing technical system. Their structural resemblance has been defined too. The system components of the technological systems have been designated. The conclusions about the essence of their control with the help of commands (directions) produced by the organizing technical systems, have been made. There have been decided the tasks of the systems analysis and synthesis for these systems. And there have been decided the main bases that form requirements for problem-solving methods of the management tasks too.

**Keywords:** system, technological system, organizing technical system, management, projects, tasks, methods, parameters, configuration.

**Постановка наукової проблеми.** З кожним роком в усіх галузях виробництва розробляються та створюються різного виду автоматизовані системи управління, які забезпечують підвищення їх результативності. Однак для багато галузей, підприємств та виробничих формувань таких систем ще не створено. А тому їх продукти та послуги є ще недостатньо конкурентоспроможними. Зокрема, це стосується сільськогосподарського виробництва, яке в Україні належить до пріоритетних галузей економіки. Не створено ще також якісних автоматизованих систем управління проектами та програмами, яке застосовується сьогодні в усіх сферах людської діяльності. Таким чином як у світі, так і в Україні існує проблема застосування у багатьох галузях економічної діяльності автоматизованих систем управління програмами та проектами. Ключовим моментом вирішення цієї проблеми є обґрунтування конфігурації таких систем, яка визначається прикладною сферою їх застосування.

**Аналіз останніх публікацій та досліджень.** У свій час питання створення автоматизованих систем проектування та управління були стандартизовані [1]. У чинних стандартах розписано порядок проектування та створення відповідних систем. Однак передбачений ними етап науково-дослідних робіт детально не розписано як стосовно розв'язуваних управлінських задач, так і обґрунтування конфігурації. Відомі наукові розробки стосовно створення інформаційно-аналітичних систем [2], також не розкривають завдання їх дослідження та обґрунтування конфігурації.

**Метою статті** є розкриття системних засад створення автоматизованих організаційно-технічних систем.

**Методи дослідження.** У дослідженні використано методи індукції та дедукції, системного аналізу та синтезу, аналогій та моделювання.

**Виклад матеріалу.** Організаційно-технічною системою (ОТС) будемо називати систему, що складається з виконавців (управлінців), інформаційно-технічних засобів, персональних комп'ютерів (ПК) та програмного забезпечення. Ця система має використовувати автоматизовані

системи управління (АСУ), а також системи підтримки до прийняття рішень (СППР). Тобто складовими ОТС є АСУ та СППР. Не вдаючись на початковому етапі дослідження до деталізації ОТС, зазначимо, що її структурні складові мають обґрунтуватися на основі системного підходу. Розкриємо його сутність. Організаційно-технічні системи не можна досліджувати ізольовано від технологічних систем (ТС), які є об'єктами управління. У ТС відбуваються технологічні процеси, які також є об'єктами управління. Технологічними системами будемо називати системи, що складаються з предметів праці (речових об'єктів), технічних засобів та виконавців. [3]. Проекти та програми є частковим видом ТС. За допомогою технічних засобів відповідно до чинних регламентів (технологій) здійснюють якісні перетворення предметів праці – із заданого у бажаний стан. Ці перетворення відбуваються у певних виробничих умовах. З огляду на викладене, ТС можна відобразити еліпсоїдною діаграмою, початком (ядром) якої є предмет праці ( $I$ ) (рис. 1).

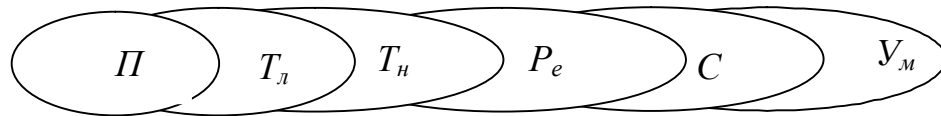


Рис. 1. Відображення технологічної системи:  $I$ ,  $T_{л}$ ,  $T_{н}$  – відповідно предмет праці, технологія та техніка;  $P_{е}$ ,  $C$ ,  $U_{м}$  – відповідно енергетичні ресурси, виконавці та виробничі умови

Функціонування ТС не можливе без управління ними. Для управління ТС, як уже згадувалося, створюються ОТС. В ОТС є також предмети праці – це інформація про стан складових ТС. Ця інформація перетворюється з одного якісного стану в інший на основі відповідних алгоритмів (регламентів), які, як і технології ( $T_{л}$ ) у ТС, відіграють роль знань про її якісні перетворення. Перетворення інформації здійснюється за допомогою ПК, а її збір та передача – за допомогою інформаційно-технічних засобів (ІТЗ). Для функціонування ПК потрібні енергетичні ресурси ( $P_{е}$ ), а також відповідні умови ( $U_{м}$ ). Таким чином, ОТС можна також відобразити у вигляді еліпсоїдної діаграми (рис. 2). У ній інформація ( $I$ ) займає центральне місце. На основі певних алгоритмів за допомогою ПК вони перетворюються до вигляду, зручного для користування управліннями ( $C$ ).

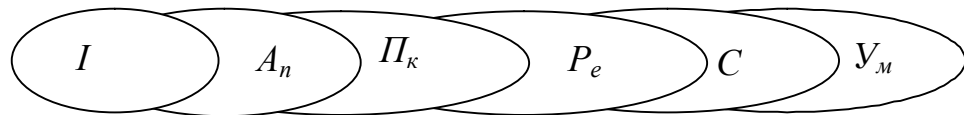


Рис. 2. Відображення організаційно технічної системи:  $I$ ,  $A_{н}$ ,  $П_{к}$  – відповідна інформація, алгоритм (програма) її обробки та персональний комп'ютер;  $P_{е}$ ,  $C$ ,  $U_{м}$  – відповідно енергетичні ресурси, управління та умови праці Діаграмне відображення двох зазначених систем дає змогу з'ясувати їх структурну подібність, що є підставою спільних методологічних засад їх дослідження (моделювання). Ці засади стосуються як застосування аналогій у моделюванні кожної системи, так і їх системного (комплексного) наукового розгляду. Системне дослідження ТС та ОТС виконується тому, що між ними існують причинно-наслідкові зв'язки, без розкриття яких неможливо змодельовати та обґрунтувати раціональні параметри (конфігурацію) як ТС, так і ОТС. У цьому контексті зауважимо, що ТС виступає у ролі первинної, а ОТС – похідної. Розглянемо ці дві системи у системному взаємозв'язку (рис.3).

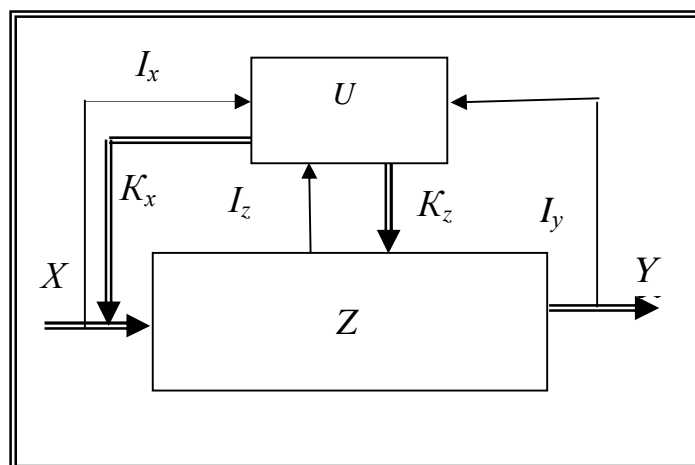




Рис.3. Схема системного розгляду технологічної та організаційно-технічної систем:  $Z, U$  – відповідно параметри технологічної та організаційно-технічної системи;  $X, Y$  – відповідно характеристик вхідного потоку та показники вихідного потоку;  $I_x, I_z, I_y$  - відповідно вхідна інформація про характеристики, параметри та показники технологічної системи;  $K_x, K_z$  - відповідно команди стосовно зміни вхідного потоку та параметрів технологічної системи.

Особливостями цього взаємозв'язку є те, що показники у вихідного потоку ( $Y$ ) продукції (продукту) визначаються як характеристиками ( $X$ ) вхідного потоку предметів праці, так і параметрами ( $Z$ ) технологічної підсистеми їх якісного перетворення [4]:

$$Y = f(X, Z, T) \quad (1)$$

де  $T$  - час функціонування технологічної системи.

Характеристики ( $X$ ) вхідного потоку предметів праці та параметри ( $Z$ ) технологічної підсистеми певною мірою визначаються змістом відповідних команд  $K_x, K_z$ :

$$\begin{aligned} X &= f(X_n, \Delta X); \\ Z &= f(Z_n, \Delta Z), \end{aligned} \quad (2, 3)$$

де  $X_n, Z_n$  - відповідно попередні значення характеристик вхідних потоків предметів праці та параметрів технологічної підсистеми їх якісного перетворення;  $\Delta X, \Delta Z$  - відповідно зміни вхідного потоку предметів праці та параметрів технологічної підсистеми.

Зміни  $\Delta X$  та  $\Delta Z$  визначаються змістом відповідних команд:

$$\Delta X = f(K_x); \Delta Z = f(K_z). \quad (4)$$

Розкриємо основні підстави доцільних змін характеристик  $\Delta X$  вхідних потоків предметів праці та параметрів  $\Delta Z$  технологічної підсистеми їх якісного перетворення. Ці підстави можна розділити на безпосередні та опосередковані. До безпосередніх підстав відносяться появи в часі небажаних (негативних для функціонування ТС) відхилень або характеристик ( $\delta X$ ) вхідного потоку, або ж параметрів ( $\delta Z$ ) технологічної підсистеми. До опосередкованих підстав (причин) необхідності змін  $\Delta X$  та  $\Delta Z$  відноситься поява у часі небажаних відхилень показників  $\delta Y$  вихідного потоку продуктів. Для виявлення відхилень (небажаних змін)  $\delta X, \delta Z$  та  $\delta Y$  здійснюється моніторинг основних системних складових ( $X, Z, Y$ ) ТС та передача відповідної інформації ( $I_x, I_z, I_y$ ) до ОТС. А тому можна записати відповідні зв'язки між ТС і ОТС:

$$I_{\delta X} = f(\delta X); I_{\delta Z} = f(\delta Z); I_{\delta Y} = f(\delta Y). \quad (5)$$

де  $I_{\delta X}, I_{\delta Z}, I_{\delta Y}$  - відповідно потоки інформації про відхилення  $X, Z$  та  $Y$ .

Інформаційні зв'язки стосовно основних системних складових ( $X, Z, Y$ ) ТС забезпечуються моніторингом їх стану та передачею відповідної інформації до ОТС. Продукування команд (розпоряджень)  $K_x$  і

$K_z$  відбувається на основі перетворення цієї інформації. Ці перетворення, як уже згадувалося, здійснюються за певними алгоритмами ( $A_l$ ). У цьому разі можемо записати відповідну дію

$$A_l(X): K_x = f^x(I_{\delta x}, I_{\delta z}, I_{\delta y}); \quad (6)$$

$$A_l(Z): K_z = f(I_{\delta x}, I_{\delta z}, I_{\delta y}). \quad (7)$$

Залежність змісту команд  $K_x$  і  $K_z$  від змісту наявної інформації про стани усіх трьох системних складових є умовою достатньою, однак не завжди необхідною. Ці умови визначаються сутністю алгоритму ( $A_l$ ), в основі яких лежать методи розв'язання (вирішення, обґрунтування)

управлінських задач (завдань). Водночас ці методи ( $M$ ) визначаються змістом ( $Z_3$ ) відповідних задач (завдань). З огляду на це, можемо записати відповідний причинно-наслідковий зв'язок:

$$A_i = f(M); M = f(Z_3) \quad (8)$$

Окрім того зміст команд (розпоряджень)  $K_x$  та  $K_z$  не лише визначається наявністю інформації ( $I_x, I_z, I_y$ ), але й результатами розв'язання (вирішення, обґрунтування) відповідних задач (завдань).

Розглянемо більш детально питання розв'язання управлінських задач. У цьому контексті зауважуємо, що управлінські задачі поділяються на стратегічні, тактичні та оперативні. Стратегічні задачі розв'язуються з метою створення або ж розвитку ТС. Тобто ці задачі можуть розв'язуватися коли ТС ще не існує, коли у наявних ТС змінюються масштаби, або ж технології якісного перетворення предметів праці. Тактичні та оперативні задачі розв'язуються з метою підтримання ефективного функціонування чинних ТС. Джерелом інформації для розв'язання задач стратегічного управління визначаються предметною галуззю. Наприклад, у галузі сільськогосподарського виробництва джерелами такої інформації є офіційні дані державних органів статистики, результати багаторічних спостережень агрометеорологічних станцій, офіційні дані земельного кадастру, каталоги та проекти фірм-виробників сільськогосподарської техніки тощо. За тактичного та оперативного управління цим виробництвом джерелами інформації є складові ТС – поля, вирощений урожай, техніка, люди тощо. Дані поточного моніторингу стану полів, посіяних на них культур, роботоздатності наявної техніки, функціональної готовності механізаторів, Інтернет-прогнозу погоди тощо – це та інформація, яка отримується зі згаданих джерел. Отримання відповідної інформації забезпечується наявністю в ОТС підсистем моніторингу, а також наявності звітності про функціонування ТС.

Обсяги вхідної інформації та періодичність її надходження до ОТС визначаються предметною галуззю, змістом управлінських задач та методами їх розв'язання. Тому основною задачею аналізу ОТС є задача визначення характеристик ( $I_x, I_z, I_y$ ) вхідної інформації за заданого змісту розпоряджень (команд):

$$(I_x, I_z, I_y) = f(Z, M), \text{ за умови } (K_x, K_z) = const. \quad (9)$$

Водночас характеристики ( $I_x, I_z, I_y$ ) вхідних потоків потрібної інформації визначають параметри ( $U$ ) підсистем їх перетворення:

$$U = f(I_x, I_z, I_y), \text{ за умови } (K_x, K_z) = const. \quad (10)$$

Цей вираз відображає системне розв'язання іншої основної задачі аналізу ОТС. Задача синтезу ОТС записується таким чином:

$$(K_x, K_z) = f(U, I_x, I_z, I_y). \quad (11)$$

Означені вирази (9,10,11) задач аналізу та синтезу ОТС є важливою методологічною засадою обґрунтування їх конфігурації (параметрів) ( $U$ ) та створення ОТС у заданій прикладній галузі виробництва. Зокрема, стосовно сільськогосподарського виробництва дві основні задачі аналізу розв'язуються системно на основі використання методу ітерацій. У цьому випадку насамперед формулюються управлінські задачі стосовно як характеристик ( $X$ ) вхідного потоку предметів праці, так і параметрів ( $Z$ ) технологічної підсистеми (складової) їх якісного перетворення. Ці задачі, як уже згадувалося, поділяються на стратегічні, тактичні та оперативні. Методи їх розв'язання лежать в основі вимог до вхідної інформації. Водночас, основною вимогою до методів розв'язання управлінських задач є отримання вірогідних (достовірних) результатів. Не вдаючись до поглибленої класифікації управлінських задач проектів (програм) сільськогосподарського виробництва, обґрунтуємо основні вимоги до методів їх розв'язання, які займають центральне місце в ОТС.

Задачі з управління проектами (програмами) сільськогосподарського виробництва стосуються усіх складових проектного управління: змісту та часу, інтеграції, людських та матеріальних ресурсів, закупівель, ризиків, інформації, конфігурації тощо [5]. Складність

проектного управління цим виробництвом зумовлюється нестабільністю агрометеорологічних умов [6], які значною мірою визначають особливості часового перебігу відповідних проектів. Зокрема, ці умови визначають час та зміст виконання механізованих робіт у відповідних проектах, а також є причиною виникнення перерв під час виконання робіт. З огляду на це, методи розв'язання задач з управління сільськогосподарськими проектами мають враховувати стохастичний вплив агрометеорологічних умов на їх перебіг. З огляду на це, використовують метод статистичного імітаційного моделювання [7]. Суть цього методу полягає в тому, що моделювання проектів (процесів) ТС сільськогосподарського виробництва виконують з відображенням стохастичної дії агрометеорологічних умов. Цю дію досліджують на основі емпіричної інформації агрометеорологічних станцій стосовно появи на календарній осі часу відповідних проектних подій – часу початку відновлення весняної вегетації озимих культур, часу досягнення фізичної стиглості ґрунту, часу прогрівання ґрунту для сівби холодостійких та теплолюбних культур, часу виникнення шкочинних об'єктів на сільськогосподарських культурах, погожих та непогожих проміжків часу, часу початку досягання вирощеного врожаю, часу початку біологічних втрат цього врожаю через несвоєчасність його збирання тощо. Отримані статистичні закономірності (теоретичні закони розподілів та оцінки їх статистичних характеристик) зазначених подій кладуться в основу моделювання відповідних проектів сільськогосподарських ТС. Для цього використовується метод Монте-Карло, який дає змогу розкрити стохастичну дію агрометеорологічних умов у цих проектах [7].

Особливістю статистичного імітаційного моделювання проектів ТС сільськогосподарського виробництва є те, що моделювання кожного з них виконують циклічно – від початку до завершення проекту (календарної тривалості сезонів виконання певних робіт). Число циклів (реалізацій) того чи іншого проекту обґрунтовується таким чином, щоб сукупно (системно) врахувати стохастичну зміну усіх подій. Це досягається завдяки планування комп'ютерних експериментів над статистичними імітаційними моделями.

**Висновки.** 1. Створення організаційно-технічних систем автоматизованого управління проектами та програмами має базуватися на результатах дослідження як об'єктів управління (технологічних систем), так і управляючих (організаційно-технічних) систем. 2. В основі цих досліджень лежать взаємозв'язки між технологічними та організаційно-технічними системами, які визначаються їх функціональним призначенням та міжсистемними взаємодіями. 3. Системне розкриття взаємодії між технологічними та організаційно-технічними системами дали змогу означити між системні зв'язки, котрі визначаються об'єктами управління та змістом управлінських задач. 4. Дослідження організаційно-технічних систем передбачає розв'язання задач їх системного аналізу та синтезу, а обґрунтування їх параметрів (конфігурації) базується на результатах цих досліджень. 5. Розв'язання задач з управління проектами ТС вимагає їх моделювання, яке для проектів ТС сільськогосподарського виробництва, що характеризується стохастичною дією агрометеорологічних умов, має базуватися на методі Монте-Карло.

1. ГОСТ 34.601-90 [Чинний від 1992-01-01] – Комплекс стандартів на автоматизированные системы.
2. Кисіль Н. М. Класифікація інформаційних систем; лісове господарство, лісова, паперова і деревообробна промисловість / Н.М. Кисіль, З.П. Гаталяк, Н.І. Горбаль // Міжвідомчий науково-технічний збірник. Львів: УкрДЛТУ. 2004. Вип. 29. С. 242-249.
3. Сидорчук, О. Інженерія машинних систем: монографія [текст] / О. Сидорчук. – К., 2007. – 263 с.
4. Альянах И.Н. Моделирование вычислительных систем. /И.Н. Альянах/ - Л.: Машиностроение, 1988. - 224 с.
5. Бушуев С. Д. Креативные технологии управления проектами и программами / С. Д. Бушуев, Н. Д. Бушуева, И. А. Бабаев и др./ - К.: «Саммит-Книга», 2010. – 768 с.
6. Сидорчук О.В. Планування механізованих зернозбиральних робіт і проектів [монографія], /О.В.Сидорчук/ - Ніжин: вид.П.П.Лисенко, 2013.-157 с.
7. Шенон Р. Имитационное моделирование систем - искусство и наука /Р. Шенон / - М.: Мир, 1978.- с.424 с.