

CONFERENCE PROCEEDINGS

МАТЕРІАЛИ КОНФЕРЕНЦІЇ

 **summer**  
**infoCom**2018

# Summer InfoCom Advanced Solutions 2018

VI МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ  
20-21 травня 2018 року

ISBN 978-966-2344-65-3

Україна, Київ



**Summer InfoCom  
Advanced Solutions 2018**

VI МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ  
20-21 травня 2018 року

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ІНСТИТУТ МОДЕРНІЗАЦІЇ ЗМІСТУ ОСВІТИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ  
УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ  
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ  
СІКОРСЬКОГО»**

**SUMMER  
INFOCOM  
ADVANCED  
SOLUTIONS 2018**

**МАТЕРІАЛИ**

**VI МІЖНАРОДНОЇ НАУКОВО-ПРАКТИЧНОЇ  
КОНФЕРЕНЦІЇ**

**З ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

**CONFERENCE PROCEEDINGS**

**6th SCIENTIFIC AND PRACTICAL CONFERENCE**

**КИЇВ, УКРАЇНА**

**20-21 ТРАВНЯ 2018 РОКУ**

## **УДК 004**

### *Редакційна колегія:*

Бідюк П.І., д.т.н., проф., ІПСА, КПІ ім. Ігоря Сікорського, Україна, Київ

Павлов О.А., д.т.н., проф., КПІ ім. Ігоря Сікорського, Україна, Київ

Теленик С.Ф., д.т.н., проф., КПІ ім. Ігоря Сікорського, Україна, Київ

Грішин І.Ю., д.т.н., проф., Кубанський державний технологічний університет, Російська Федерація

### *Головний редактор:*

Писаренко А.В., к.т.н., доц., КПІ ім. Ігоря Сікорського, Україна, Київ

Summer InfoCom 2018: Матеріали VI Міжнародної науково-практичної конференції з інформаційних систем та технологій, м. Київ, 20-21 травня 2018 р. – К.: Вид-во ТОВ "Інжиніринг", 2018. – 82с. – Мови укр., рос., англ.

Конференція входить до Переліку міжнародних та всеукраїнських науково-практичних конференцій здобувачів вищої освіти та молодих учених у 2018 році.

Проведення конференції регламентоване наказом ректора КПІ ім. Ігоря Сікорського № 3/204 від 20 квітня 2018 р.

Усі права застережено. Передруки та переклади дозволяються лише за згодою автора та редакції. За достовірність фактів, цитат, назв та іншої інформації несуть відповідальність автори.

Редакційна колегія дотримується прийнятих міжнародною спільнотою принципів публікаційної етики, відображених, зокрема, в рекомендаціях Комітету з етики наукових публікацій (Committee on Publication Ethics, COPE), а також враховує досвід авторитетних міжнародних видавництв. Щоб уникнути недобросовісної практики в публікаційній діяльності (плагіат, виклад недостовірних відомостей та ін.), з метою забезпечення високої якості наукових публікацій, визнання громадськістю отриманих автором наукових результатів, кожен член редакційної колегії, автор, рецензент, видавець, а також установи, які беруть участь в видавничому процесі, зобов'язані дотримуватися етичних стандартів, норм і правил та вживати всіх можливих заходів для запобігання їх порушень. Дотримання правил етики наукових публікацій усіма учасниками цього процесу сприяє забезпеченню прав авторів на інтелектуальну власність, підвищенню якості видання і виключення можливості неправомірного використання авторських матеріалів в інтересах окремих осіб.

**ISBN 978-966-2344-65-3**

# **ПРОГРАМА КОНФЕРЕНЦІЇ**



# ПРОГРАМА / PROGRAM

## *Інформаційні системи та технології*

**Самотий В.  
Дзелендзяк У.  
Палюшок В.** Інтерактивний веб-додаток з використанням хмарного сховища даних

**Попко А.В.  
Дорогий Я.Ю.** Визначення образ у коротких текстах

**Уалханова А.Т.  
Денисова Н.Ф.  
Крак Ю.В.** Разработка технологии 3D-моделирования геологических объектов

**Харабет Р.И.  
Дорогий Я.Ю.** Использование Amazon Connect для автоматизации колл-центра

**Писаренко О.А.** Аналіз алгоритмів оптимізації обчислення локального мінімуму функції Розенброка

**Моргаль О.М.  
Шихутський С.О.** Автоматизація інструментального дослідження параметрів кабельних ліній

**Боярин О.В.  
Дячук І.С.  
Дорогий Я.Ю.** Чатбот на основі нейронних мереж

**Podrubailo O.  
Saverchenko V.** Index structure optimizations for in-memory data storage systems

## *Системи керування*

**Майєр І.В.  
Писаренко А.В.** Алгоритм покоординатного спуску в методі динамічного програмування для задач цифрового оптимального керування

**Нагайко Д.Ю.  
Репнікова Н.Б.** Удосконалена формула розрахунку коефіцієнтів ПД-регуляторів для синтезу складних систем

## *Технології програмування*

**Кравінський В.О.** UML діаграми для програмного комплексу дистанційного управління рухомими об'єктами

## *Безпека та захист інформації*

**Радецький Т.С.** Розподіл ключів

20 травня

## **Інформаційні системи та технології**

<b>Дорогий Я.Ю. Левченко К.В.</b>	Порівняння фреймворків машинного навчання
<b>Стенин А.А. Пасько В.П. Лемешко В.А. Русакова А.В.</b>	Ситуационное управление городским транспортом с интеллектуальной поддержкой диспетчерских решений
<b>Огульчанський Ф.Я. Репнікова Н.Б.</b>	Розширення області використання двигунів Стірлінга для підтримання альтернативного енергопостачання житла
<b>Юрчук А.Ю.</b>	Проектування діяльності користувача людино-машинного інтерфейсу АСУ ТП
<b>Кицмен Д.Р.</b>	Неблокуюча база даних з ACID транзакціями
<b>Бердник Ю.М.</b>	Реалізація та дослідження штучного нейрону на ПЛІС різних сімейств
<b>Антосєв А.С. Дорогий Я.Ю.</b>	Дослідження впливу колірних моделей при визначенні кольорових характеристик об'єктів розпізнавання на зображенні
<b>Савчук О.В. Казанцева А.В. Пирожков О.Ю.</b>	Про вибір моделі надійного функціонування програмного забезпечення у хмарних технологіях

**21 травня**

## **Системи керування**

<b>Хлівненко М.В.</b>	Модель системи виявлення пішоходів та їх трекінгу з використанням рухомого джерела світла
<b>Якимчук В.В. Репнікова Н.Б.</b>	Керування складськими роботами на базі алгоритму синтезу з бажаними коренями цифрової системи

## **Підготовка кадрів у галузі інформаційних технологій**

<b>Гришин І.Ю. Тимиргалеева Р.Р.</b>	Особенности формирования программ подготовки магистров информационной безопасности
<b>Драбинко В.П. Жеребко В.А.</b>	Моделювання типових технологічних процесів засобами CoDeSys



# ЗМІСТ / CONTENTS

<b>Інформаційні системи та технології / Information Systems and Technologies.....</b>	9
<b>Самотий В., Дзелендзяк У., Палюшок В.</b>	
Інтерактивний веб-додаток з використанням хмарного сховища даних.....	11
<b>Попко А.В., Дорогий Я.Ю.</b>	
Визначення образ у коротких текстах .....	13
<b>Уалханова А.Т., Денисова Н.Ф., Крак Ю.В.</b>	
Разработка технологии 3D-моделирования геологических объектов...	15
<b>Харабет Р.И., Дорогий Я.Ю.</b>	
Использование Amazon Connect для автоматизации колл-центра.....	17
<b>Писаренко О.А.</b>	
Аналіз алгоритмів оптимізації обчислення функції Розенброка.....	19
<b>Моргаль О.М., Шихутський С.О.</b>	
Автоматизація інструментального дослідження параметрів кабельних ліній.....	21
<b>Боярин О.В., Дячук І.С., Дорогий Я.Ю.</b>	
Чатбот на основі нейронних мереж .....	23
<b>Podrubailo O., Saverchenko V.</b>	
Index structure optimizations for in-memory data storage systems.....	26
<b>Дорогий Я.Ю., Левченко К.В.</b>	
Порівняння фреймворків машинного навчання.....	29
<b>Стенин А.А., Пасько В.П., Лемешко В.А., Русакова А.В.</b>	
Ситуационное управление городским транспортом с интеллектуальной поддержкой диспетчерских решений.....	32
<b>Огульчанський Ф.Я., Репнікова Н.Б.</b>	
Розширення області використання двигунів Стірлінга для підтримання альтернативного енергопостачання житла.....	34
<b>Юрчук А.Ю.</b>	
Проектування діяльності користувача людино-машинного інтерфейсу АСУ ТП.....	36
<b>Кицмен Д.Р.</b>	
Неблокуюча база даних з ACID транзакціями.....	38
<b>Бердник Ю.М.</b>	
Реалізація та дослідження штучного нейрону на ПЛІС різних сімейств.....	40
<b>Антосєв А.С., Дорогий Я.Ю.</b>	
Дослідження впливу колірних моделей при визначенні кольорових характеристик об'єктів розпізнавання на зображенні.....	42



<b>Савчук О.В., Казанцева А.В., Пирожков О.Ю.</b>	
Про вибір моделі надійного функціонування програмного забезпечення у хмарних технологіях.....	44
<b>Системи керування / Control Systems.....</b>	47
<b>Майер І.В., Писаренко А.В.</b>	
Алгоритм покоординатного спуску в методі динамічного програмування для задач цифрового оптимального керування .....	49
<b>Нагайко Д.Ю., Репнікова Н.Б.</b>	
Удосконалена формула розрахунку коефіцієнтів ПД-регуляторів для синтезу складних систем.....	51
<b>Хлівненко М.В.</b>	
Модель системи виявлення пішоходів та їх трекінгу з використанням рухомого джерела світла.....	53
<b>Якимчук В.В., Репнікова Н.Б.</b>	
Керування складськими роботами на базі алгоритму синтезу з бажаними коренями цифрової системи.....	55
<b>Технології програмування / Programming technologies.....</b>	57
<b>Кравінський В.О.</b>	
UML діаграми для програмного комплексу дистанційного управління рухомими об'єктами.....	59
<b>Безпека та захист інформації / Information security.....</b>	61
<b>Радецький Т.С.</b>	
Розподіл ключів.....	63
<b>Підготовка кадрів у галузі інформаційних технологій/ Training in information technologies field.....</b>	67
<b>Гришин І.Ю., Тимиргалеева Р.Р.</b>	
Особенності формування програм подготовки магистров информационной безопасности.....	69
<b>Драбинко В.П., Жеребко В.А.</b>	
Моделювання типових технологічних процесів засобами CoDeSys.....	71
<b>Abstracts.....</b>	73

**ІНФОРМАЦІЙНІ СИСТЕМИ ТА  
ТЕХНОЛОГІЇ**

**INFORMATION SYSTEMS AND  
TECHNOLOGIES**



# Інтерактивний веб-додаток з використанням хмарного сховища даних

Самотий Володимир

Краківська політехніка

Краків, Польща

Львівський державний університет безпеки

життєдіяльності

Львів, Україна

vsamotyj@gmail.com

Дзелендзяк Уляна, Палюшок Віталій

Національний університет «Львівська політехніка»

Львів, Україна

u.dzelendzyak@gmail.com

**Анотація.** Розроблено інтерактивну систему «Притулок», яка складається з бази даних, сховища даних, прогресивного мобільного додатку та основного веб інтерфейсу. Система дає можливість створювати, відправляти та опрацьовувати заявки користувачів, а також переглядати, змінювати, видаляти та переносити елементи системи, управляти додатковими функціями додатку, що є недоступними для звичайного користувача.

**Ключові слова:** нереляційна база даних, сховище даних, прогресивний мобільний додаток, веб інтерфейс, хмарні технології.

## ВСТУП

Інтерактивні веб-додатки є програмними системами та продуктами, які виконують операції, процедури та дії, під які вони адаптовані: починаючи з простого відображення даних і закінчуючи складними інтерактивними системами обміну даними та системами штучного інтелекту. При цьому вони можуть активно взаємодіяти з користувачем, широко застосовуються у всіх сферах надання послуг, управління інформацією і ресурсами. Знаходять застосування в інтернет банкінгу та адмініструванні даних. Оскільки, в наш час більшість задач виконуються через інтернет, то такі системи мають великий попит.

## СТРУКТУРА ІНТЕРАКТИВНОГО ВЕБ-ДОДАТКУ

Для розроблення інтерактивної системи «Притулок», яка складається з бази даних, сховища даних, прогресивного мобільного додатку та основного веб інтерфейсу було використано Google Firebase [1]. База даних у Firebase - це БД класу NoSQL (нереляційна), яка дає можливість зберігати і синхронізувати дані між користувачами в реальному часі. Типом бази даних є сховище "ключ-значення". Це сховище, яке використовує ключ для отримання доступу до значення. Такий підхід знімає всі обмеження реляційної моделі (недостатня продуктивність, трудомістке горизонтальне масштабування) і полегшує можливості збереження і доступу до даних. Такі дані можуть представляти собою одиночний об'єкт JSON і разом з тим коректно відповідати на запити по полях.

У базі даних використовується неструктурований підхід, тобто структура створюється «на льоту», тим самим знімаючи обмеження жорстких зв'язків і

пропонуючи різні типи доступу до специфічних значень.

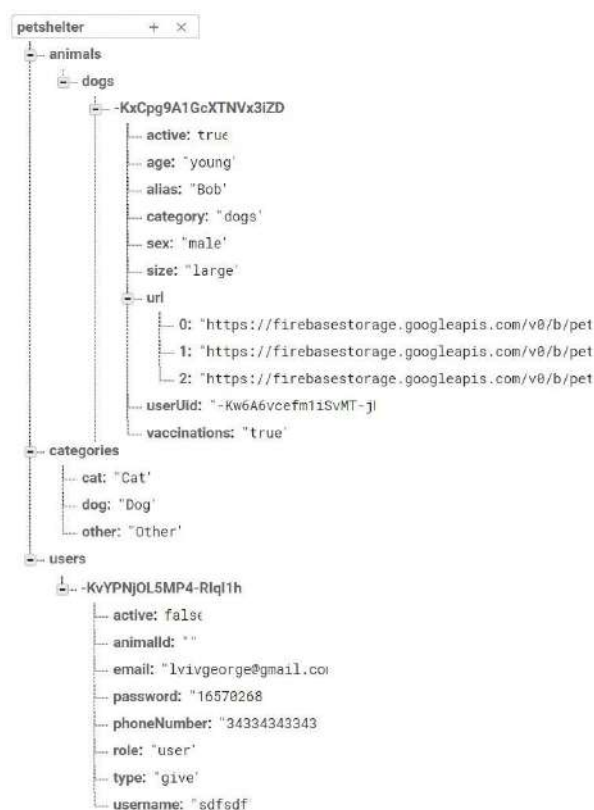


Рис. 1. Структура бази даних системи Firebase

База даних системи Firebase та її структура наведена на рис. 1. Розроблена база являє собою JSON об'єкт, який складається з ключів-значень. Наприклад, Animals містить об'єкт dogs, що складається з об'єктів, ключами яких є унікальні ідентифікатори, що генеруються базою при додаванні нового об'єкта. Значеннями кожного з ключів є властивості, які містять інформацію відповідного змісту. Однією з властивостей є Url, що містить посилання на місце збереження фото у сховищі даних Firebase.

У розробленій системі використовується сховище даних Firebase Cloud Storage. Хмарні технології надають доступ до особистих файлів з будь-якого комп'ютера, що має доступ в Інтернет. Ця технологія дозволяє значно ефективніше керувати, обробляти та працювати з інформацією за рахунок централізації цієї самої інформації. Сховище даних (рис. 2) – це об'єкт з даними, який

```

▼ {animals: {...}, categories: {...}}
  ▼ animals:
    ▼ cats:
      ▼ -Kw0808sEox-rzA2j06R:
        0: "0sfjkgd.jpg"
        1: "1fdgteov.jpg"
      ► dogs: {0: "..."}
      ► other: {1: "..."}
    ▼ categories:
      ► cat: {0: "..."}
      ► dog: {1: "..."}
      ► other: {2: "..."}

```

Рис. 2. Сховище даних

містить Animals та Categories. Кожна з категорій має дочірні підкатегорії зі значеннями, які можна отримати за присвоєними унікальними ключами. Схожим чином можна записати нові об'єкти та додати їм властивості, які будуть доступні за новими згенерованими ключами.

Іншою важливою особливістю Firebase є сервіс авторизації (Authentication), який дає змогу створювати аккаунти, використовуючи різні сервіси, авторизуватись у додатку та управляти аккаунтами. Також у Firebase Authentication є методи, які дозволяють авторизовувати користувача, надсилаючи повідомлення на його телефон. Крім того, даний сервіс може керувати надсиланням повідомлень із скиданням паролю на пошту користувача. Це розширює можливості та покращує умови реєстрації для користувачів, оскільки кожен може вибирати найзручніший спосіб для себе. Реєстрація є швидкою та безпечною. Наступним кроком йде авторизація користувача в додатку. Цей процес теж захищений. Після введення логіну та паролю користувачем, відбувається запит до серверу. Якщо той знаходить у себе такі дані (логін

та пароль), то повертає одноразовий, зашифрований код (token), завдяки якому користувач потрапляє в додаток і може користуватись усіма доступними для його аккаунту функціями.

Система “Притулок” включає в себе прогресивний мобільний додаток (PWA), який дає можливість сайту взаємодіяти з користувачем як додаток [2]. Тобто користувач може встановлювати його на будь-який гаджет, отримувати сповіщення і працювати з ним, навіть без інтернету. PWA додаток пропонуватиме встановити його на ваш пристрій вже після другого візиту на сайт. Перевагою PWA є можливість запускати його, використовуючи той самий код, що й для сайту і не потрібно створювати окремий нативний додаток.

Іншою важливою особливістю системи є можливість роботи offline [3]. При першому вході на сайт чи додаток дані зберігаються та кешуються, щоб бути доступними для використання при відсутності інтернету. PWA підтримується і продовжує імплементуватись усіма браузерами. Він є доступний для становлення на усіх пристроях крім Apple. При цьому, навіть без встановлення, PWA може виконувати свої функції так само, як на сайті.

#### ВИСНОВКИ

Розроблений інтерактивний веб-додаток можна використовувати у різних сферах залежно від заповнення бази даних. Крім того, він є дуже гнучким, оскільки може працювати та швидко підлаштовуватись під різні операційні системи: десктопні (Windows, Linux, MacOS) та мобільні (IOS, Android).

#### ЛІТЕРАТУРА

1. Housseem Yahiaoui. *Firestore Cookbook: Over 70 recipes to help you create real-time web and mobile applications with Firestore.* – 1st Edition / Housseem Yahiaoui. – Birmingham: Packt Publishing, 2017. – 288p.
2. Tal Ater, *Building Progressive Web Apps: Bringing the Power of Native to the Browser.* / Tal Ater – Sebastopol: "O'Reilly Media, Inc.", 2017. – 288 p.
3. Алекс Бэнкс, Ева Порселло. *React и Redux. Функциональная веб-разработка.* / А. Бенкс, Е. Порселло. – Санкт-Петербург: Питер, 2018. – 336 с.

# Визначення образ у коротких текстах

Попко Андрій

КПІ ім. Ігоря Сікорського  
Київ, Україна  
andrew.popko.j@gmail.com

Дорогий Ярослав

КПІ ім. Ігоря Сікорського  
Київ, Україна  
cisco.rna@gmail.com

**Анотація.** Тези доповіді містять опис вибору математичної моделі для вирішення задачі класифікації коротких текстів за наявності в них образ. В роботі коротко описано дослідження методів їх виявлення. Авторами була запропонована власна модель класифікації з описом вдосконалень, що покращують ефективність її роботи.

**Ключові слова:** класифікація текстів, машинне навчання, методи машинного навчання, логістична регресія.

## ВСТУП

Незліченна кількість публічних обговорень різноманітних тем у просторі інтернету багато в чому змінила способи спілкування з іншими. Незалежно від того, чи мова йде про коментарі до статті з сенсаційною новиною, чи про обговорення конкретної відеогри на форумі, сучасні онлайн-простори дозволяють нам легко поділитися власними думками та враженнями, а також отримати теж саме від інших. На зважаючи на те, що часто ці обговорення можуть бути результативними та конструктивними, відносна анонімність, яка приховується в акаунтах користувачів, дозволяє людям залишати образливі та/або неприйнятні коментарі. Ці публікації часто можуть створювати негативне та небажане середовище для інших учасників, що може навіть перешкоджати їм відвідувати сайт. Дана проблема являється достатньо серйозною для адміністрації та власників веб-сайтів.

Одним з потенційно можливих способів пом'якшення цієї проблеми, окрім ручного адміністрування, є створення системи, яка може визначати чи являється кожен вхідний коментар образливим чи ні. За наявності такої власники веб-сайтів матимуть достатньо непогану гнучкість у вирішенні сформульованої вище проблеми. Наприклад, можна вибрати автоматичне блокування точно визначених «образливих» коментарів та приховувати ті, що під сумнівом у системи, для подальшої обробки модератором сайту. Мета цієї роботи полягає в побудові системи, реалізованої на основі методів машинного навчання, яка може класифікувати короткі тексти за ознакою образливості та/чи емоційного забарвлення.

## ПОШУК РІШЕННЯ

У вигляді даних для навчання майбутньої системи та тестування було обрано набір з сайту Kaggle [1], який є найпопулярнішим ресурсом та площадкою, де проводяться змагання з машинного навчання. Набір даних містить близько 4000 прикладів, кожний з яких складається з реального тексту, взятого з просторів інтернету та позначки, що свідчить про його забарвленість (1 – образливий, 0 –

звичайний). Приблизне співвідношення класифікованих текстів 3/8 відповідно.

Основними показниками оцінки, які використані, були точність тренування (training accuracy) та перехресна перевірка (10-fold cross-validation). Перша необхідна для визначення наскільки добре мережа працює з вхідними даними та модель оптимізується на навчальному наборі текстів, і в якій мірі вона «перенавчається» на ньому. Точність другої перевірки була більш значущою, оскільки вона визначає наскільки добре модель узагальнюється для тестування на прикладах, що ще не подавались до системи. Це значення є більш точним уявленням про те, як реалізована модель буде функціонувати у реальній ситуації.

Мовою для кодування було обрано Python, що є найпопулярнішою мовою для реалізації застосувань пов'язаних з неймережами та машинним навчанням. Це дозволило використовувати наявні безкоштовні бібліотеки для підключення, що значно скоротило час на реалізацію алгоритмів. Для виділення характеристик тексту було використано бібліотеку під назвою NLTK (Natural Language Toolkit) [2].

В ході досліджень наявних моделей машинного навчання було визначено що найпопулярнішими є: Naive Bayes (наївний баєсів класифікатор), SVM (метод опорних векторів), and Logistic Regression (логістична регресія) [3] для класифікації текстів. Спочатку було застосовано кожний з підходів з 1-граммами (unigram), утвореними з вхідних текстів, попередньо підданими обробці (видалення знаків пунктуації, приведення літер до нижнього регістру), а потім вибрано найкращий.



Рис. 1. Порівняння базових моделей

Дивлячись на показники для перехресної валідації, очевидно, що модель з логістичною регресією (Logistic Regression) справляється краще ніж дві інші моделі. Тож було вирішено використати даний підхід як базовий для системи класифікації коротких текстів. Також вектор параметрів піддаватиметься навчання, використовуючи метод стохастичного градієнтного спуску, оскільки він достатньо простий в реалізації.

Для того, щоб підвищити точність моделі були додані деякі функції для створюваного

класифікатора. Список тих, що дійсно допомогли покращити показники ідентифікації:

- стемінг (скорочення слів до основи) слів за допомогою NLTK Lancaster Stemmer;
- видалення випадкових символів;
- виділення лексем за допомогою NLTK Tokenizer;
- використання списку «поганих» слів;
- використання буквених 4- та 5-грам.

Процес плинності показника точності для класифікатора можна побачити на рис. 2.

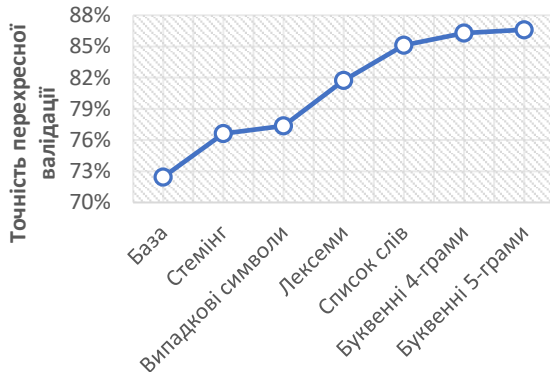


Рис. 2. Покращення моделі в залежності від реалізованих додаткових функцій

Також була спроба замінити використання 1-грам на 2-, 3-, 4-, 5-грами. Крім останніх двох, помітне покращення спостерігалось тільки при використанні останніх двох функцій.

Завдяки додаванню описаних вище функцій до моделі логістичної регресії система досягла точності навчання 100% на повному наборі даних. В той же час система досягає лише приблизно 86% по точності перехресної перевірки. Отже, стає зрозумілим, що модель значно перенавчилась під час тренування. Два методи, які використані для вирішення цієї проблеми, полягають у наступному:

- байсовий апріорний розподіл (Bayesian prior) [4]: замість того, щоб використовувати оцінку максимальної правдоподібності, використано байсову оцінку максимуму апостеріорної імовірності для визначення ваг. Дані підходи тісно пов'язані, але останній застосовує розширену цільову функцію, що включає апріорний розподіл оцінюваної величини. Це сприяє зменшенню норми вектора параметрів, що дозволяє бути менш чутливим до перенавчання.

- Фільтрація вибору ознак: для кожної з ознак призначено значення, що рівне абсолютній величині синаптичної ваги, визначеній методом стохастичного градієнтного спуску. Ці ознаки будуть більш вагомо впливати на результат під час класифікації. Беручи до уваги лише частину з усіх добутих ознак, величина вектору ознак буде зменшена, що також позитивно впливає на ситуацію з перенавчанням.

Після реалізації вище описаних підходів, точність навчання знизилась до 94%, в той час же показник перехресної валідації дещо збільшився, що свідчить про краще узагальнення отриманих знань.

#### ПЕРСПЕКТИВИ РОЗВИТКУ ТА ВИСНОВКИ

Завдяки всім перерахованим вище функціям і методам, що закладені в системі, з різними

маніпуляціями в налаштуваннях, перехресну точність перевірки вдалось підняти до 86,6% від повного набору даних. Хоча вихідні результати цієї системи є перспективними, існує декілька способів, за допомогою яких можна продовжити покращувати результати. Можна використовувати додаткові характеристики. Даний класифікатор розглядає тексти окремо, проте можна розглядати зв'язок між повідомленнями, наприклад, кількість образливих повідомлень у межах одного потоку повідомлень. На нашу думку, в реальному застосуванні ми також можемо включити аналіз історії користувача, наприклад, кількість образливих публікацій, які він написав раніше. Також необхідно проєкспериментувати, наскільки добре будуть виконуватися інші алгоритми оптимізації, наприклад, метод Ньютонна. Ще одним можливим напрямком покращення показників є навчання системи на ідентифікацію сарказмів, коли образа в тексті є прихованою. За допомогою NLP проєктів, таких як, наприклад, «Stanford CoreNLP» можна зрозуміти структуру висловлювання та в певній мірі його зміст, що допоможе покращити результати класифікації.

#### ЛІТЕРАТУРА

1. Kaggle: Your Home for Data Science [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://www.kaggle.com/> (дата звернення 16.04.2018). – Назва з екрана.
2. Natural Language Toolkit. NLTK 3.2.5 documentation [Електронний ресурс] : [Веб-сайт]. – Режим доступу: <https://www.nltk.org/> (дата звернення 16.04.2018). – Назва з екрана.
3. Логистическая регрессия для решения задач классификации [Електронний ресурс] – Режим доступу: <https://goo.gl/uu9DsV>.
4. Математические методы обучения по прецедентам (теория обучения машин) [Електронний ресурс] / Воронцов К. В. // Курс лекций «Машинное обучение» – С. 18-23. – Режим доступу: <https://goo.gl/CoKKBC3>. – Назва з екрана.



# Разработка технологии 3D-моделирования геологических объектов

Уалханова А.Т., Денисова Н.Ф.

Восточно-Казахстанский государственный  
технический университет им. Д. Серикбаева  
Усть-Каменогорск, Казахстан  
AUalkhanova@ektu.kz, NDenisova@ektu.kz

Крак Ю.В.

Киевский национальный университет имени Тараса  
Шевченко  
Киев, Украина  
Yuri.krak@gmail.com, Krak@univ.kiev.ua

**Аннотация.** Статья посвящена исследованию технологий трехмерного моделирования различных геологических объектов, как небольших, так и занимающих обширные территории. В работе исследованы особенности, проблемы и технологии построения 3D-моделей объектов на основе фотограмметрии. Рассмотрены возможности использования программного продукта Agisoft Photoscan, позволяющего в полуавтоматическом режиме получать облако точек и 3D-модели объектов на основе их двумерных фотоснимков.

**Ключевые слова:** фотограмметрия, 3D-модель, 3D-моделирование, облако точек, ортофотоплан.

## ВВЕДЕНИЕ

Восточно-Казахстанская область уникальна по своим туристическим возможностям. Большое разнообразие ландшафтно-климатических зон, наличие памятников истории и культуры обуславливает привлекательность Восточного Казахстана в плане туризма. В связи с этим популяризация геологических знаний и применение для этого современных информационных технологий, в том числе и 3D-визуализации уникальных геологических памятников будет способствовать развитию геотуризма в регионе. Поэтому весьма актуальной задачей является разработка информационно-аналитической системы на основе ГИС-технологий, учитывающих специфику геологических задач и особенности Восточно-Казахстанского региона. Интеграция геоинформационных систем с технологиями 3D-моделирования и 3D-панорамирования позволит визуализировать геологические объекты с привязкой к географическим координатам и объединять данные в единую информационно-аналитическую систему.

Целью данной работы является разработка технологии построения трехмерных моделей геологических объектов, с использованием программного обеспечения, реализующего алгоритмы Structure from motion (Структура из движения) для дальнейшей их интеграции в информационно-аналитическую систему.

## ПОСТРОЕНИЕ ТРЕХМЕРНОЙ МОДЕЛИ ГЕОЛОГИЧЕСКОГО ОБЪЕКТА ПОСРЕДСТВОМ ФОТОГРАММЕТРИИ

Технология, позволяющая получить трехмерную модель геологического объекта на основе фотограмметрии, была описана в работах [1], [2]. Трехмерную цифровую модель геологического

объекта можно создать с использованием фотографий, сделанных из разных точек съемки, и последующей обработкой этих фотографий с использованием программного обеспечения (в данной работе используется приложение Agisoft Photoscan). Трехмерное изображение обеспечивается наличием, по меньшей мере, двух изображений одной и той же сцены, снятой с разных позиций. Для заданного набора частично перекрывающихся изображений, алгоритмы Structure From Motion (SFM) автоматически обнаруживают набор общих точек в каждой паре изображений, а затем распознают параметры камеры для каждой фотографии. Это, в свою очередь, позволяет определить 3D-координаты каждой точки, распознаваемые как минимум двумя фотографиями и, следовательно, создать облако точек, представляющее поверхности объекта в пределах целевой сцены. По существу, перекрывающиеся фотографии должны быть взяты с нескольких точек зрения. Поэтому стоит использовать одну и ту же камеру для минимизации ошибок от использования различных объективов и датчиков камеры. Также рекомендуется устанавливать фиксированное фокусное расстояние. Фотографии также следует снимать последовательно или, по меньшей мере, при тех же условиях. Эти рекомендации гарантируют, что каждая часть сцены, представленная аналогичным рисунком пикселей на разных фотографиях, облегчает распознавание точек алгоритмами SFM. Эти процедуры обеспечивают и максимизируют распознавание точек, позволяя создавать более плотные облака точек. Однако, для получения фотоснимков геологических объектов, как правило, наземная фотосъемка представляет собой довольно трудоемкий, а порой и невозможный процесс, так как геологические объекты занимают обширные территории в несколько гектаров. В данном случае подходящим и современным решением является фотосъемка с беспилотных летательных аппаратов – дронов. Возможности использования беспилотных летательных аппаратов для отображения геологических структур описаны в работе [3]. Управление дроном можно осуществлять как вручную, так и с помощью программного обеспечения, позволяющего задавать маршруты облета и дистанционно управлять дроном с мобильного устройства. Программное управление квадрокоптером можно осуществлять с помощью

следующих приложений: DJI Go, Altizure, 4dPixCapture.

Таким образом, после того как были получены фотографии моделируемого объекта, можно приступить к их обработке с целью создания 3D-модели объекта. Следует отметить, что для получения качественной модели потребуется около 50-100 фотографий исследуемого объекта.

Доступны различные пакеты фотограмметрии, которые имеют широкий спектр затрат, простоты установки и использования, а также возможность экспорта результатов. Agisoft Photoscan был выбран в нашем рабочем процессе из-за его удобного интерфейса, наличия академического лицензирования и инструментов, позволяющих экспортировать результаты в различных форматах. Для восстановления текстурированной 3D-модели объекта программой PhotoScan необходимо прохождение четырех этапов обработки. Первый этап – определение положений и параметров внешнего и внутреннего ориентирования камер. Результатами этого этапа являются разреженное облако общих точек в 3D пространстве модели и данные о положении и ориентации камер. На втором этапе PhotoScan выполняет построение плотного облака точек (Рис.1.). На третьем этапе PhotoScan строит трехмерную полигональную модель, описывающую форму объекта, на основании плотного облака точек. Заключительный этап – Текстурирование объекта, он включает в себя наложение текстур и/или построение ортофотоплана (Рис.2).

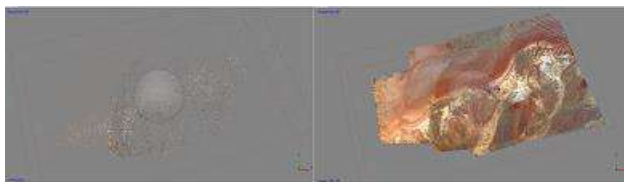


Рис. 1. Результаты первого и второго этапа обработки PhotoScan

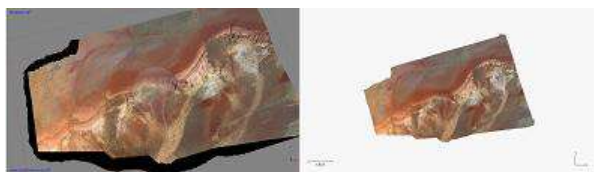


Рис. 2. 3D-модель и ортофотоплан

Таким образом, используя описанную технологию можно получить трехмерные модели различных геологических объектов.

#### ЗАКЛЮЧЕНИЕ

Описанную в данной работе технологию построения пространственных моделей геологических объектов на основе фотограмметрии, в том числе и алгоритмы осуществления фотосъемки с наземных фотокамер и с фотокамер, установленных на беспилотных летательных аппаратах, а также использование программного обеспечения реализующего алгоритмы Structure from motion, можно использовать, как для создания 3D-моделей геологических объектов, занимающих обширные

территории, так и для создания небольших 3D-моделей объектов, расположенных или на маршруте или на объекте, например объекты инфраструктуры, памятники, палеонтологические образцы и др. Полученные описанным способом, 3D-модели и ортомозаики могут быть экспортированы в различные форматы, что позволяет интегрировать их с ГИС, например с популярной Google Earth.

Дальнейшие исследования будут направлены на изучение особенностей различных ГИС, выбор ГИС, наиболее применимой для интеграции элементов 3D-визуализации, разработку технологии интеграции элементов визуализации геологических объектов и ГИС.

#### ЛИТЕРАТУРА

1. Tavani S. Building a virtual outcrop, extracting geological information from it, and sharing the results in Google Earth via OpenPlot and Photoscan: An example from the Khaviz Anticline (Iran)/ S. Tavani, P. Granado, A. Corradetti, M. Girundo, A. Iannace, P. Arbués, J.A. Muñoz, S. Mazzoli // *Computers & Geosciences*. 2014. No 63. P.44–53.
2. Bistacchi A. Photogrammetric digital outcrop reconstruction, visualization with textured surfaces, and three-dimensional structural analysis and modeling: Innovative methodologies applied to fault-related dolomitization (Vajont Limestone, Southern Alps, Italy)./ A. Bistacchi; F. Balsamo, F. Storti, M. Mozafari, et al. // *Geosphere*. 2015, Vol. 11 Iss. 6, P.2031-2048.
3. Yathunathan V. Semi-automatic mapping of geological Structures using UAV-based photogrammetric data: An image analysis approach / Yathunathan Vasuki, Eun-Jung Holden, Peter Kovesi, Steven Micklethwaite // *Computers & Geosciences*. 2014. No 69. P. 22–32.

# Использование Amazon Connect для автоматизации колл-центра

Харабет Р.И.  
КПИ им. Игоря Сикорского  
Киев, Украина

Дорогой Я.Ю.  
КПИ им. Игоря Сикорского  
Киев, Украина

**Аннотация.** В данной статье рассматривается проблема автоматизации современных колл-центров и предлагается ее решение с использованием Amazon Connect. Рассматриваются компоненты Amazon Web Services, с помощью которых можно настроить мощную и гибкую систему для обслуживания клиентов на базе Amazon Connect.

**Ключевые слова:** колл-центр, Amazon Connect, автоматизация, обслуживание клиентов, Amazon Web Services, звонок.

## ВВЕДЕНИЕ

Обслуживание клиентов имеет важное значение для успеха любого бизнеса. Для обеспечения телефонного обслуживания клиентов многие организации создают в колл-центры. К 2020 году, колл-центры будут удовлетворять до 85% обращений клиентов без подключения оператора [1]. В основе колл-центров лежит обработка входящих звонков в очереди и перенаправление их к любому доступному оператору. Более сложные системы поддерживают более сложную маршрутизацию и взаимодействие, в том числе возможность создавать настраиваемые деревья вызовов и другие интегрированные системы голосового ответа (IVR - Interactive Voice Response). Традиционно, IVR-системы имеют сложную архитектуру и их содержание становится все дороже с увеличением масштаба. Несмотря на это, только 7% решений с использованием IVR соответствуют уровню обслуживания человека-оператора [2].

В системах, где число операторов превышает 50, бизнес сталкивается с проблемами. Нужно компоновать несколько решений программного обеспечения, которые целиком охватывают инфраструктуру контакт-центра и решают такие проблемы как очереди, составление рабочих графиков с учетом личных навыков каждого оператора (знание языков, опыт и т.д.), совершение исходящих звонков. С постепенным расширением колл-центра, в архитектуру системы внедряются все новые решения, которые призваны решать появившиеся новые задачи. И, зачастую, в такой ситуации никто не заботится о модернизации и поддержании более старых решений. К тому же, если все это делается в кратчайшие сроки, то о хорошей документации к системе говорить не приходится. Еще одной проблемой данных систем является расширение и поддержка инфраструктуры, технической базы.

Подход, представленный в этой статье, базируется на использовании AWS (Amazon Web Services),

который предоставляет возможность создания облачного колл-центра на базе сервиса Amazon Connect [3]. Преимущество полностью облачного сервиса состоит в том, что компании не надо иметь в штате специалиста, который обслуживал бы всю техническую базу колл-центра. Amazon берет на себя все задачи по масштабируемости системы [4]. Так же Amazon Connect имеет тесную интеграцию с множеством других сервисов, предоставляемых AWS для построения гибких и масштабируемых систем. Каждый компонент имеет подробную документацию, поддерживается и постоянно расширяет свои возможности.

## ИНТЕГРИРУЕМЫЕ КОМПОНЕНТЫ AMAZON WEB SERVICES

Amazon S3 (Amazon Simple Storage Service) [5] – это объектное хранилище данных, позволяет использовать неограниченное пространство для хранения записей звонков и отчетов с использованием шифрования.

Amazon Lambda [6] – дает возможность размещать и исполнять код в облаке с использованием вычислительных мощностей Amazon. Это позволяет получать и использовать информацию из базы данных во время звонка и динамически влиять на сценарий разговора.

Amazon Lex [7] – позволяет строить диалоговые интерфейсы с использованием голоса или текста для взаимодействия с клиентом во время звонка. Эта технология позволяет клиенту просто говорить, что ему нужно, вместо прослушивания пунктов меню и обдумывания, какую цифру ему следует нажать для своей проблемы.

AWS CloudWatch [8] – обеспечивает логирование и мониторинг в реальном времени метрик колл-центра. Например, количество звонков в минуту, количество отклоненных или пропущенных звонков, ошибки при соединении или в процессе разговора.

Amazon Kinesis [9] - платформа для потокового траслирования событий. Данные сохраняются в формате JSON и содержат в себе информацию про звонки, активность операторов. Дополнительно, эти данные могут передаваться в Amazon Elasticsearch для их анализа с помощью графического интерфейса.

AWS Identity and Access Management [10] – сервис для назначения прав и политик доступа к данным, создающим вертикальную иерархию доступа к используемым ресурсам Amazon Web Services аккаунта.

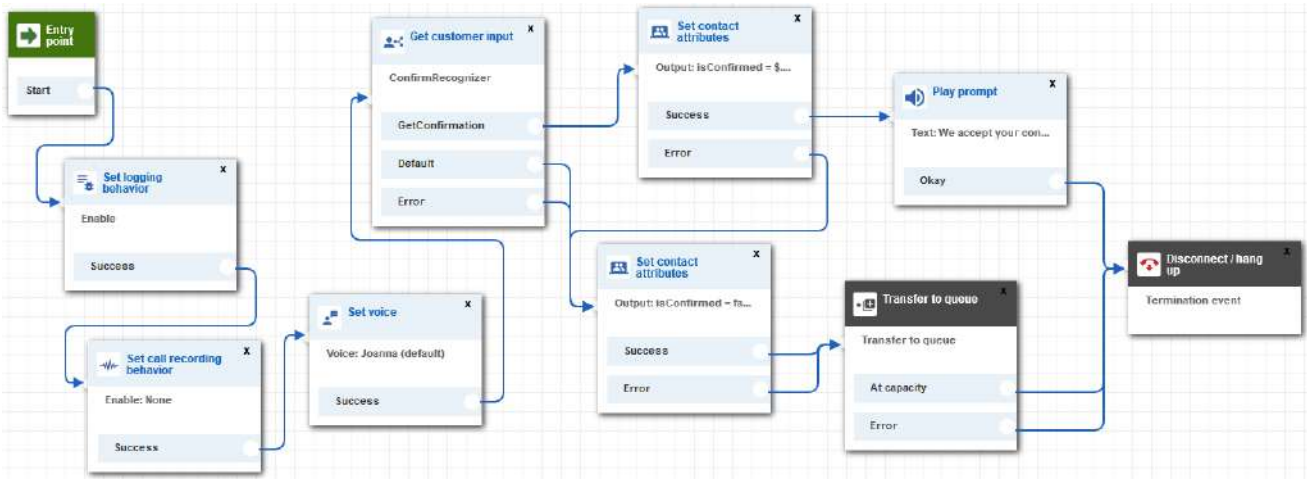


Рис. 1. Сценарий исходящего звонка для подтверждения заказа

### ПОСТРОЕНИЕ КОЛЛ-ЦЕНТРА НА БАЗЕ AMAZON CONNECT

Ключевым понятием в Amazon Connect является сценарий звонка (Contact Flow), который представляет из себя последовательность блоков, которая реализуется в ходе звонка. Разрабатывая различные сценарии, система получает возможность обрабатывать входящий звонок, и динамически изменять ход событий в течение разговора. Рассмотрим пример сценария автоматизированного исходящего звонка клиенту без участия оператора для оповещения о неподтвержденном заказе рис. 1.

При соединении с абонентом первые три блока в сценарии (Set logging behavior, Set call recording behavior, Set voice) устанавливают логирование, запись разговора и тип голоса которым будет общаться бот с клиентом соответственно. Get customer input использует бота Amazon Lex. Бот озвучивает текст по шаблону, подставляя в него имя клиента и всю необходимую информацию, которая были получена из базы данных во время установления соединения с помощью AWS Lambda. Затем бот прослушивает ответ клиента, анализирует его. На выходе блока мы получаем положительный или отрицательный ответ клиента. Если клиент подтвердил заказ, то в базу сохраняется его подтверждение (блок Set contact attributes), зачитывается резюмирующий текст и система кладет трубку. Если же клиент не подтверждает заказ, система перенаправляет звонок в очередь к любому свободному оператору колл-центра.

### ВЫВОДЫ

Таким образом, при грамотном планировании сценариев исходящих и входящих звонков, можно обрабатывать 20-50% звонков без участия оператора, что можно считать хорошим показателем. К тому же, системы синтеза речи в наше время разработаны настолько хорошо, что около 27% клиентов не уверены общались ли они с роботом или человеком [11]. Преимущество использования Amazon Connect состоит в том, что есть возможность гибко настраивать и компоновать различные сервисы из единой экосистемы Amazon Web Services для

достижения желанного результата. Такое решение освобождает операторов от монотонной работы, экономит затрачиваемые человеческие ресурсы, позволяя направлять их туда, где интеллект человека действительно необходим.

### ЛИТЕРАТУРА

1. <https://medium.com/@electrobabe/gartner-stated-by-2020-customers-will-manage-85-of-their-relationship-with-the-enterprise-f34736d4bf19> (дата обращения 24.04.2018)
2. <https://www.icmi.com/Resources/Self-Service/2014/11/Can-IVR-Destroy-the-Customer-Experience> (дата посещения 24.04.2018)
3. <https://aws.amazon.com/ru/connect/> (дата обращения 24.04.2018)
4. <https://aws.amazon.com/autoscaling/features/> (дата посещения 24.04.2018)
5. <https://aws.amazon.com/ru/s3/> (дата обращения 24.04.2018)
6. <https://aws.amazon.com/ru/lambda/> (дата обращения 24.04.2018)
7. <https://aws.amazon.com/ru/lex/> (дата обращения 24.04.2018)
8. <https://aws.amazon.com/ru/cloudwatch/> (дата обращения 24.04.2018)
9. <https://aws.amazon.com/ru/kinesis/> (дата обращения 24.04.2018)
10. <https://aws.amazon.com/ru/iam/> (дата обращения 24.04.2018)
11. <https://economictimes.indiatimes.com/small-biz/security-tech/technology/the-rise-of-the-chatbot-in-the-contact-center/articleshow/60709336.cms> (дата обращения 24.04.2018)

# Аналіз алгоритмів оптимізації обчислення локального мінімуму функції Розенброка

Писаренко Олег  
КПІ ім. І. Сікорського  
Київ, Україна  
oa.pisarenko@gmail.com

**Анотація.** В роботі демонструється аналіз алгоритмів оптимізації обчислень на прикладі функції Розенброка з використанням середовища LabView. Були обрані метод градієнтного спуску та генетичний алгоритм. Описано принцип роботи алгоритмів. Проведено обчислення локального мінімуму функції Розенброка. Була порівняна кількість виконаних ітерацій для кожного алгоритму. Кращу швидкість виконання показав генетичний алгоритм.

**Ключові слова:** функція Розенброка, генетичний алгоритм, метод градієнтного спуску, LabView, оптимізація.

## ВСТУП

В сучасному світі для вирішення багатьох проблем в галузях промисловості та автоматики нерідко піднімається проблема оптимізації. Оптимізація – пошук набору параметрів, що мінімізують функцію. Наприклад, оптимізацію можна використати для визначення певних параметрів для створенні систем управління або пошуку умов, які можуть мінімізувати вартість процесу виробництва. В даній роботі розглядаються метод градієнтного спуску та генетичний алгоритм та виконується їх порівняння на основі кількості ітерацій, що необхідні для знаходження локального мінімуму функції Розенброка.

## ФУНКЦІЯ РОЗЕНБРОКА

Для реалізації та тесту обраних алгоритмів була взята функція Розенброка [1] – невіпукла функція, яка використовується для оцінки продуктивності алгоритмів оптимізації.

Функція Розенброка для двох змінних визначається так:

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2 \quad (1)$$

Локальний мінімум функції Розенброка знаходиться в точці  $(x, y) = (1, 1)$ , де  $f(x, y) = 0$ .

Графічне зображення функції наведено на рис. 1.

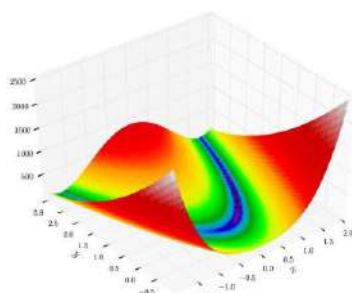


Рис. 1. Графік функції Розенброка

## МЕТОД ГРАДІЄНТНОГО СПУСКУ

Метод градієнтного спуску – також відомий як метод деформованого багатогранника і симплекс-метод, – метод безумовної оптимізації функції від декількох змінних, а тому легко застосовується до негладких і / або зашумлених функцій [2].

Суть методу полягає в послідовному переміщенні і деформації симплекса навколо точки екстремуму.

Градієнтний спуск працює в просторах будь-якої кількості розмірів, навіть у нескінченновимірних.

Також, алгоритм може вимагати багато ітерацій для обчислення локального мінімуму з необхідною точністю, якщо кривизна в різних напрямках для даної функції дуже відрізняється.

Метод знаходить локальний екстремум і може «застрягти» в одному з них. Якщо все ж потрібно знайти глобальний екстремум, можна пробувати вибрати інший початковий симплекс. Більш розвинений підхід до виключення локальних екстремумів пропонується в алгоритмах, заснованих на методі Монте-Карло, а також в еволюційних алгоритмах.

## ГЕНЕТИЧНИЙ АЛГОРИТМ

Одним із еволюційних алгоритмів є генетичний – алгоритм пошуку, який використовується для вирішення завдань оптимізації та моделювання шляхом випадкового підбору, комбінування і варіації шуканих параметрів. Він є різновидом еволюційних обчислень, за допомогою яких вирішуються оптимізаційні задачі з використанням методів природної еволюції, таких як успадкування, мутації, відбір і кросингвер (генетичний оператор) [3]. Відмінною особливістю генетичного алгоритму є акцент на використання оператора «схрещування», який виробляє операцію рекомбінації рішень-кандидатів, роль якої аналогічна ролі схрещування в живій природі.

Рішення, знайдене генетичним алгоритмом, називається «хромосомою» [4]. Хромосома складається з генів, і її значення може бути числовим, бінарним або символічним, в залежності від проблеми, яку потрібно вирішити. Ці хромосоми проходять через фітнес-функцію (функцію пристосування), для визначення придатності рішення, створеного генетичним алгоритмом, з задачею. З отриманої множини рішень, беручи до уваги значення «пристосованості» обираються рішення до яких застосовується кросингвер, в результаті чого отримуються нові рішення. Для них також обраховується значення пристосованості, а потім

проводиться відбір кращих рішень в наступне покоління.

Цей набір дій повторюється ітеративно і моделює еволюційний процес, що продовжується декілька життєвих циклів, доки не буде виконана умова зупинки алгоритму. Однієї із таких умов зупинки є знаходження локального або глобального екстремуму.

#### ОПТИМІЗАЦІЯ ОБЧИСЛЕННЯ ФУНКЦІЇ РОЗЕНБРОКА

Для реалізації обраних алгоритмів та обчислення за допомогою них функції Розенброка був обраний програмний засіб LabView. Він надає можливості, що спеціально призначені для задач моделювання, оптимізації, управління і автоматизації, прискорюючи процес розробки.

Інструмент LabView дозволяє реалізувати базові алгоритми оптимізації власними, завчасно створеними об'єктами. Для методу градієнтного спуску існує стандартний віртуальний інструмент (програма, скорочено VI) Downhill Simplex nD VI.

Для коректного запуску алгоритму треба підключити необхідні вхідні параметри та вивести на передню панель можливість задавати дані та відображення результатів.

Для реалізації генетичного алгоритму на базі LabVIEW була обрана стороння бібліотека Wapria – genetic optimization algorithm. Вона намагається знайти мінімальне значення заданої функції, використовуючи функцію пристосованості [5].

Генетичний алгоритм є більш складним в реалізації за базовий, який розглядався вище, але дозволяє більш гнучко і точно налаштувати роботу алгоритму для досягнення бажаних результатів.

Після реалізації методу градієнтного спуску та генетичного алгоритму було проведено п'ять експериментальних обчислень для пошуку локального мінімуму для функції Розенброка. Результати обчислень для кожного алгоритму наведені в табл. 1.

Таблиця 1

Кількість ітерацій для знаходження мінімуму функції Розенброка

№ експ.	Градієнтний спуск	Генетичний алгоритм
1	726	20
2	726	19

№ експ.	Градієнтний спуск	Генетичний алгоритм
3	725	20
4	720	20
5	723	21

#### ВИСНОВКИ

В результаті виконання даної роботи було розглянуті два алгоритми оптимізації. А саме: базовий – градієнтного спуску та сучасний – генетичний алгоритм. Алгоритми були програмно реалізовані в середовищі LabView. Як можна побачити з результатів роботи алгоритмів для обчислення локального мінімуму функції Розенброка, метод градієнтного спуску для знаходження мінімуму функції потребує понад 700 обчислень. В той же час, для досягнення такого ж результату генетичному алгоритму достатньо лише близько 20 обчислень. Таким чином, можна зробити висновок, що базовий алгоритм не підходить для оптимізації складних функцій і для цього краще використовувати більш сучасний генетичний алгоритм, який дає більше можливостей для конфігурації його роботи та показує малу кількість ітерацій для знаходження екстремуму функції.

#### ЛІТЕРАТУРА

1. [https://en.wikipedia.org/wiki/Rosenbrock\\_function](https://en.wikipedia.org/wiki/Rosenbrock_function) (дата звернення 03.04.2018)
2. Nelder J.A. A simplex method for function minimization / J.A. Nelder, R. Mead // Computer Journal. – 1965. – vol. 1, is. 7. – P. 308-313.
3. Sivanandam S. N. Introduction to Genetic Algorithms / S. N. Sivanandam, S. N. Deepa. – Berlin: Springer, 2007. – С. 127.
4. Kramer O. Genetic Algorithm Essentials / Oliver Kramer. – Oldenburg: Springer, 2017. – С. 50.
5. <https://towardsdatascience.com/how-to-define-a-fitness-function-in-a-genetic-algorithm-be572b9ea3b4> (дата звернення 10.04.2018)

Рецензент: к.т.н., доц. каф. АУТС, КІІІ ім. Ігоря Сікорського  
А.В. Писаренко



# Автоматизація інструментального дослідження параметрів кабельних ліній

Моргаль Олег  
КПШ ім. Ігоря Сікорського  
Київ, Україна  
m\_olegm@ukr.net

Шихутський Сергій  
КПШ ім. Ігоря Сікорського  
Київ, Україна  
sergey.shykhutskyi@gmail.com

**Анотація.** Сертифікаційні випробування ліній структурованих кабельних систем (СКС) на відповідність вимогам певних категорії чи класу не дають повної картини поточного стану СКС і не мають інструментів для порівняльного аналізу для виявлення змін у стані системи. Автори пропонують доповнити програму випробувань вимірами параметрів TDR та TDX і автоматизувати методику порівняльного аналізу вимірів з метою виявлення відхилень у стані системи.

**Ключові слова:** СКС, TDR, TDX, кабельний тестер

Фізичним комунікаційним середовищем для побудови сучасних інформаційних систем є структуровані кабельні системи (СКС). Елементарною одиницею СКС є базова лінія (надалі – лінія), яка складається з відрізка кабелю, прокладеного від передавача даних до приймача та приєднаних до його кінців термінальних елементів («роз'ємів»), що забезпечують технічну можливість підключати до лінії передавально-приймальне обладнання.

При побудові СКС, запорукою досягнення нею найвищих характеристик є суворе дотримання положень технології виконання монтажу кабелів і термінальних елементів. Це дозволяє максимально використати потенційні можливості, закладені у компоненти СКС при їх виробництві. Вимоги до характеристик СКС задаються міжнародними стандартами. Найбільш поширеними системами стандартів є американська – ANSI/TIA/EIA [1] (в ній СКС класифікуються за категоріями: 3, 4, 5, 5e, 6, 6a, 7), та міжнародна – ISO [2] (в ній СКС класифікуються за класами: C, D, E, Ea, F, Fa). Щоб переконатись чи задовольняють параметри ліній побудованої СКС умовам надійної роботи приймально-передавального обладнання після завершення монтажних робіт виконується сертифікація кожної лінії – індивідуальне випробування, в ході якого перевіряється відповідність параметрів піддослідної лінії вимогам міжнародних стандартів для ліній певної категорії чи класу. Наприклад, щоб по лінії СКС можна було передавати дані зі швидкістю 100Мбіт/с її характеристики повинні задовольняти вимогам категорії 5, а для швидкості передачі 40Гбіт/с – вимогам класу Fa.

Сертифікаційні випробування виконуються за допомогою спеціалізованих приладів – кабельних тестерів, які є також ефективним інструментом дослідження характеристик ліній СКС завдяки можливості виконувати вимірювання в більш широкому діапазоні частот для ширшого переліку параметрів ніж вимагають стандарти. Саме до таких параметрів, які відомі, але широко не використовуються для оцінки стану СКС, відносяться

рефлектограми ліній по рівню зворотних сигналів (TDR) і по рівню зворотних перехресних завад (TDX).

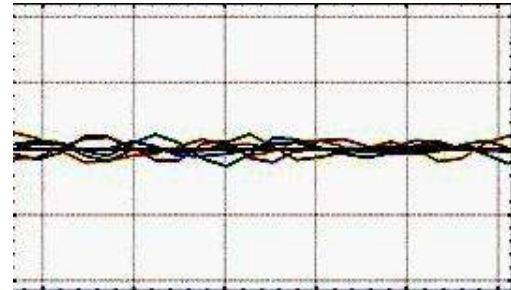


Рис. 1. Ділянка кабелю до деформації

Непопулярність у використанні цих вимірів зумовлена відсутністю автоматизованих засобів їх обробки – в кабельних тестерах результати цих вимірів представляються лише у графічному вигляді і оцінювати їх можна лише візуально.

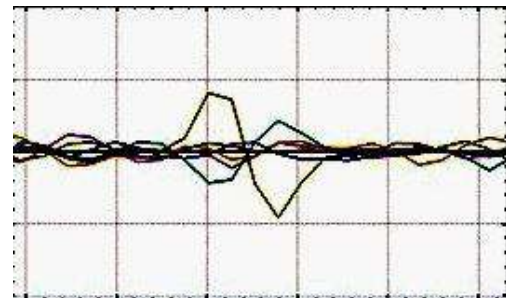


Рис. 2. Ділянка кабелю після згину

Ми пропонуємо метод автоматизованої обробки вимірів TDR і TDX для забезпечення документування стану ліній СКС і можливості більш глибокого його аналізу.

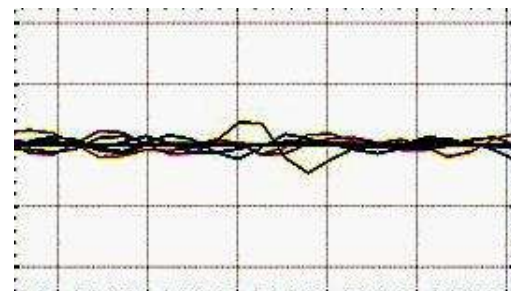


Рис. 3. Ділянка кабелю після розрівнювання

Якість елементної бази визначають асимптотичні характеристики лінії СКС, які можуть бути досягнуті при ідеально виконаному монтажі. В реальних умовах навіть незначні порушення технології монтажу призводять до змін у просторовій будові кабельної



лінії, що веде до її локального розбалансування, а як результат – до росту згасання корисного сигналу і підвищення рівня перехресних завад.

Документування вимірів TDR/TDX дає можливість зберегти унікальну інформацію про початковий стан конкретної лінії СКС на всій її протяжності (від одиниць до 90м) на момент завершення її побудови з можливістю порівняння у майбутньому поточного стану з початковим і прогнозування його зміни в майбутньому та оцінки терміну ефективної експлуатації лінії.

Нами пропонується метод перевірки дотримання технологічного регламенту при виконанні монтажних робіт, побудований на виявленні місць, в яких лінія була пошкоджена, шляхом порівняльного аналізу вимірних значень TDR/TDX з типовими величинами. Для реалізації такого підходу на підготовчому до обслідування СКС етапі монтується еталонна лінія з таких же компонент, як і лінії піддослідної СКС, і експериментальним шляхом будується бібліотека параметрів для типових вузлів.

Фізичне підґрунтя для реалізації цього підходу ілюструють наведені діаграми параметра TDX. (Рис.1 – Рис.3). Вони отримані на ділянці кабеля, на якій кабель був перегнутий, а потім розрівняний.

Як видно з наведених діаграм деформації кабеля не проходять безслідно, а значить, при досягненні їхньої кількості критичної межі категорія (клас) кабельної лінії понизиться.

Експериментальна частина нашого дослідження була виконана з використанням кабельного тестера FLUKE DTX-1800 [3]. При цьому деякі операції виконувались у ручному режимі, зокрема, експорт результатів вимірювання і обробка результатів засобами Microsoft Excel. Тому, як чергова задача, по доведенню запропонованої методики дослідження кабельних ліній СКС до зручного практичного застосування, сформульоване завдання на розробку програмного інструменту для автоматизації документування і обробки результатів вимірів TDR/TDX.

#### ВИСНОВКИ

Отримані результати інструментальних досліджень підтверджують діагностичну ефективність запропонованого підходу, а його програмна реалізація стане зручним інженерним інструментом для оцінки поточного технічного стану ліній СКС.

#### ЛІТЕРАТУРА

1. ISO/IEC 11801 Information technology – Generic cabling for customer premises.
2. ANSI/TIA/EIA 568-B Commercial Building Telecommunications Cabling Standard.
3. DTX Series CableAnalyzer April 2004, Rev. 12 4/12 ©2004-2012 Fluke Corporation.

# Чатбот на основі нейронних мереж

Боярин Олесь  
КПІ ім. Ігоря Сікорського  
Київ, Україна  
boiaryn.o@gmail.com

Дячук Іван  
КПІ ім. Ігоря Сікорського  
Київ, Україна  
diachukivan96@gmail.com

Дорогий Ярослав  
КПІ ім. Ігоря Сікорського  
Київ, Україна  
argusyk@gmail.com

**Анотація.** Здійснено аналіз та порівняння різних архітектур на базі нейронних мереж для реалізації чатботу. Було обрано найбільш вдалу архітектуру для втілення поставленої задачі.

**Ключові слова:** нейронні мережі, чатбот, метону networks, LSTM, наука про дані.

## ВСТУП

У зв'язку із ростом популярності додатків-месенджерів, набувають високої популярності так звані чатботи. Чатбот являє собою програму, яка може спілкуватися з користувачем, імітуючи поведінку людини, для досягнення певної цілі, або ж розваги.

Найпростішим варіантом побудови чатботу є його реалізація на основі певних правил (ключові слова, ключові фрази, регулярні вирази тощо). Проте, цей варіант є досить неефективним у зв'язку з тим, що спочатку потрібно побудувати ці правила, й час від часу змінювати їх власноруч, що значно ускладнює розробку та підтримку чатботу.

Застосування нейронних мереж для побудови чатботів відкидає вище описані недоліки, а також робить їх більш витонченими, та придатними для широкого кола завдань.

Якість побудованого чатботу на основі нейронних мереж залежить від трьох параметрів: об'єму та якості даних для навчання і, відповідно, якості побудованої архітектури.

## ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

Архітектура на базі TF-IDF: на основі алгоритму TF-IDF обираються найбільш вживані слова у корпусі, а також найбільш рідкі слова у корпусі. На основі зібраних слів будуються вектори, після чого рахується косинусна відстань від питання до відповіді, після чого обирається відповідь найбільш ближча до питання [6].

Архітектура на базі LSTM: в основі цієї архітектури лежить рекурентна нейронна мережа з вузлами, які базуються на архітектурі довгої короткочасної пам'яті (LSTM).

Дані архітектури були використані для побудови чатботу на корпусі Ubuntu [7], який містить діалоги зібрані з форуму Ubuntu. Набір даних складається з 1 мільйона пар питання-відповідь, для навчання було обрано 0,5 мільйонів пар та 0,5 мільйонів пар для тестування. Позитивні відповіді є справжніми відповідями від людей, а негативні – обрані випадковим чином.

Для оцінювання якості вище описаних архітектур було обрано 2% розмов з корпусу для створення тестового набору. Поставлене завдання – вибір найкращої відповіді. Системі надається  $n$  відповідей, і

її завданням є ранжування цих відповідей. Для більш точної оцінки чатботу, робота системи вважається коректною якщо правильна відповідь лежить у межах  $k$ -перших кандидатів (у табл. 1 це зображено як  $n R@k$ ).

Таблиця 1

Порівняння існуючих архітектур

	TF-IDF	LSTM
1 з 2 R@1	65.9%	90.1%
1 з 10 R@1	41%	63.8%
1 з 10 R@2	54.5%	78.4%
1 з 10 R@5	70.8%	94.9%

Отже, роблячи висновки на основі наведених даних у табл. 1 – можна сказати, що архітектури на базі нейронних мереж дають набагато кращі результати, ніж архітектури на основі правил.

## HYBRID CODE NETWORKS

В основі реалізації даної архітектури лежить рекурентна нейронна мережа з вузлами, які базуються на архітектурі довгої короткочасної пам'яті (LSTM) [1]. Дана архітектура вимагає набору заздалегідь визначених шаблонів дій [2].

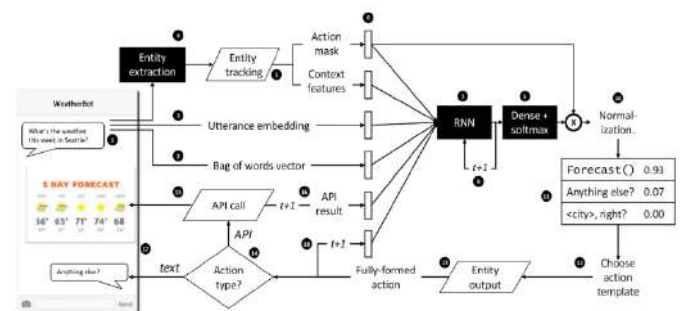


Рис. 1. Архітектура Hybrid code networks

Для навчання мережі використовуються такі параметри:

- питання користувача у текстовому вигляді;
- представлення питання у вигляді вектору мішки слів (Bag of Words);
- представлення слів у вигляді векторів word2vec;
- визначення сутностей у вираженні, як ім'я, місце, час тощо;
- вилучення та відслідковування сутностей під час діалогу, та представлення у вигляді бітової маски;

- маска дій – бітне векторне уявлення про те, які дії дозволяються в даному контексті.

Процес навчання мережі виглядає наступним чином: на вхід до мережі подаються наступні параметри: питання користувача у вигляді вектору мішку слів, слова вилучені з питання у вигляді векторів word2vec, вилучені сутності. Всі ці параметри об'єднуються та подаються на вхід. На виході з мережі ми отримуємо вірогідність появи кожного шаблону дій, помноживши на бітову маску дій – обираємо результат з найбільшою вірогідністю – це й буде відповідь, яку поверне бот на поставлене питання.

Переваги:

- швидкість навчання;
- висока точність.

Недоліки:

- потрібен додатковий аналіз для визначення шаблонів дій;
- вузькоспеціалізована архітектура.

MEMORY NETWORKS

Представлення вхідної пам'яті: припустимо ми маємо вхідний набір  $x_1, \dots, x_i$  для збереження в пам'яті. Вхідний набір  $\{x_i\}$  конвертується в вектори пам'яті  $\{m_i\}$  розміру  $d$ , який розраховується шляхом вкладення кожного  $x_i$  в безперервний простір, у найпростішому випадку з використанням матриці вкладення  $A$  розміру  $d \times V$ . Запит  $q$  також вкладений в безперервний простір для отримання внутрішнього стану  $u$ . Співвідношення між внутрішнім станом  $u$  та кожною пам'яттю  $m_i$  розраховується за наступною формулою:

$$p_i = \text{Softmax}(u^T m_i),$$

$$\text{де } \text{Softmax}(z_i) = e^{z_i} / \sum_j e^{z_j}.$$

Визначений таким чином вектор  $p$  – вектор імовірності входів.

Представлення вихідної пам'яті: кожен  $x_i$  має відповідний вихідний вектор  $c_i$ . Вектор відповіді з пам'яті  $o$  являє собою суму всіх трансформованих входів  $c_i$ , кожен з яких має вагу, отриману з вектору імовірності входу.

$$o = \sum_j p_j c_j.$$

Оскільки функція від вводу до виводу є гладкою, ми можемо легко обчислити градієнти під час навчання мережі, що значно спрощує цей процес.

Генерація остаточного прогнозу: у випадку одного шару, сума вихідного вектору  $o$  і вхідного вкладення  $u$  потім пропускається через остаточну матрицю ваги  $W$  (розміру  $V \times d$ ) і softmax для отримання прогнозованої мітки:

$$\hat{a} = \text{Softmax}(W(o + u)).$$

Загальна модель показана на рис.1 (а). Під час тренування всі три матриці вкладення  $A$ ,  $B$  і  $C$ , а також  $W$  спільно навчаються шляхом мінімізації стандартної втрати крос-ентропії між  $\hat{a}$  і істинною міткою  $a$  [3]. Навчання проводиться з використанням стохастичного градієнтного спуску.

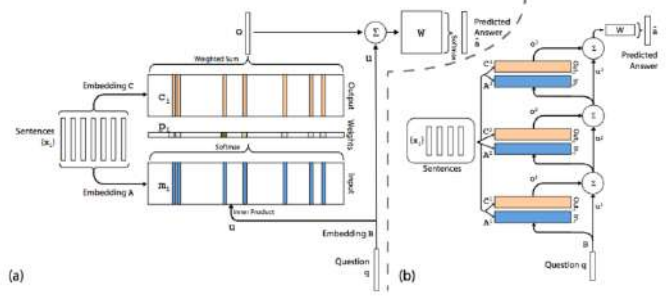


Рис. 2. (а) Одношарова архітектура. (б) Багатошарова архітектура

У випадку з багатошаровою архітектурою шари пам'яті пов'язані таким чином:

- вхід до шарів над першим є сумою виходу  $o^k$  і входу  $u^k$  із шару  $k$ :

$$u^{(k+1)} = o^k + u^k;$$

- кожен шар має власні матриці вкладень  $A^k, C^k$  для вкладення  $\{x_i\}$  у безперервний простір;
- у верхній частині мережі вхід до  $W$  також поєднує вхід і вихід верхнього рівня пам'яті:

$$\hat{a} = \text{Softmax}(W(o^k + u^k)).$$

Переваги:

- відноситься до типу end-to-end систем, тобто не потребує додаткового аналізу над вхідними даними.

Недоліки:

- швидкість навчання;
- програє у точності у порівнянні з вузькоспеціалізованою архітектурою.

GATED MEMORY NETWORKS

В основі реалізації даної моделі лежить архітектура Memory Network [4], яка була розглянута вище. Вона відрізняється від базової архітектури тим, що в даній моделі внутрішній стан  $u$  визначається наступним чином:

$$u^{(k+1)} = o^k * T^k(u^k) + u^k * (1 - T^k(u^k)), \text{ де}$$

$$T^k(u^k) = \sigma(W_T^k u^k + b_T^k)$$

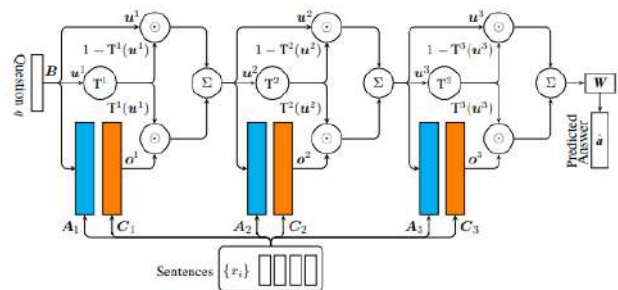


Рис. 3. Архітектура Gated Memory Networks

Переваги:

- також є реалізацією end-to-end системи;
- дає більшу точність у порівнянні з базовою реалізацією.

Недоліки:

- швидкість навчання;
- програє у точності у порівнянні з вузькоспеціалізованою архітектурою.

ДАНІ ДЛЯ НАВЧАННЯ

Для навчання та порівняння роботи моделей був обраний dialogue bAbI tasks, який вміщує в собі 5 різних завдань для тестування кінцевих систем діалогу в домені ресторану. Завдані 1 і 2 – управляють тестовим діалогом, щоб дізнатися, чи можуть кінцеві системи вивчати можливість неявного відстеження стану діалогу, тоді як завдання 3 та 4 перевіряють, чи можуть вони навчитися використовувати факти бази знань у діалоговому вікні [5]. Завдання 3 також перевіряє вміння сортування. Завдання 5 об'єднує всі завдання. Саме 5 завдання було обрано для порівняльної характеристики моделей.



Рис. 4. Приклад діалогу

Загалом було обрано 1000 діалогів для навчання моделей, а також 1000 діалогів для тестування.

РЕЗУЛЬТАТИ РОБОТИ АРХІТЕКТУР

Таблиця 2

Порівняльна характеристика моделей

	Hybrid Code Network	Memory Network	Gated Memory Network
Task 5	100%	96.1%	96.3%
Task 5 – OOV*	98%	65.6%	66.7%

\*OOV (Out of vocabulary) – тестування моделей у випадку, коли трапляються слова, які були відсутні під час навчання

ВИСНОВКИ

Були розглянуті різні архітектури на основі нейронних мереж для побудови чатботу. За результатами дослідження поданими у табл. 2 видно, що найбільш вдалою реалізацією є архітектура Hybrid Code Networks, проте для реалізації даної моделі потрібен додатковий аналіз набору вхідних даних для створення визначених шаблонів дій.

ЛІТЕРАТУРА

1. Long short-term memory [Електронний ресурс] – Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Long\\_short-term\\_memory](https://en.wikipedia.org/wiki/Long_short-term_memory).
2. Jason W. Hybrid Code Networks [Електронний ресурс] / W. Jason, A. Kavosh, Z. Geoffrey – Режим доступу до ресурсу: <https://arxiv.org/pdf/1702.03274.pdf>.
3. End-To-End Memory Networks [Електронний ресурс] / S.Sainbayar, S. Arthur, W. Jason, F. Rob – Режим доступу до ресурсу: <https://arxiv.org/pdf/1503.08895.pdf>.
4. Fei L. Gated End-to-End Memory Networks [Електронний ресурс] / L. Fei, P. Julien – Режим доступу до ресурсу: <https://arxiv.org/pdf/1610.04211.pdf>.
5. Antoine B. Learning End-to-End Goal-Oriented Dialog [Електронний ресурс] / B. Antoine, B. Y-Lan, W. Jason – Режим доступу до ресурсу: <https://arxiv.org/pdf/1605.07683.pdf>.
6. Rudolf K. Improved Deep Learning Baselines for Ubuntu Corpus Dialogs [Електронний ресурс] / K. Rudolf, S. Martin, K. Jan – Режим доступу до ресурсу: <https://arxiv.org/pdf/1506.08909.pdf>.
7. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems [Електронний ресурс] / L.Ryan, P. Nissan, S. Iulian, P. Joelle – Режим доступу до ресурсу: <https://arxiv.org/pdf/1506.08909.pdf>.

# *Index structure optimizations for in-memory data storage systems*

Oleksandr Podrubailo, Vasyl Saverchenko  
 Igor Sikorsky Kyiv Polytechnic Institute  
 Kyiv, Ukraine  
 alex.podrubailo@gmail.com

**Abstract.** In-memory data grid systems uses RAM as main storage device, which takes the performance of such systems on the new level. However, new in-memory systems often uses old data structures and algorithms which were proposed for relational database management systems and are not suitable for main memory, in particular, they do not take into account number of cache misses. This paper describes methods of index structure optimizations for in-memory storage solutions, which will allow to significantly reduce the number of comparatively slow reads from main memory.

**Keywords:** in-memory data grid, deleting key fields, rearrange fields.

## INTRODUCTION

Modern information technology requires the availability of reliable, fast and convenient, data storage systems. In-memory data grid (IMDG) is potentially able to meet these requirements.

The concept of IMDG does not coincide with that of traditional databases on long-term media. The main idea of such grids is the placement of data in distributed maps (so-called caches or regions) that are completely located in RAM. This allows to reduce the time of information access. Regions of distributed storage contain objects. This type of storage eliminates the costs of marshalling/unmarshalling when working with data in object-oriented environments.

However, due to its relative novelty, the technology of distributed storage in RAM has a number of drawbacks [1]. Among them, there are inefficient data structures that are used to build IMDG indexes. [2]

Over the last 10-15 years there have been many methods developed to optimize cache utilization in various areas: database [3, 4], a packages of linear algebra [5, 6], virtual machines and the garbage collector [7], the support software for network routers [8], etc.

Here we propose meta-classification of common techniques to optimize the use of cache memory. We will distinguish between methods of optimization whether their application fields: algorithms or data structures. Methods from first type re-factors algorithms so that the resulting algorithms gets minimal number of cache misses due to reduction of the algorithm working set. Second type methods improve cache utilization by changing the data structure, the algorithms on this data structure (e.g., search algorithms) remains the same.

In this paper we describe data structure changes which will make IMDG indexes more effective.

## DATA STRUCTURE OPTIMIZATIONS

The methods discussed in this paper assume that data structure consists of nodes. Links between nodes can be implemented by pointers. The purpose of all considered methods is to reduce the number of cache misses that occur

when the scanning algorithm is executed, the generalized pseudocode of which is shown in Listing 1.

```
00 void dataScan (Node start) {
01   Node current = start;
02   while (current != null) {
03     current = getNext (current);
04   }
05 }
```

Listing 1. A generalized algorithm for scanning a data structure

The main in this algorithm is a function getNext, which analyzes the current node and returns a null, if the search is complete, or the next node otherwise. Note that the code above is summarized key search in any data structure, hierarchical like B- and B+ tree, or search in the hash table collisions list, as well as view the array. In addition, since current changes at each iteration, and the Node size is larger than the cache block, address to current is always a reference to the new cache block. All methods of optimizing data structures are characterized the locality of their usage: these methods reduce cache misses for the first call under the assumption that there is no results of previous searches in the cache.

## NODE STRUCTURE

We can achieve improvements by optimizing the use of cache block by Node. It is obvious that the smaller the size of Node gives us advantages in terms of the use of the cache block.

Consider possible ways to optimize the use of cache block by Node. We assume that the Node = (field<sub>1</sub>, ..., field<sub>N</sub>), where N – number of fields in Node. Fields is either the key field or a pointer to another Node. We assume that the key fields constitute the information part of the node, and pointers are needed to support the data structure.

*Deleting key fields.* Removing the key fields (fluff extraction) is proposed in [9]. Key fields that are not used in getNext functions only occupy space in the cache block and may be removed from the assembly, for example by removal into a separate data structure. Depending on the specific situation, this optimization may or may not be applicable, since it leads to the appearance of a second data structure, the connection in which links are completely repeated in the original one. Obviously, two structures occupy, in general, more memory space than one, but in some cases such conversions are beneficial.

*Example.* Consider the ratio Persons (Age int, Name char (30), Position char (30)). Assume that the relationship is stored as an array of records and pointers are absent, i.e. Node = (Age, Name, Position). getNext is used for calculating the average age of persons, so, a single field in Node, Age, is necessary for getNext. If cache block size is

128 bytes, then one cache block can hold almost 2 nodes. Thus, during the scan occurs about  $\lfloor \text{Persons}/2 \rfloor$  of first call cache misses.

We can move Name and Position fields in a separate array. Then  $\text{Node} = (\text{Age})$ , and one cache block can hold 32 nodes. The resulting number of misses of the first circulation thus decreases by more than 10 times.

*Rearrange fields.* The regrouping of fields (field reordering) is applicable in a wide range of cases, the removal of the key fields. It is not always possible to make unused Examine-Node field in a separate structure. In this case, you can try to rearrange the fields in such a way that the fields used at close moments of time are nearby in the node. This transformation concerns not only key fields, but also pointers.

*Example 1.* Consider another embodiment relationship Students (PassportNumber char (10), Year int, FirstName char (20), LastName char (20), Address char (50), PhoneNumber char (8), RecordBookNumber char (6), AvgGrade int), where the Year and AvgGrade - a course on which the student learns, and the average score for the current academic year, respectively. Obviously, the size of the record is 122 bytes. If the cache block size is 64, the recording takes almost two cache blocks. Assume that the scanning ratio, during which the calculated average scores of students who are studying various courses (ExamineNode in this case uses field Year a AvgGrade). since the Year a AvgGrade are in different cache blocks, ExamineNode function is, generally speaking, the two cache miss. It is therefore advantageous to reorganize the record so that Year AvgGrade and are close, e.g., Students = (PassportNumber, Year, AvgGrade, FirstName, LastName Address, PhoneNumber, RecordBookNumber). In the process under consideration operation dataScan will use only the first cache block during getNext and will lead to a single cache miss.

*Example 2.* Consider node of k-ary tree. In this case, the  $\text{Node} = (\text{ptr}_0, \text{key}_1, \text{ptr}_1, \dots, \text{key}_k, \text{ptr}_k)$ , where  $\text{key}_i$  - key field, and  $\text{ptr}_i$  - pointers to child nodes, and  $\text{key}_0 < \text{key}_1 < \dots < \text{key}_k$ . Suppose that performed search for key  $\text{key}$  and Examine-Node finds  $l = \max \{i: \text{key}_i \leq \text{key}\}$ . In this case it's useful to group all the keys together:  $\text{Node} = (\text{key}_0, \dots, \text{key}_k, \text{ptr}_1, \dots, \text{ptr}_k)$ . At the same time appeal to the in-dex for the search will require no more than 1 cache miss in addition to the cache misses that have occurred during the (binary) search in the array  $(\text{key}_0, \dots, \text{key}_k)$ .

*Compression of key fields.* Compression of key fields allows reducing size of key fields and, consequently, the number of occupied node cache blocks (or putting more nodes in a cache block). Often, the key fields are non-negative integers, which are located in the node in ascending order - for example, as was discussed in the previous example k-ary tree. In this case, one of the methods of compression of the increasing sequence of non-negative integers considered in [10] can be applied. Such methods are based on the fact that instead storage of real number, the difference between the number and previous number, which is the (presumably small) positive number, is stored. This difference is encoded using one of coding methods, like  $\gamma$ - or  $\sigma$ -encoding [10]. The disadvantage of this method is that after its application, key fields can only be accessed for sequential viewing, since the previous value is required to decode the next value. In addition, the coded numbers can take a non-integer number of bytes, and access to them requires a relatively slow on modern processors bit shift instructions and bitwise AND. As a

means of correcting the last drawback, you can align the encoded numbers to byte boundaries, but this reduces the effect of compression [11].

Another method of compression, which is used in situations where key fields are large and variable in length (in particular, are string values), is the separation from each key value of the prefix of fixed length. Long key node in this case is replaced by a prefix and a pointer to the actual key:  $\text{key}_i \Rightarrow (\text{prefix}_i, \text{key\_pointer}_i)$  (note that the  $\text{key\_pointer}_i$  is the information field although it contains a pointer to a string). The reason for this is the fact that in many cases, instead of a key, you can use a prefix - for example, when comparing strings, you can compare prefixes, and refer to the lines themselves only if the prefixes match.

*Example 1.* Consider again the node of 5-ary tree  $\text{Node} = (1980, 1998, 2002, 2004, 2005, \text{ptr}_0, \text{ptr}_1, \text{ptr}_2, \text{ptr}_3, \text{ptr}_4, \text{ptr}_5)$ . Using compression integers, we transform the sequence of key values (1980, 18 4 2 1). When using coding integers alignment is byte for each key, starting with the second is 1 byte, so all we have saved  $4 \cdot (\text{sizeof}(\text{int}) - 1) = 12$  bytes.

*Example 2.* Let us assume key values as strings. Consider the 3-ary tree node:

$\text{Node} = ('Ivanovsky', 'Lebedev', 'Pavlovsky', 'Petrovsky', \text{ptr}_0, \text{ptr}_1, \text{ptr}_2, \text{ptr}_3, \text{ptr}_4)$ . Instead of rows will use the two-letter prefixes:  $\text{key}_i \Rightarrow (\text{prefix}_i, \text{key\_pointer}_i)$ . We'll get:  $\text{Node} = ('Iv', 'Le', 'Pa', 'Pe', \text{key\_ptr}_1, \text{key\_ptr}_2, \text{key\_ptr}_3, \text{key\_ptr}_4, \text{ptr}_0, \text{ptr}_1, \text{ptr}_2, \text{ptr}_3, \text{ptr}_4)$ . As a result, we can save  $3 \cdot 9 + 7 - (3 \cdot 2 + 3 \cdot 4) = 16$  bytes per node, in addition, comparing short prefixes is faster than the original rows.

*Removing pointers.* Typically, pointer fields are not used in getNext function - algorithm for the next iteration of the loop is addressed to only one of the node pointers, which is selected as a result of getNext call. By decreasing the number of pointers, you can reduce the number of cache blocks occupied by the node. Note that pointers are only needed to maintain links in the structure, so they can be replaced by another way of maintaining these links - in particular, their calculation. As a rule, the calculation of links is based on the fact that, if there is a certain regularity in the structure, the address of the child node can be calculated from the address of the parent node and the serial number of the child node among the child nodes of the parent node.

*Example 1.* Consider k-ary tree,  $\text{Node} = (\text{key}_0, \dots, \text{key}_k, \text{ptr}_1, \dots, \text{ptr}_k)$ . Assume that all child nodes are arranged in a sequence memory so  $\text{ptr}_i = \text{ptr}_0 + i \cdot \text{sizeof}(\text{Node})$ . In this case the assembly can be removed from all pointers except  $\text{ptr}_0$ , thus saving  $(k - 1) \cdot \text{sizeof}(\text{ptr}) = 4(k - 1)$  bytes within the cache block. Of course, this requires special support for the modification of the tree, in particular, when adding new nodes and increases the total amount of memory occupied by the tree.

*Example 2.* In conditions of the previous example, suppose that the tree is complete and stored in memory so that all levels and all nodes on the same level are arranged sequentially. Offset of unit  $i$  at level  $l$  ( $l > 1$ ) comparatively to the tree  $\text{addr}_0$  in memory in this case is equal to  $(1 + k + \dots + k^{l-1} + i) \cdot \text{sizeof}(\text{Node})$ . This makes it possible, knowing the address of current in the algorithm 1, to determine the level  $l$  of unit current, that allows you to find the address of the child node, knowing its serial number among child nodes current. In order to avoid calculating the level  $l$  at each iteration of the algorithm 1, it can be

maintained as an additional  $i$  loop variable, increasing it at each iteration. Thus, in this situation, storage of pointers to child nodes is not required at all, however such a data structure is unsuitable if new keys can be added to the tree.

#### MUTUAL ARRANGEMENT OF NODES

In addition to optimizing the structure of nodes, optimization of the relative location of nodes can also be used. Typically, this optimization is performed after the node structure is improved because only small node size makes it possible to successfully optimize the relative location of the nodes. The purpose of such optimization is to reduce number of cache misses in each iteration of the algorithm 1. Obviously, the way to achieve this is to place the nodes so that the current and new nodes *current* are located within a single cache block. Of course, this optimization is possible only for small node size:  $\text{sizeof}(\text{Node}) < (\text{segment size})/2$ . In addition, restrictions on the ability of such optimization imposes *getNext* function - if the current node can transition to many following sites (such as B-tree heavily enough, where each node can be dozens of child nodes), then place the current and all possible next nodes in one cache block will fail.

*Example.* Consider a 3-ary tree whose node key has two integer values:  $\text{Node} = (\text{key}_0, \text{key}_1, \text{ptr}_0, \text{ptr}_1, \text{ptr}_2)$ .  $\text{sizeof}(\text{Node}) = 2 * 4 + 3 * 4 = 20$  and segment size is 128. Then we can group the nodes of even levels with their child nodes, so that the node and all its child nodes are stored sequentially in memory. In Fig. 1, the groups of nodes are shown using dotted lines. The number of cache misses when searching for a key in such a tree is reduced by an average of 2 times, since cache misses do not occur when moving from an even level to an odd level. However, for a similar 4-ary tree where  $\text{sizeof}(\text{Node}) = 3 * 4 + 4 * 4 = 28$ , the node and its children will not fit in a single cache block. Nevertheless, described optimization is still effective in this case, since the probability of use *getNext* that part of the

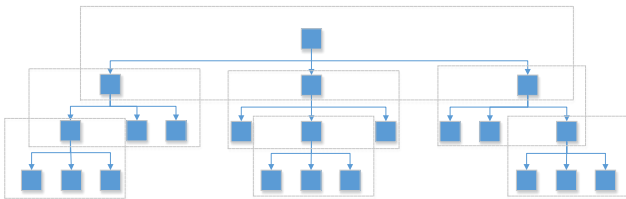


Fig. 1. Grouping of 3-ary tree nodes.

last node that does not fit in the cache block is small.

#### CONCLUSION

Showed the set of methods, which can be used in distributed in-memory data storage to reduce the number of cache misses during the index search. All of them requires refactor of index structure, which may be not suitable for existing data storages with large volume of legacy code and necessity to provide the backward compatibility with old versions. However, usage of optimized index structured can be good solution for ne in-memory data grids, because reducing of cache misses significantly decreases the usage of RAM, which is much slower than processor cache.

#### REFERENCES

- [1] Busowski O., Podrubailo O. Actual problems of in-memory data grid. // News of the National Aviation Administration. Problems of information and management - 2015 - №2(50) 2015 - P. 36-43 (in Ukrainian)
- [2] Shaporenkov D. Effective methods of data indexing and querying in the database control systems in main memory – PhD research - St. Petersburg, 2006 (in Russian)
- [3] Rao J., Ross K. A. Cache Conscious Indexing for Decision-Support in Main Memory. // Proc. of the VLDB 1999 Conference. — Morgan Kaufmann, 1999. - P. 78-89.
- [4] Rao J., Ross K. A. Making B -Trees Cache Conscious in Main Memory. // Proc. of the SIGMOD 2000 Conference. - ACM, 2000. - P 475-486.
- [5] Nonlinear Array Layouts for Hierarchical Memory Systems. / S. Chatterjee, V. V. Jain, A. R. Lebeck et al. // Proc. of the ICS 1999 Conference. - 1999. - P. 444-453.
- [6] Chatterjee S., Sen S. Cache-Efficient Matrix Transposition. // Proc. of the HPCA 2000 Conference. - 2000. - P. 195-205.
- [7] Chilimbi T. M., Parus J. R. Using Generational Garbage Collection To Implement Cache-Conscious Data Placement. // Proc. of the ISMM 1998 Conference. - 1998. - P. 37-48.
- [8] P. Patros, D. Dilli, K. B. Kent, M. Dawson, T. Watson, "Multitenancy benefits in application servers", Proceedings of the 25th Annual International Conference on Computer Science and Software Engineering, P. 111-118, 2015.
- [9] A. Sabeeh et al., "A hybrid intelligent approach for optimising software-defined networks performance", Proc. ICICM, P. 47-51, Oct. 2016.
- [10] Witten I., Moffat A., Bell T. Managing Gigabytes : Compressing and Indexing Documents and Images. — 2nd edition. — Morgan Kaufmann, 1999.
- [11] Data Compression in Full-Text Retrieval Systems. / T. C. Bell, A. Moffat, G. Nevill-Manning et al. // -IASIS. — 1993. — Vol. 44, no. 9. — P. 508-531

Рецензент: д.т.н., проф. КИІ ім. Ігоря Сікорського  
О.В. Бузовський



# Порівняння фреймворків машинного навчання

Дорогий Ярослав  
КПІ ім. Ігоря Сікорського  
Київ, Україна  
cisco.rna@gmail.com

Левченко Ксенія  
КПІ ім. Ігоря Сікорського  
Київ, Україна  
ksulevchenko95@gmail.com

**Анотація.** Тези доповіді містять порівняння існуючих фреймворків машинного для мови Python. В роботі коротко описано переваги та недоліки обраних фреймворків. Авторами були дані рекомендації щодо використання кожного з фреймворків.

**Ключові слова:** розпізнавання зображень, машинне навчання, згорткові нейронні мережі, фреймворки машинного навчання.

## Вступ

У наш час є актуальною проблема побудови нейронних мереж, адже вони використовуються у таких галузях як медична діагностика (виявлення вад на рентгенівських знімках), розпізнавання текстів, мови, технічна діагностика, системи контролю.

Нейронні мережі дозволяють розв'язувати складні задачі класифікації, у яких без автоматизації було б задіяно велику кількість людських ресурсів. Це такі галузі як медична діагностика (виявлення вад на рентгенівських знімках), розпізнавання текстів, мови, технічна діагностика, системи контролю.

Для побудови схожої власної системи (або для початку побудови таких систем) дуже зручно використовувати вже готові рішення, які значно понижують поріг входження у галузь розробки систем розпізнавання. На сьогодні існує велика кількість фреймворків, що полегшують реалізацію системи на основі нейронних мереж та дозволяють сфокусуватись на архітектурі системи, на задачах, що вона розв'язує, а не на розробці конкретного програмного забезпечення.

## ПОСТАНОВКА ЗАДАЧІ

Спочатку визначимо задачу, на прикладі якої буде проводитись порівняння найпопулярніших

фреймворків машинного навчання. Нехай, нам необхідно побудувати систему розпізнавання певних об'єктів на зображеннях. Оберемо такі об'єкти, що присутні у публічних датасетах, наприклад об'єкти з CIFAR-10 датасету [1]. Оскільки це задача розпізнавання зображень, то для своєї системи оберемо convolutional neural network (згорткову нейронну мережу) [2], що найкраще справляється з таким типом задач через особливості своєї архітектури. Мову програмування обрано Python, оскільки для неї фреймворки представлені найширше.

Виходячи з постановки нашої задачі можна виділити такі критерії порівняння: простота опису архітектури нейронної мережі, підтримка роботи із згортковими нейронними мережами, простота переналаштування архітектури мережі, що є дуже важливим фактором, оскільки швидке переналаштування мережі дозволяє легко проводити експерименти з різними варіантами нейронних мереж можливість навчання на графічному процесорі (а саме підтримка платформи CUDA [3]), доступність прикладів, підтримка попередньо натренованих моделей, які дозволяють покращити результати навчання, підтримка різних ОС.

## ПОРІВНЯННЯ ФРЕЙМВОРКІВ

Для порівняння було обрано такі фреймворки: PyTorch [4], TensorFlow [5], Caffe [6], MxNet [7].

PyTorch. Це фреймворк, який дозволяє проводити обчислення швидко на графічному процесорі та будувати нейронні мережі. Переваги: підтримка попередньо натренованих моделей, швидка зміна архітектури нейронної мережі

Таблиця 1

Порівняння фреймворків за обраними критеріями

Фреймворк	Архітектура	Переналаштування	Підтримка попередньо натренованих моделей	Наявність прикладів	Тренування	ОС	Паралельні обчислення
PyTorch	Методи, що мають назву шарів CNN	Auto-differentiation	Присутня велика кількість моделей, легке тренування	Представлені на сайті фреймворку, репозиторії на github	Треба писати самому	Windows, Linux	+
TensorFlow	Методи, що мають назву шарів CNN	Auto-differentiation	Не так розповсюджено	Представлені на сайті фреймворку, репозиторії на github, велика кількість окремих статей	Готові методи	Windows, Linux	+

CAFFE	Методи, що мають назву шарів CNN; JSON структура	Модель має бути перенатренована	Фреймворк націлений саме на fine-tuning	Репозиторії на github	Не потрібен код, тільки налаштування	Linux, формальна підтримка Windows	-
MXNet	Методи, що мають назву шарів CNN	Auto-differentiation	Присутня велика кількість моделей, легке тренування	Представлені на сайті фреймворку, репозиторії на github	Готові методи	Windows, Linux	+

Таблиця 2

Оцінка кожного фреймворку за обраними критеріями

Фреймворк	Архітектура	Переналаштування	Підтримка попередньо натренованих моделей	Наявність прикладів	Тренування	ОС	Паралельні обчислення	Сума (max = 35)
PyTorch	4.5	5	4	4	3	5	5	30.5
TensorFlow	4.5	5	3	5	5	5	5	32.5
CAFFE	5	0	5	3.5	5	4	0	22.5
MXNet	4.5	5	4.5	4	5	5	5	33

завдяки техніці reverse-mode auto-differentiation (автоматичне диференціювання з рухом назад), підтримка візуалізації побудованої архітектури, опис моделі за допомогою методів, що мають назви шарів згорткової нейронної мережі, підтримка візуалізації побудованої архітектури, опис моделі за допомогою методів, що мають назви шарів згорткової нейронної мережі (conv, relu, max\_pool), що пришвидшує початок роботи з цим фреймворком, підтримка CUDA. До недоліків можна віднести те, що код для тренування необхідно писати самому, що сповільнює розробку [8].

TensorFlow. Фреймворк від Google, який замінив Theano, став простішим і швидшим на відміну від попередника, є найпопулярнішим серед розробників. Переваги: архітектура мережі описується у термінах, що використовуються у CNN, має детальну візуалізацію TensorBoard, що будує граф під час навчання, готова реалізація тренування, використовує auto-differentiation для переналаштування архітектури, підтримує паралельні обчислення, підтримка CUDA. Недоліки: значно повільніший за інші фреймворки (наприклад, MxNet), невелика кількість попередньо натренованих моделей, складний для початківця [9].

Caffe. Фреймворк, який спеціалізується саме на розпізнаванні зображень. Переваги: гарна підтримка CNN, велика кількість попередньо натренованих моделей, опис моделей відбувається за допомогою JSON, що дозволяє краще розуміти вкладеність та структуру шарів без візуалізації, або за допомогою вбудованих методів, присутня візуалізація, типи шарів описуються у відомих термінах для CNN, тренування проводиться без написання додаткового коду, лише за допомогою файлів налаштувань, підтримка CUDA. Недоліки: важко розширювати для виконання інших задач розпізнавання повільно працює з великими мережами, які не були попередньо натреновані, відсутність швидкого переналаштування, навчання

проводитиметься спочатку, майже відсутня підтримка [10].

MXNet. Фреймворк для машинного навчання від Microsoft і Amazon. Переваги: набагато швидший за інші фреймворки (швидше навчання, менше використання оперативної пам'яті), підтримує попередньо натреновані моделі (у прикладах приступний готовий код для fine-tuning), опис моделі проводиться у термінах CNN, присутня візуалізація, можливе швидке переналаштування завдяки auto-differentiation, готовий код для тренування, підтримка CUDA. Недоліки: необхідно встановлювати окремі версії фреймворка для GPU та CPU, невелика кількість документації [11].

Основні критерії порівняння винесені в табл. 1. Варто додати, що усі фреймворки підтримують навчання на GPU за допомогою CUDA, та мають візуалізацію побудованої мережі.

В результаті порівняння кожного фреймворку за обраними критеріями, їх було оцінено по калі від 0 до 5 (від найгіршого результату до найкращого). Результат цієї оцінки представлено в табл. 2.

Кожен критерій також було оцінено по шкалі від 0 до 5 (від зовсім не важливого, до дуже важливого) з точки зору важливості для прийняття рішення, який фреймворк використовувати. Результати представлено в табл. 3.

Таблиця 3

Оцінка критеріїв за їх важливістю

Критерій	Оцінка (0-5)
Архітектура	5
Переналаштування	5
Підтримка попередньо натренованих моделей	3
Наявність прикладів	3

Тренування	4
ОС	2
Паралельні обчислення	4

#### ВИСНОВКИ

Порівняння, що приведено вище полегшує попередній вибір фреймворку для розв'язку задач, які є подібними до поставленої. В результаті оцінки кожного фреймворку, за обраними критеріями було отримано, що MXNet є найкращими фреймворком, а Caffe – найгіршим. MXNet є наймолодшим представником, включає у себе необхідні для розробника компоненти, швидко розвивається, тому і отримав найкращий результат. Caffe навпаки, хоч і використовується досі, є застарілим фреймворком, тому деякі моменти його реалізації не є актуальними.

Звичайно, що кожен розробник особисто обирає важливість того чи іншого критерію, в залежності від умов поставленої задачі, обчислювальних можливостей, необхідності часто змінювати і перенавчати архітектуру нейронної мережі. Якщо було обрано техніку навчання fine-tuning, то не варто обирати TensorFlow, адже це не основна задача цього фреймворку, в цьому випадку Caffe має перевагу. Якщо важлива швидкість навчання і перенавчання мережі, то краще звернути увагу на MxNet та PyTorch, адже вони можуть забезпечити цю швидкість. Якщо задача складна, не вистачає готових рішень, потрібна підтримка професіоналів у сфері машинного навчання, то можна обрати TensorFlow як найрозповсюдженіший фреймворк, що означає, що легше буде отримати певну допомогу при вирішенні задачі.

#### ЛІТЕРАТУРА

1. CIFAR-10 and CIFAR-100 datasets [Електронний ресурс]: [Веб-сайт]. – Режим доступу: <https://www.cs.toronto.edu/~kriz/cifar.html> (дата звернення 30.04.2018). – Назва з екрана.
2. CS231n: Convolutional Neural Networks for Visual Recognition [Електронний ресурс]: [Веб-сайт].

– Режим доступу: <http://cs231n.github.io/convolutional-networks/> (дата звернення 30.04.2018). – Назва з екрана.

3. CUDA Zone | NVIDIA Developer [Електронний ресурс]: [Веб-сайт]. – Режим доступу: <https://developer.nvidia.com/cuda-zone> (дата звернення 30.04.2018). – Назва з екрана.

4. PyTorch [Електронний ресурс]: [Веб-сайт]. – Режим доступу: <http://pytorch.org/> (дата звернення 30.04.2018). – Назва з екрана.

5. TensorFlow [Електронний ресурс]: [Веб-сайт]. – Режим доступу: <https://www.tensorflow.org/> (дата звернення 30.04.2018). – Назва з екрана.

6. Caffe | Deep Learning Framework [Електронний ресурс]: [Веб-сайт]. – Режим доступу: <http://caffe.berkeleyvision.org/> (дата звернення 30.04.2018). – Назва з екрана.

7. MXNet: A scalable deep learning framework [Електронний ресурс]: [Веб-сайт]. – Режим доступу: <https://mxnet.incubator.apache.org/> (дата звернення 30.04.2018). – Назва з екрана.

8. Adam Paszke Automatic differentiation in PyTorch/ A. Paszke, S. Gross // 31st Conference on Neural Information Processing Systems (NIPS 2017) – Long Beach, CA, USA, 2017. – С. 1 -3

9. Martin Abadi TensorFlow: A system for large-scale machine learning/ M. Adabi, P. Barham // 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), USENIX Association (2016) – SAVANNAH, GA, USA, 2016. – P. 1 - 5.

10. Yangqing Jia Caffe: Convolutional Architecture for Fast Feature Embedding/ Y. Jia, E. Shelhamer // ACM MULTIMEDIA 2014 OPEN SOURCE SOFTWARE COMPETITION – 2014 – P. 2 – 3.

11. Tianqi Chen MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems/ T. Chen, M. Li // In Neural Information Processing Systems, Workshop on Machine Learning Systems – 2016 – P. 2 – 5.

# Ситуационное управление городским транспортом с интеллектуальной поддержкой диспетчерских решений

Стенин А.А., Пасько В.П., Лемешко В.А., Русакова А.В.

КПИ им. Игоря Сикорского  
Киев, Украина  
alexander.stenin@yandex.ua

**Аннотация.** Предлагается алгоритм нечеткого ситуационного управления, который является основой интеллектуальной АСППР для диспетчера по управлению конкретным видом городского транспорта. Алгоритм базируется на методе ПРОБА, позволяющим осуществить выбор лучшей из группы альтернатив управляющих воздействий, сформированных на основе метода экспертного анкетирования.

**Ключевые слова:** управление городским транспортом, база знаний предметной области, нечеткая ситуационная сеть, АСППР.

На сегодняшний день развитие и эффективное управление транспортом невозможно без разработки и применения интеллектуальных систем управления [1, 2, 4]. Для обеспечения высокой эффективности и стабильности принимаемых диспетчером решений в процессе управления движением общественного городского транспорта требуется разработка интеллектуальной автоматизированной системы поддержки принятия решений (АСППР) [1, 4].

В докладе предлагается алгоритм нечеткого ситуационного управления, построенный на основе нечеткой ситуационной сети, который является основой интеллектуальной АСППР для диспетчера по управлению конкретным видом городского транспорта и базируется на методе ПРОБА.

Нечеткая ситуационная сеть строится на основе существующей базы знаний АСППР о функционировании общественного городского транспорта и условий его работы в различных режимах эксплуатации с формированием типовых «штатных» ситуаций, для которых на основе экспертных методов детально разработаны управляющие воздействия [2]. Следует отметить, что, как база знаний по функционированию городского транспорта, так и ситуационная сеть, носят эволюционный характер и периодически обновляются за счет включения новой типовой («штатной») ситуации (фактически это отработанный экспертами вариант возникшей «нештатной» ситуации). Фактически, нечеткая ситуационная сеть представляет собой нечеткий ориентированный граф, вершины которого соответствуют существующим «штатным» нечетким ситуациям, дуги взвешены возможными решениями, необходимыми для перехода по ситуациям, и степенями предпочтения этих решений.

Для определения текущего состояния процесса работы городского транспорта необходимо сравнить

данную нечеткую ситуацию с каждой нечеткой ситуацией из некоторого набора существующих типовых ситуаций  $S = (S_1, \dots, S_K)$  в базе знаний о функционировании городского транспорта.

В качестве меры для определения степени близости нечеткой ситуации  $S_{ТЕК}$  и  $S_k$  ( $k = 1, K$ ) из базы знаний можно использовать:

- степень нечеткого включения нечеткой ситуации  $S_{ТЕК}$  в нечеткую ситуацию  $S_k$ ;
- степень нечеткого равенства  $S_{ТЕК}$  и  $S_k$ ;
- степень нечеткой общности  $S_{ТЕК}$  и  $S_k$ .

Такой подход на основе нечеткого ситуационного управления целесообразно использовать в связи с большим количеством неопределенностей и получением количественной и вербальной информации о возникшей ситуации в работе городского транспорта. Алгоритм нечеткого ситуационного управления для диспетчера по управлению конкретным видом городского транспорта при наличии «штатных» ситуаций позволяет диспетчеру всегда выбрать правильное решение по управлению городским транспортом. В случае возникновения «нештатных» ситуаций, на основе разработанного авторами метода ПРОБА, формируются новые «штатные» ситуации и заносятся в базу знаний о функционировании городского транспорта.

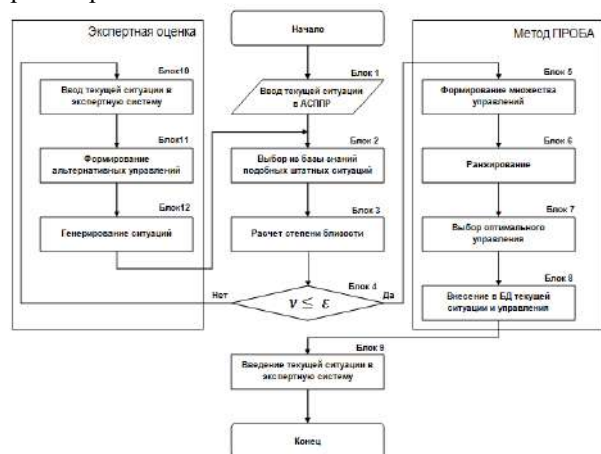


Рис. 1. Алгоритм нечеткого ситуационного управления

Предлагаемый метод ПРОБА (Процедура Обработки Альтернатив) отличается от известных методов

ПАРК, ЗАПРОС, ОРКЛАСС, ШНУР тем, что осуществляется критериальное оценивание альтернатив, причем по каждому критерию оценивается все множество альтернатив. Кроме того, данный метод предусматривает то, что экспертом задаются оценки как на лингвистической шкале оценок, так и на количественной шкале с использованием функций полезности.

Основными этапами метода являются:

1. построение системы гипотез-оснований и гипотез-следствий;
2. формирование количественных и лингвистических шкал критериев оценок альтернатив;
3. составление структуры и заполнение анкеты;
4. нормирование шкал критериев оценок альтернатив с использованием функций принадлежности;
5. анализ результатов экспертной оценки.

В рамках выбранных шкал критериев оценок альтернатив функции принадлежности количественно градуируют принадлежность элементов множества альтернатив принятия того или иного решения некоторому множеству с нормированными переменными в универсальной шкале  $[0, 1]$ . В этом случае нормирование оценок сравниваемых альтернатив осуществляется следующим образом:

– для всех количественных оценок находится значение  $max$  и  $min$  значения рассматриваемой переменной;

– для всех лингвистических (качественных) оценок определяется максимальный и минимальный номер;

– определяются значения оценок альтернатив по универсальной шкале  $[0, 1]$  согласно формулам выбранных функций принадлежности.

Значения оценок альтернатив по номинальной шкале формируются как:

$$v'_{ij} = \mu_{ij},$$

где  $i$  – номер альтернативы,  $j$  – индекс значения количественной или лингвистической шкалы.

Сумму оценок по  $s$ -ой альтернативе можно вычислить посредством сложения суммы оценок по каждому  $j$ -му вопросу, которые суммируются по всем вопросам  $i$ -го блока и по всем блокам анкеты.

$$r_s = \sum_{S=1}^n \sum_{i=1}^m \sum_{j=1}^k v'_{ij}.$$

В результате можем сформулировать для исходного множества альтернатив упорядоченное множество их рангов  $R = \{r_1, r_2, \dots, r_n\}$ . Окончательный выбор решения остается за ЛПР. В случае необходимости учета важности отдельного критерия необходимо ввести коэффициенты веса, которые можно определить на основе известных методов экспертных оценок [4].

#### ЛИТЕРАТУРА

1. Кригер Л. С. Интеллектуальная система поддержки принятия решений при управлении движением общественного транспорта / Кригер Л. С. // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – Астрахань: 2012, №2. сс.150-155
2. Кригер Л. С. Формализация типовых ситуаций в задачах управления движением общественного транспорта/ Л. С. Кригер, И. Ю. Квятковская / Научно-технические ведомости СПбТУ №3(150). - Санкт-Петербург: 2012, сс. 106-110.
3. Мелихов А.Н. Ситуационные советующие системы с нечеткой логикой / Мелихов А.Н., Бернштейн Л.С., Коровин С.Я. – М.: Наука, 1990, 272 с.
4. Стенин А.А. Автоматизация процесса принятия инновационных решений в социотехнических системах. / Стенин А.А., Ткач М.М., Пасько В.П., Губский А.Н. Проблемы информационных технологий.– Херсон, ХНТУ, 2016. №19. сс.51-57.

# Розширення області використання двигунів Стірлінга для підтримання альтернативного енергопостачання житла

Огульчанський Федір Ярославович  
КПІ імені Сікорського  
Київ, Україна  
fedor.ohul@gmail.com

Репнікова Наталія Борисівна  
КПІ імені Сікорського  
Київ, Україна  
n.repnikova@gmail.com

**Анотація.** В роботі обґрунтовується можливість сумістити сонячну енергію з двигуном Стірлінга для ефективного енергопостачання житла.

**Ключові слова:** двигун Стірлінга, побутове використання, відновлювана енергія.

## ВСТУП

В сучасному світі існує нагальна проблема економії природних ресурсів та підтримання технологій, що не забруднюють навколишнє середовище. Альтернативна енергетика – це сукупність способів отримання електроенергії з альтернативних джерел, зазвичай з тих, що містять відновлювану енергію.

Сутність запропонованого рішення полягає в тому, що енергія сонячних променів, перетворена у теплову енергію за допомогою сонячного колектора, може бути використана для генерування електричної енергії за допомогою двигуна Стірлінга. Цей двигун здатен використовувати будь-який перепад температур для створення механічної енергії і в даній системі під'єднання його до електрогенератора може бути вигідною альтернативою використанню в приватному житлу централізованого енергопостачання.

## ПОСТАНОВКА ТА РІШЕННЯ ЗАДАЧІ

В даній роботі пропонується структурна схема ефективного енергопостачання приватного житла з конфігуруванням сонячних колекторів та двигуна Стірлінга. Необхідно досягти постачання електроенергії з частотою 50 Гц і напругою 220 В [1].

## ДВИГУН СТИРЛІНГА

Як відомо, двигун Стірлінга – це тепловий двигун, який працює завдяки циклічному розширенню та стисненню повітря або інших газів між двома камерами із різними температурами, де воно по черзі нагрівається та охолоджується, перетворюючи теплову енергію у механічну роботу. Через це тиск робочої речовини на робочий поршень або збільшується, або зменшується, в залежності від такту роботи двигуна [2].

Існують три класичні загальноприйняті конфігурації двигуна Стірлінга –  $\alpha$ -конфігурація,  $\beta$ -конфігурація та  $\gamma$ -конфігурація. За довгий час існування двигуна інженери розробили і інші конфігурації. Кожна з них має особливості будови, переваги та недоліки. В роботі розглядатиметься

двигун Стірлінга  $\beta$ -конфігурації (Рис.1) через переваги у відсутності проблеми перегріву ізоляції та через загальну ефективність його роботи [3].

Двигун Стірлінга – це двигун зовнішнього згорання, який працює за замкненим термодинамічним циклом Стірлінга. В ідеальному випадку ефективність цього циклу досягає ефективності циклу Карно [4] – два ізотермічні процеси і два адіабатні процеси, що поперемінно чергуються між собою. Теоретично досяжний ККД двигуна Стірлінга розраховується, як ККД циклу Карно:

$$\eta = \frac{T_1 - T_2}{T_2}$$

Під час ізотермічного розширення виконується робота, під час адіабатного процесу йде перерозподілення теплоти. Загальну роботу за один цикл в ідеальному двигуні Стірлінга можна розрахувати за такою формулою:

$$A = \nu R P V (T_2 - T_1) \ln(T_2 / T_1),$$

де  $\nu$  – число молей газу,  $R$  – універсальна газова константа,  $P$  – робочий тиск,  $T_2$  – температура нагрівача,  $T_1$  – температура холодильника.

## СОНЯЧНИЙ КОЛЕКТОР

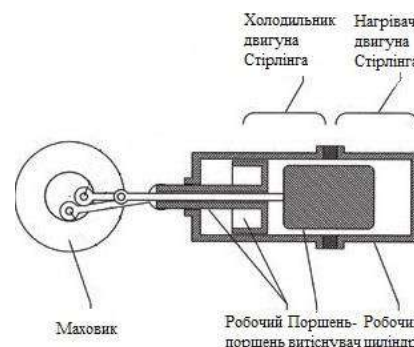


Рис.1. Спрощена схема двигуна Стірлінга  $\beta$ -конфігурації

Оскільки, для роботи двигуна Стірлінга необхідна подача тепла для створення різниці температури, в схемі буде використовуватися енергія Сонця. Для акумулювання цієї енергії у вигляді тепла використовується сонячний колектор.

Це пристрій, в якому сонячне випромінювання нагріває теплоносії, який знаходиться за ізолюючим

шаром та циркулює по системі, віддаючи тепло споживачу і повертаючись до колектора охолодженим. Сонячний колектор має високі теплоізолюючі якості і тому температура теплоносія (води) в ньому може досягати 60-100 °С [5].

#### СТРУКТУРНА СХЕМА МОДЕЛІ СИСТЕМИ

Нехай задана динаміка роботи системи енергопостачання на базі двигуна Стірлінга та сонячного колектора описується моделлю з такою структурною схемою (Рис.2):

На схемі зображені такі елементи та зв'язки:

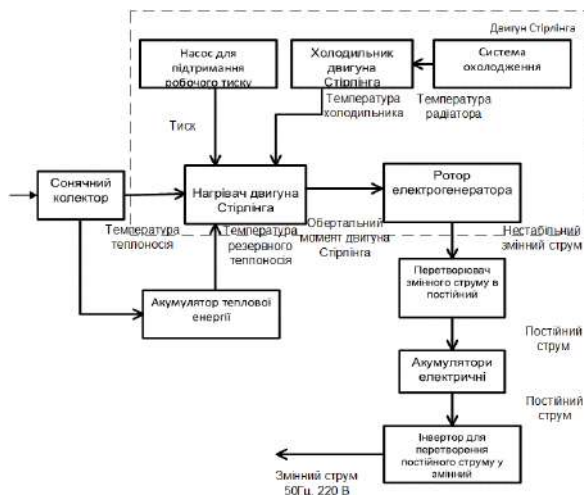


Рис.2. Схема структурна системи

Сонячний колектор це пристрій для збору енергії сонячного випромінювання. Теплоносієм виступає вода, яка циркулює по трубах та передає тепло в акумулятор теплової енергії. Це тепло підводиться до двигуна Стірлінга через тонкі стінки робочого циліндру. Нагрівач двигуна це та його частина, що отримує зовнішню теплову енергію, що створює підвищений об'єм в циліндрі під поршнем-витіснувачем, рух якого призводить до руху робочого циліндру, який створює механічну роботу в процесі обертання маховика. В циліндрі має підтримуватися високий тиск – це відбувається завдяки використанню спеціального насоса. Поршень-витіснувач в процесі роботи двигуна витісняє гаряче повітря до охолоджувальної камери через регенератор. Для підтримання низької температури холодильник двигуна знаходиться поза приміщенням та оснащений повітряною охолоджувальною системою. Механічна

енергія обертання двигуна Стірлінга передається на електрогенератор, який створює змінний струм різної частоти та напруги, в залежності від потужності двигуна в окремий момент. За допомогою перетворювача цей струм перетворюється у постійний та за допомогою нього заряджаються електричні акумулятори. Для побутового використання електроенергії, постійний струм з акумулятора на інверторі перетворюється у змінний струм заданої частоти та напруги.

#### ВИСНОВКИ

Запропонована система дійсно може задовольняти поставлені задачі і має низку переваг в порівнянні з використанням систем електрогенерування, побудованих на двигунах внутрішнього згорання. По-перше, з причин відсутності палива, що згорає, система не забруднює довкілля. По-друге, двигун Стірлінга дуже тихий і тому комфортний для використання на ділянці. По-третє, двигун надійний і потребує невелику кількість обслуговування, а інші елементи системи окупаються за час експлуатації. Перевагами системи перед системами, основаними на вітрогенераторах та сонячних панелях є загальна універсальність води, що нагрівається завдяки сонячному колектору. Враховуючи, що тепло з колектора може бути використане не лише для генерації електроенергії, а й для опалення та, власне, гарячої води для щоденного використання, то система може бути конкурентоспроможною для використання в побути на півдні України.

#### ЛІТЕРАТУРА

1. U.S. Department of Commerce. Electric Current Abroad [Електронний ресурс] / U.S. Department of Commerce // International Trade Administration. – 2002. – Режим доступу до ресурсу: <http://www.webcitation.org/66hSp0MIT>.
2. G. Walker. Stirling Engines / G. Walker. – Oxford: Clarendon Press, 1980.
3. M. Keveney. Two Cylinder Stirling Engine / M. Keveney., 2000.
4. T. Finkelstein. The History, Science, and Reality of the Perfect Engine / T. Finkelstein, A.J. Organ., 2001.
5. F. Kamel Abdalla. Optimum Operating Temperature for Evacuated Tube Solar Collectors [Електронний ресурс] / F. Kamel Abdalla, Paul Wilson – Режим доступу до ресурсу: [https://eprints.usq.edu.au/4087/1/Kamel\\_ANZSES\\_2002.pdf](https://eprints.usq.edu.au/4087/1/Kamel_ANZSES_2002.pdf).



# Проектування діяльності користувача людино-машинного інтерфейсу АСУ ТП

Юрчук Леонід  
КПІ ім. Ігоря Сікорського  
Київ, Україна  
leonidyu0@gmail.com

*Анотація. Розглядаються питання проектування діяльності користувача людино-машинного інтерфейсу систем управління технологічними процесами відповідно до вимог та рекомендацій міжнародного стандарту ANSI/ISA-101.01-2015 - Human Machine Interfaces for Process Automation Systems.*

**Ключові слова:** людино-машинний інтерфейс, користувач, діяльність, АСУ ТП, навігація

## ВСТУП

Ця стаття є продовженням статей [1, 2] присвячених огляду рекомендацій стандарту ANSI/ISA-101.01-2015 Human Machine Interfaces for Process Automation Systems [3] по принципам проектування людино-машинного інтерфейсу (ЛМІ) і присвячена питанням взаємодії «людина-машина».

У загальному випадку під користувачем ЛМІ розуміються не тільки оператори, що керують технологічним процесом, а і інший інженерний та адміністративний персонал.

Основна мета проектування взаємодії користувача з ЛМІ – забезпечення швидкої та безпомилкової реакції оператора на поведінку системи керування.

У стандарті звертається увага на наступні аспекти взаємодії:

- методи введення даних;
- методи навігації;
- запобігання помилок поза системою обміну повідомленнями безпеки та доступу користувача;
- off-системи обміну повідомленнями;
- безпека доступу користувачів.

Необхідно враховувати що кожний з цих аспектів включає багато конкретних вимог (умов) для їх успішної реалізації.

Рекомендуються основні принципи проектування:

- а) послідовність виконання у всіх режимах взаємодії;
- б) своєчасність зворотного зв'язку при введенні даних і управлінні діями;
- в) оптимізація взаємодії з користувачем (мінімізована кількість виділень або обсяг друку);
- г) використання відповідних засобів привертання уваги для повідомлень про помилки;
- д) обмежене використання складних умовних методів.

Інженерні підрозділи повинні бути доступними для довідки оператора.

## МЕТОДИ ВВЕДЕННЯ ДАНИХ

На етапі «Введення даних» слід використовувати послідовне введення даних. Загальні підходи включають в себе:

- а) введення відповідного поля;
- б) чітке визначення поточного вибору;
- в) визначення кольором (сірим) елементів, які не доступні для запису;
- г) курсор, щоб відобразити елемент, що обирається.

Якщо запис введено у неправильному форматі, він повинен бути відхилений, і оператор повинен отримати повідомлення про помилку у візуальній і/або у звуковій формі.

Меню можливих взаємодій з системою повинно бути представлено на частині екрану або у спливаючому вікні.

При представленні та введенні чисел та тексту вибраний шрифт повинен бути помітний з нормального положення оператора.

Представлення чисел повинно відповідати роздільній здатності десяткового формату, необхідного користувачам для виконання своїх завдань, мати відповідну точність і не більше (менше) значущих цифр, ніж потрібно.

Числа повинні бути представлені у формі безпосередньо придатній для використання. Оператор не повинен проводити обчислення для зміни одиниць фізичних величин.

При представленні негативних та десяткових чисел, тексту необхідно враховувати регіональні особливості використання ЛМІ (напрямоком читання та інше).

Не рекомендується представляти текст великими літерами. Слід уникати аббревіатур та скорочень, якщо вони не є частиною нормальної мови оператора.

Не рекомендується привертати увагу підкресленням, бо воно використовується для гіперпосилань.

Текст повинен бути орієнтовано горизонтально для чіткого сприйняття.

При спілкуванні оператора з системою слід використовувати позитивні твердження, а не негативні висловлювання.

Введення команд повинно відповідати умовам:

- а) всі входи і ефекти, які дана команда відпрацьовує, повинні бути видимі користувачеві, де це можливо;

б) команди, які вимагають декількох дій введення від оператора для забезпечення прямого доступу до процесу повинні мати блокування від можливого спрацювання від одного необережного вхідного впливу;

в) користувачі повинні мати засоби для скасування зміни команди в процесі зміни і/або мати можливість для швидкого відновлення до похідного рівня конфігурації.

При проектуванні елементу «Кнопка» необхідно забезпечити щоб:

а) текстова мітка кнопки була зрозуміла користувачам і бути чітко пов'язана з кнопкою, розміщена на або поруч з нею;

б) якщо текстова мітка пов'язана з динамічними умовами процесу, мітка повинна відображати дію після натискання кнопки, використовуючи просте дієслово або активний голос (наприклад, зупинка-насоса);

в) кнопки, які взаємодіють безпосередньо з процесом повинні візуально відрізнятися від кнопок, які забезпечують навігаційні зв'язки або запуск додатків;

г) повинно бути забезпечено підтвердження для виконання дії кнопки (зворотній зв'язок).

Лицьові панелі та спливаючі вікна повинні відповідати наступним умовам:

а) повинен бути налаштований період тайм-ауту (після чого спливаюче вікно закриється без будь-якої взаємодії з користувачем). Оператор повинен мати доступ до процесу, коли спливаючі вікна спеціального використання закриті, в тому числі можливість змінювати будь-який заданий період тайм-ауту;

б) спливаючі вікна не повинні закривати або приховувати важливі частини відеокадру ЛМІ. Оператор повинен мати можливість перемістити спливаюче вікно.

#### МЕТОДИ НАВІГАЦІЇ

Ефективна і інтуїтивно зрозуміла схема навігації по ЛМІ може безпосередньо впливати на швидкість і точність втручання оператора. Основними принципами розробки навігації є продуктивність, послідовність та інтуїтивність. Залежно від вимог користувача, для забезпечення надійності та швидкого і постійного доступу до відеокадрів має бути передбачено кілька методів навігації.

Засоби навігації включають в себе графічні символи на відеокдрах, клавіатурні кнопки, пункти меню, кнопки панелі інструментів, дерева папок, вкладки та контекстні меню.

Для документування шляху навігації між різними дисплеями може бути доречною схема мережі відеокдрів, де повинні бути відображені основні шляхи переходів, а також інші бічні шляхи, які можуть бути також використані.

Не має єдиної схеми навігації, яка підходила б до всіх процесів.

Схема навігація повинна бути адаптована для задоволення функціональних вимог ЛМІ конкретного виробництва.

При проектуванні навігації слід враховувати наступні принципи:

а) доступ до дисплея повинен бути розроблено так, щоб звести до мінімуму натискання клавіш оператором або еквівалентних дій;

б) оператору не повинно бути необхідним введення ім'я або точки ідентифікації, хоча повинна бути забезпечена підтримка цієї взаємодії;

в) повинні мати послідовне і чітке візуальне кодування символи відображення, які є об'єктами навігації і символи, які можуть бути обрані;

г) відеокадри повинні бути реалізовані в стандартизованій структурі контенту, яка підтримує розкриття докладної інформації. Структура вмісту повинна бути представлена логічним угрупованням, яке інтуїтивно зрозуміле для оператора і підтримує функціональні відносини;

д) ЛМІ повинен підтримувати роботу при нормальних і ненормальних умовах.

У стандарті розглянуті більш докладно методи навігації та приклади схем навігації для різних виробництв.

#### OFF-СИСТЕМИ ОБМІНУ ПОВІДОМЛЕННЯМИ

Off-системи обміну повідомленнями використовуються щоб допомогти операційній команді управляти процесом. Можуть використовуватися як у приміщеннях управління так і за межами диспетчерської для відправка персоналу повідомлень про важливі події.

Доступ до експлуатації екранів з-за меж приміщення виробничого контролю може бути дуже ефективним і часто зустрічається в деяких галузях промисловості. Додатковий користувач може знадобитися в віддалених місцях або при використанні систем з віддалених місць.

Системи обміну повідомленнями повинні мати відповідну надійність для підтримки функціональних вимог, забезпечувати інформаційну безпеку. Як мінімум, ці системи повинні бути автоматично контролюватися на предмет доступності.

#### ЛІТЕРАТУРА

1. Юрчук Л.Ю. Міжнародні нормативні засади людино-машинного інтерфейсу систем управління технологічними процесами. - Summer InfoCom 2017: Матеріали IV Міжнародної науково-практичної конференції, м. Київ, 1-2 червня 2017 р – К.: Вид-во "Інжиніринг", 2017. – с. 66-68.

2. Юрчук Л.Ю. Проектування відеокдрів людино-машинного інтерфейсу АСУ ТП/Юрчук Л.Ю., Бублінський С.М./ - Winter InfoCom 2017: Матеріали V Міжнародної науково-практичної конференції, м. Київ, 1-2 грудня 2017 р – К.: Вид-во "Інжиніринг", 2017. – С. 76, с.35-37.

3. Стандарт ANSI/ISA-101.01-2015 Human Machine Interfaces for Process Automation Systems [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: <https://www.isa.org/store/ansi/isa-10101-2015,-human-machine-interfaces-for-process-automation-systems/41921133>.

4. Hollifield, B. The High Performance HMI Handbook./ Hollifield, B. Oliver, D., Habibi, E., & Nimmo, I. // – PAS,2008 – 218 p.

# Неблокуюча база даних з ACID транзакціями

Кицмен Дмитро Романович  
КПІ ім. Ігоря Сікорського  
Київ, Україна  
dmitrokytsmen@gmail.com

**Анотація.** Для досягнення неблокуючої обробки транзакцій пропонується використання таких алгоритмів обробки транзакцій та структур даних для подальшого збереження нових даних, які суттєво сприяють не тільки оптимізації та прискоренню роботи бази даних, але й гарантують наявність ACID властивостей.

**Ключові слова:** атомарність, узгодженість, ізоляваність, довговічність, доступність, стійкість, транзакція, конкурентний доступ.

## ВСТУП

На даний момент більшість реляційних СУБД використовують механізм управління паралельним доступом з допомогою багатоверсійності (MVCC — Multi Version Concurrency Control) [1], який полягає в наданні кожному користувачу “зліпку” бази даних, який володіє властивістю, при якій зміни, які вніс один користувач, не є видимими для інших користувачів до моменту фіксації транзакції. Головна проблема полягає в тому, що не існує єдиного стандарту для реалізації MVCC, тому кожен вендор трактує його по своєму і програмні реалізації цього механізму на рівних рівнях абстракції за допомогою різних механізмів призводять до утворення вузьких місць в різних реляційних базах даних.

У ситуації, коли система обробляє десятки одночасних транзакцій і має вимоги до несприятливої латентності, системи без блокування є гарним компромісом між складністю розробки та вимогами високого паралелізму [3]. Сервер бази даних для веб-сайту є гарним кандидатом для неблокуючого дизайну. Хоча будь-яка конкретна транзакція може бути заблокована, тим часом для обробки завжди є більше транзакцій, тому процесори ніколи не залишаються без роботи.

## МОДЕЛЬ ЗБЕРЕЖЕННЯ ДАНИХ

Задля простоти, розглянемо бази даних, яка зберігає тільки один об’єкт (рядок) і потім перейдемо до множини рядків (таблиці), а в кінці до множини таблиць (схеми). Бінарне збалансоване дерево було обране як структура даних, котра дозволяє швидко знаходити дані по ідентифікатору рядка. Оскільки така структура даних є гнучкою, то згодом її можна спокійно розширити для виконання інших операцій над базою даних. Об’єкт (рядок) представлений як ідеально збалансоване бінарне дерево, в якого сини містять значення об’єктів.

## ОБРОБКА ТРАНЗАКЦІЙ

Давайте розглянемо конкурентні зміни об’єкту.

Транзакція 1: *UPDATE Person SET age = 23 AND occupation = “Software Engineer”.*

Транзакція 2: *UPDATE Person SET occupation = “Computer Scientist” and name = “DMYTRO”.*

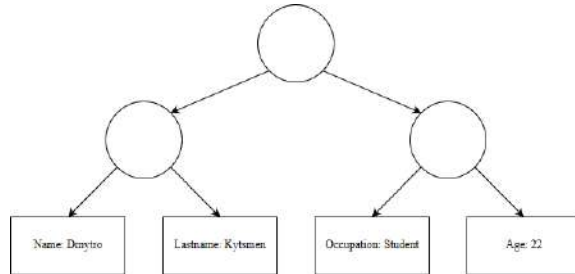


Рис. 1. Схема представлення запису за допомогою дерева

Кожна транзакція створює нове дерево, яке містить тільки значення тих змінних, які вона міняє. Після того, як транзакція надходить, вона додається в кінець ланцюга транзакцій. Транзакції записують та зчитують дані в строго вказаному порядку і не можуть перемішуватись у порядку виконання. Це дозволяє нам гарантувати таку властивість як ізоляція [3].

На рис. 1 показано процес отримання об’єкту, при виконанні транзакції зчитування.

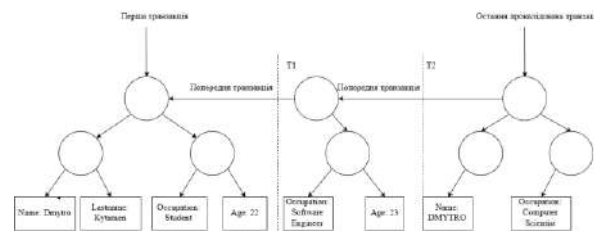


Рис. 2. Схема виконання транзакцій зміни даних

Надалі приведемо приклад запиту, який зчитує дані: *SELECT name, age, occupation FROM Person.*

В блокованому (м’ютекс) підході, потік *T2* очікує поки *T1* завершиться і тільки тоді пише в об’єкт [5]. В цьому випадку кожен з них пише в окрему область пам’яті паралельно і є лише одна невелика синхронізація – атомарна вставка кореня в ланцюг. Потік зчитування може читати стан об’єкту в будь-який момент часу – на початку, після завершення потоку *T1*, після завершення потоку *T2*.

Зчитування складається з наступних етапів:

1. Пошук першої транзакції в ланцюгу згідно з вказаним режимом зчитування.
2. Скопіювати відсутні вказівники на дані, які намагались зчитати в попередній транзакції.
3. Зчитати значення кожного об’єкту за час

$$O(\log N) \quad (1)$$

шляхом переходу від кореня транзакції до листа.

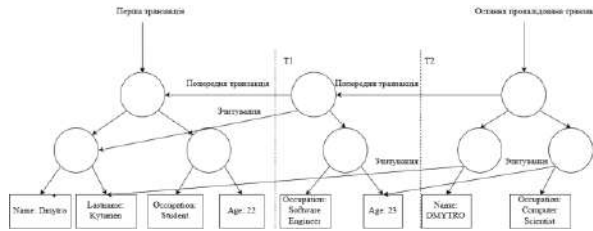


Рис. 3. Схема виконання транзакції зчитування даних

Кожне дерево транзакції містить шляхи до елементів, які воно поміняло. Ми повинні скопіювати посилання з попередніх транзакцій й під час кроку 2 отримати дані. Як тільки покажчики скопійовані, то ми можемо зчитувати будь-яке значення з кореню транзакції за час 1.

Можна зауважити що, незважаючи на одночасну роботу двох потоків з одним записом і його подальшою модифікацією, база гарантує ізоляваність, яка в свою чергу означає неможливість видимості проміжних змін за межами транзакції до її завершення [3].

#### LOCK-FREE АЛГОРИТМ ЗАСТОСУВАННЯ ТРАНЗАКЦІЙ

Враховуючи наведені вище аргументи, сформуємо алгоритм завершення транзакції:

- 1) Створити новий корінь транзакції T.
- 2) Встановити значення стану транзакції ORDERED.
- 3) Атомарно встановити T як новий корінь ланцюга транзакції (сам ланцюг реалізований з використанням неблокуючого однонаправленого списку) [6].
- 4) Побудувати шлях розміром  $\log N$  для кожного зміненого поля об'єкту.
- 5) Встановити значення стану транзакції BUILT.
- 6) Спробувати оновити значення покажчика, який вказує на останню послідовну побудовану транзакцію.
- 7) Валідувати транзакцію, якщо режим зобов'язання - EAGER:
  - а) встановити значення стану VALID і спробувати оновити покажчик, який вказує на останню послідовну провалідовану транзакцію, якщо транзакція успішно провалідована;
  - б) встановити стан транзакції в INVALID, якщо вона не завершилась.

Даний алгоритм обробки транзакцій буде забезпечувати атомарність і гарантуватиме, що транзакція або буде завершена повністю, або жоден з її етапів не матиме впливу, якщо якась помилка при обробці транзакції здійсниться. Також даний механізм гарантуватиме консистентність, оскільки при виконанні транзакції, якщо вона буде успішною, покажчик оновиться і запис представлятиме собою нову структуру даних, якщо ж ні, то стан запису залишиться незмінним [3].

#### ОЦІНКА ЕФЕКТИВНОСТІ

Якщо ми маємо ланцюг транзакцій M і об'єкт (рядок) з N полями, тоді, в найгіршому випадку, часова складність доступу є

$$O(M \log N) \quad (2)$$

ми повинні пройти через всі M дерев. Часова складність може бути зменшена до

$$O(M + \log N) \quad (3)$$

шляхом збереження кореню транзакції в фільтрі Блума [4], який дозволить перевірити чи конкретне поле було змінено в певній транзакції.

Також необхідно розглянути два випадки:

1) Поля з частим доступом. Нехай K - кількість запитів на зчитування конкретного поля, тоді середня вартість поширення посилання

$$\begin{aligned} O((M + \log N) / K) &= \\ &= O(M / K + \log N / K), \end{aligned} \quad (4)$$

де  $M/K$  є відношенням запису до зчитування. Для  $K \sim M$  така вартість є недостатньою.

2) Поля з не частим доступом. Кожен раз ми повинні проходити по ланцюгу поки транзакція з посиланням на поле буде досягнута. Відповідно, необхідно розробити доволі малий алгоритм для збереження M (кількість останніх послідовних дерев з неопрацьованими посиланнями).

В описаному підході збереження/модифікації даних є застереження щодо продуктивності: тільки один потік може додавати транзакцію до ланцюга в один момент часу. Ця проблема може бути вирішена шляхом дозування транзакцій локально кожним потоком:

- 1) Кожен потік підтримує власний ланцюг транзакцій.
- 2) Якщо розмір локального ланцюга наближується до порогового значення або час закінчується, потік атомарно додає свій ланцюг до кінця основного ланцюга.

#### ВИСНОВКИ

В статті наведено розроблені основні механізми, які можуть бути використані для програмної реалізації неблокуючої бази даних. Показано загальні принципи, на основі яких база даних може задовільняти принципам ACID. Описані основні компоненти та структури даних, за допомогою яких можна реалізувати базу даних. Вказано переваги та недоліки обраного підходу а також особливості розробленого рішення.

#### ЛІТЕРАТУРА

1. Muro S. Multi-version concurrency control scheme for a database system / S. Muro, T. Minoura, T. Kameda. // Journal of Computer and System Sciences.
2. Navathe S. Fundamentals of Database Systems / S. Navathe, R. Elmasri., 2010. – 882 с.
3. Date C. SQL and Relational Theory, 2nd Edition / O'Reilly Media, 2011. – 172 с.
4. Parsian M. Data Algorithms. Recipes for Scaling Up with Hadoop and Spark / O'Reilly Media, 2015. – 693 с.
5. Shavit N. The Art of Multiprocessor Programming / N. Shavit, M. Herlihy., 2012. – 73с.
6. Harris T. A Pragmatic Implementation of Non-Blocking Linked-Lists [Електронний ресурс] / Timothy L. Harris // DISC '01 Proceedings of the 15th International Conference on Distributed Computing. – 2001.

Рецензент: к.т.н., ст. викл. КПІ ім. Ігоря Сікорського О.П. Сирота

# Реалізація та дослідження штучного нейрону на ПЛІС різних сімейств

Юрій Бердник  
КПІ ім. Ігоря Сікорського»  
Київ, Україна  
berdnyk@gmail.com

**Анотація.** В роботі наведені результати реалізації штучного нейрону з сигмоїдальною функцією активації на ПЛІС різних сімейств за розробленим методом реалізації нелінійних функцій активації та алгоритмами реалізації штучного нейрону. Отримані результати дослідження, а саме використаний ресурс ПЛІС та швидкодія представлені в таблиці.

**Ключові слова:** нейронні мережі, штучний нейрон, функції активації, ПЛІС, VHDL.

Однією з перших проблем при апаратній реалізації нейромережових систем управління є реалізація штучного нейрону і його активаційної функції засобами ПЛІС [1-3].

Активаційна функція нейрону виконує нелінійне перетворення сигналів, здійснюване нейроном. Найбільш поширеними видами активаційних функцій є гранична функція, лінійна функція, кусково-лінійна функція, функції сигмоїдального типу та функція Гауса. Найбільш широко використовуваним типом активаційних функцій є сигмоїдальні функції

Найскладнішою з погляду цифрової реалізації є реалізація функцій активації сигмоїдного типу, як суттєво нелінійних функцій. При розробці методів та алгоритмів реалізації нейронів з сигмоїдальною функцією активації слід мати на увазі особливості ПЛІС, які мають в якості ресурсів таблиці перетворення (ЛТ), мультиплексори, суматори, блоки множення, але не мають блоків ділення. Також слід враховувати кількість використаного ресурсу ПЛІС та оптимально його використовувати.

Існуючі підходи цифрової реалізації нелінійних функцій використовують різні методи апроксимації, такі як табличний метод, розкладання в ряд Тейлора, кусково-лінійна апроксимація і т.д. Розкладання в ряд Тейлора вимагає виконання багатьох множень, а тому неприйнятні для реалізації в ПЛІС, так як блок множення займає велику кількість ресурсу. Табличний метод передбачає створення глобальної змінної, таблиці можливих значень цільової функції, непередбачуваний і неконтрольований доступ до таблиць всіх нейронів мережі, що у свою чергу створює велику часову затримку, а створення окремої локальної таблиці для кожного нейрону мережі неприйнятно з точки зору використання ресурсу ПЛІС. Тому найбільш оптимальним методом реалізації функцій активації сигмоїдального типу є її кусково-лінійна апроксимація.

У роботі [4] розглянуто реалізацію одного нейрону на FPGA. Досліджено реалізацію 3-входового нейрону

з різними функціями активації на ПЛІС типу FPGA Xilinx XCV400hq240. Наведено таблиці порівняння швидкодії та займаного нейроном ресурсу для різних варіантів реалізації.

Запропонований нами метод та алгоритми апаратної реалізації нейронів і нейронних мереж з такими функціями активації детально описані в [5, 6].

Реалізований за даним алгоритмом штучний нейрон з 4-ма входами, 4-ма синапсами і вагами до них, з сигмоїдальною функцією активації, на ПЛІС з використанням 16-розрядних чисел з фіксованою точкою зайняв 493 LUTs (Look Up Table - вентиля логічної матриці). Кожне число – 16 біт, з яких 9 біт відведено на цілу частину і 7 біт на дробову. Швидкодія, як сумарна затримка комбінаційної схеми блоку нейрону, склала 59.85 нс. При конфігуруванні нейрону на чіпі сімейства Spartan 3 – XC3S200. Результати моделювання штучного нейрону на ПЛІС різних серій представлено в таблиці 1. При зменшенні або збільшенні кількості лінійних відрізків, кількості входів нейрону або зміні розрядності чисел зміниться і використаний ресурс ПЛІС для одного нейрону, а також точність і швидкодія. Штучний нейрон розроблений на ПЛІС, як окремий обчислювальний блок, та має зовнішні виводи для запису, потрібних синаптичних ваг. Абсолютна похибка обчислення на ПЛІС сигмоїдальної функції активації не перевищує  $\pm 0.005$ .

Таблиця 1  
Результати моделювання штучного нейрону на ПЛІС різних серій

ПЛІС		Швидкодія, нс	Ресурс ПЛІС (LUT)
Серія	Speed		
Spartan 3	-5	51.643	493
	-4	59.845	
Spartan 6	-2	38.964	315
	-3	33.213	
	-3N	34.465	
Zinq	-3	17.337	309
	-2	19.647	
	-1	23.493	
Virtex 7	-1	22.161	309
	-2L	18.529	
	-2	18.528	
	-2G	18.388	
Virtex 6	-2	20.288	309
	-1	23.477	
Virtex 5	-2	24.965	369
	-1	29.401	
Virtex 4	-10	37.714	485
	-11	32.903	
	-12	29.051	
Kintex 7	-3	16.360	309
	-2	18.528	

Artix 7	-1	22.161	309
	-2L	18.528	
	-1	28.131	
	-2	23.666	
	-2L	23.666	
	-3	20.629	

При дослідженні апаратної реалізації штучного нейрона на ПЛІС різних серій, розроблений компонент показав підвищення ефективності за такими показниками як кількість використаного ресурсу, швидкодія та похибка в порівнянні з найближчими аналогами. Побудовані на таких нейронах нейромережеві системи керування можуть бути використані при керуванні складними швидкодіючими об'єктами в реальному часі [7 - 9].

#### ВИСНОВКИ

В роботі проведено дослідження штучного нейрона при його апаратній реалізації на ПЛІС різних сімейств. Результати дослідження показали високу точність реалізації сигмоїдальної функції, швидкодію роботи штучного нейрона та невеликий використаний ресурс ПЛІС в порівнянні з аналогами. Це дозволить будувати штучні нейронні мережі з великою кількістю шарів та нейронів в них, та будувати на їх базі компоненти адаптивних систем управління реального часу швидкодіючими об'єктами.

#### ЛІТЕРАТУРА

1. G. Dreyfus. *Neural Networks: Methodology and Applications*. SpringerVerlag Berlin Heidelberg. 2005. P. 498.
2. M. Lawrynczuk. *Computationally Efficient Model Predictive Control Algorithms: A Neural Network Approach*. Springer. *Studies in Systems, Decision and Control*, 2013, vol. 3. P. 316.
3. Kravets P. I. *Methods of Hardware and Software Realization of Adaptive Neural Network PID Controller on FPGA–Chip* / P. I. Kravets, T. I. Lukina, V. A. Zhrebko, V. N. Shimkovich // *Journal of Automation and*

*Information Sciences*. Begell House, New York, USA. – 2011. – Volume 43, Issue: 4, pp. 70-77. DOI: 10.1615/JAutomatInfScien.v43.i4.80

4. A. Muthuramalingam. *Neural network implementation using FPGA: Issues and application* / A. Muthuramalingam, S. Himavathi, E. Srinivasan // *Int. J. Inf. Technol.* – 2008. – vol. 4, no. 2, pp. 86-92.

5. Kravets P.I. *Technology of Hardware and Software Implementation of Artificial Neuron and Artificial Neural Networks by Means of FPGA* / P.I. Kravets, V.M. Shymkovych, G.A. Zubenko // *Visnyk NTUU 'KPI' Informatics, operation and computer systems.* – 2012. – №55, pp. 174-180.

6. Kravets P.I. *Method and algorithms of implementation on PLIS the activation function for artificial neuron chains* / P.I. Kravets, V.N. Shimkovich, D.A. Ferens // *Elektronnoe modelirovanie.* – 2013. – Vol. 37, no. 4, pp. 63-74.

7. Kravets P.I., Shymkovych V.M., Omelchenko P. *Neuronetwork components of the systems of control of dynamic objects and their hardware-software implementation on FPGA*. *Visnyk NTUU 'KPI' Informatics, operation and computer systems.* – 2013. – №59, pp. 78-85.

8. Kravets P.I. *Neural controller stability of moving object with the hardware and software realization on FPGA* / P.I. Kravets, V.M. Shymkovych, V.V. Fedorchuk, A.A. Goy // *Visnyk NTUU 'KPI' Informatics, operation and computer systems.* – 2015. – №63, pp. 4–11.

9. P. I. Kravets, V. M. Shymkovych, and V. Samotyy. *Method and technology of synthesis of neural network models of object control with their hardware implementation on FPGA*, 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Bucharest, Romania, 2017, pp. 947-951.

*Рецензент: к.т.н., доц. КІП ім. Ігоря Сікорського П.І. Кравець*



# Дослідження впливу колірних моделей при визначенні кольорових характеристик об'єктів розпізнавання на зображенні

Антосев Андрій  
КПІ ім. Ігоря Сікорського  
Київ, Україна  
gibson12ant@gmail.com

Дорогий Ярослав  
КПІ ім. Ігоря Сікорського  
Київ, Україна  
Cisco.rna@gmail.com

**Анотація.** В статті здійснено аналіз методів визначення кольорових особливостей зображень. Розглянуто популярні колірні моделі та окреслено їхні особливості.

**Ключові слова:** колірна модель, обробка зображень, комп'ютерне бачення, RGB, HSV, Lab, розпізнавання

Сьогодні слідом за швидким ростом обчислювальних можливостей комп'ютерів стають все більш популярними задачі комп'ютерного бачення. За допомогою програмного аналізу зображення вирішується безліч побутових та професійних проблем. Однією з яких є розпізнавання та слідкування за рухомими об'єктами.

Під час виявлення окремих об'єктів на зображенні недостатньо інформації лише про їх форму та розташування, адже такі параметри в більшості випадків не несуть в собі унікальних знань про ці об'єкти. Найбільш важливою інформацією про об'єкт є його колір. Саме визначення кольорової характеристики зображення та подальший її аналіз суттєво спрощує задачу розпізнавання та слідкування за об'єктами.

Після виділення окремого об'єкта на зображенні, він представляється у вигляді набору пікселів із певними числовими значеннями кольорів. Слід виділити основні колірні моделі такі, як RGB, HSV та Lab.

RGB – найбільш розповсюджена адитивна колірна модель, що представляє собою спосіб синтезу кольору за допомогою накладання червоного, синього та зеленого світла відповідно [1].

Найпростішим методом для створення кольорової характеристики зображення є підрахунок кількості пікселів однакового кольору з використанням RGB моделі. Ідея полягає в тому, щоб порівнювати такі масиви даних з різних кадрів та слідкувати за об'єктами. Такий підхід є найлегшим як у розуміння так і з боку використання обчислювальних ресурсів. Проте існує ряд суттєвих недоліків. Так як кожна складова кольору має значення у межах від 0 до 255, то кількість можливих кольорів складає 16 777 216. При чому при найменшому впливі на об'єкт світла або зміни ракурсу до камери велика частина точок будуть змінювати своє кольорове значення. Для спрощення сприйняття кольору виконується нормалізація таких значень. Попередньо обирається в якому діапазоні цілих чисел повинні відобразитись кожен з каналів. Такий діапазон зазвичай є унікальним для різних ситуацій чи потреб. Часто обирають максимальне

значення - 7. Тобто кожен кольоровий канал, що попередньо був у діапазоні від 0 до 255, після нормалізації лежатиме в межах цілих чисел від 0 до 7 відповідно. Як нескладно зрозуміти, для нормалізації необхідно поділити кожне значення кольорового каналу на 32. Маючи спрощені значення кольорів, з ними на порядок легше працювати, адже невеликі зміни у їх відтінках у частині випадків не відображаються на їх цифровому значенні. Проте даний підхід є досить ненадійним. Нормалізовані кольори мають різку зміну тону, а також накладання тінні на частину об'єкта певного кольору може змінити його сприйняття.

Отже хотілось би звернути увагу на наступний метод, що вирішує недоліки свого попередника. Метод складання кольорової характеристики на основі HSV моделі враховує ситуації зі зміною тону кольору під час затінення об'єкта. Для початку декілька слів про дану кольорову модель. HSV відображає кольори у вигляді трьох наступних складових: колірний тон (Hue), насиченість (Saturation) та значення кольору (Value) [2]. Колірний тон - це значення "чистого" кольору, тобто без накладання чорного або білого тонів. Його значення зазвичай представляють в межах 0 – 360°, або просто від 0 до 100 (рис. 1).

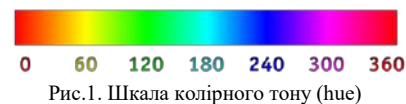


Рис. 1. Шкала колірного тону (hue)

Насиченість відповідає за відтінок білого у кольорі і лежить у межах від 0 до 100, або від 0 до 1. При показнику 1 колір "чистий", при 0 - білий. Цей параметр може змінюватись при засвіченні об'єкта білим світлом. Тобто при попаданні на об'єкт променів білого відтінку значення Hue, в ідеальному випадку, не змінюється. Змінює своє значення параметр Saturation. Отже можна зазначити, що обираючи колірну модель HSV, покращується сприйняття алгоритмом впливу світла на об'єкт розпізнавання.

Останнім параметром є - значення кольору. Дана складова відповідає за яскравість кольору та задається в аналогічних межах від 0 до 1, або від 0 до 100. При наближенні показника до 0 колір приймає чорний відтінок. Таку особливість колірної моделі дуже зручно використовувати для аналіз

падаючої на об'єкт розпізнавання тіні. Візуалізація HSV простору приведена на рис. 2 у вигляді циліндричної системи координат.

Така кольорова модель в більшості випадків дає кращі результати ніж RGB, проте вона не є ідеальною. Проблеми полягають у сприйнятті кольорів та світла. Іноді при засвіченні або затіненні об'єкта він приймає відтінок іншого кольору чим спотворює дані алгоритму обробки.

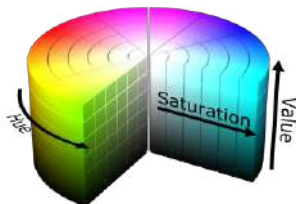


Рис. 2. Візуалізація HSV моделі

Наступною необхідно виділити кольорну модель Lab. Дана модель розроблялась з метою представлення кольорів у вигляді якомога ближчому до того, як їх сприймає людина. В просторі Lab відокремили хроматичну складову кольору від світлості. Тим самим Lab однозначно визначає колір, значення якого задається трьома параметрами. За світлість відповідає L, що лежить в діапазоні від 0 до 100, тобто від темного до світлого. Колір же визначається двома координатами - а та b. Перша координата визначає відношення зеленого до червоного, а друга відношення синього до жовтого. Так як в моделі Lab світлість відокремлюється від кольору дія світла на зображення майже не відображає змін у сприйнятті кольорів алгоритмом. А отже створення кольорової характеристики об'єкта виконується дуже подібно на різних кадрах під різним впливом на цих об'єктів світла, що є надзвичайно важливим при обробці послідовності кадрів із метою виділення окремих об'єктів.

Важливо відмітити, що перетворення RGB зображення в Lab - ресурсозатратний процес. Це може стати на заваді при реалізації алгоритму з використанням моделі Lab на слабких машинах, наприклад Raspberry Pi.

Для порівняння впливу вибору кольорної моделі на результат дії алгоритму розпізнавання було проведено наступне дослідження. Було обрано послідовність з десяти кадрів, попередньо записаних на відео з камери. На уривку відео зображена людина, що проходить по нерівномірно освітленому приміщенню, тобто на об'єкт спостереження на деяких кадрах падає світло або тінь. На кожному кадрі виділяється контур людини та знаходяться всі точки всередині контуру, кожна з яких має свій колір у відповідній кольорній моделі. Далі розраховується скільки пікселів якого кольору має цей об'єкт на кожному кадрі. Такий набір даних далі вважаємо за кольорну характеристику об'єкта. Після чого порівнюється кольорна характеристика на одному кадрі з попередньою, на скільки відсотків вони подібні. Слід зазначити, що відхилення до 10% можна вважати допустимим, адже розміри контуру можуть

змінюватись зі знаходженням об'єкта на зображенні. Також під час руху деякі частини об'єкта закриваються або навпаки з'являються.

Таблиця 1

		Порівняння кольорних моделей								
№ кадру		2	3	4	5	6	7	8	9	10
RGB		74 %	68 %	38 %	59 %	67 %	71 %	27 %	53 %	62 %
HSV		93 %	87 %	73 %	92 %	90 %	91 %	69 %	85 %	89 %
Lab		98 %	96 %	93 %	96 %	97 %	92 %	89 %	93 %	94 %

Також необхідно зауважити, що перед проведенням дослідження значення каналів були нормалізовані. Адже без нормалізації результати порівняння сусідніх кадрів склали близько 8% схожості при нормальних умовах, використовуючи кольорну модель RGB. Після нормалізації діапазони кожного параметру було зменшено у 8 разів, що не сильно спотворює дані.

На кадрах 1-3 на об'єкт падає світло; кадр 4 - вихід об'єкта у рівномірно освітлений простір; 5-7 об'єкт знаходиться у рівномірному освітленні. На кадрі 8 об'єкт заходить в тінь і рухається в темній частині приміщення.

З таблиці 1 можна побачити, що використовуючи модель RGB схожість кольорних характеристик об'єкта на сусідніх кадрах 60-70%. Проте при зміні освітлення результати різко погіршуються. Цю проблему вирішують моделі HSV та Lab. З їх використанням вплив світла та тіні не такий значущий. Найкраще себе показують саме Lab модель, через відокремлення значення світлості від кольору. Також слід зауважити, що використання двох останніх моделей при визначенні активності на відео значно покращує знаходження контурів рухомого об'єкта, адже поява тіні на зображенні ігнорується, що не можна сказати про модель RGB, де невелика зміна відтінку буде сприйматись, як рухомий об'єкт.

Таким чином при розробці алгоритму обробки зображень з урахуванням кольору рекомендовано обирати кольорну модель Lab, зважаючи на її переваги над іншими.

#### ЛІТЕРАТУРА

1. Лисиця В. Т. Колірні моделі та закони поширення світла / В. Т. Лисиця. – 2012. – С. 19.
2. Комп'ютерна графіка "Навчальний посібник". // Тернопільський національний технічний університет імені Івана Пулюя. – 2017. – С. 44.
3. Процедура коррекции цветового баланса цифровых изображений в цветовом пространстве lab. // Збірник наукових праць Харківського університету Повітряних Сил. – 2014. – №1. – С. 122–124

# Про вибір моделі надійного функціонування програмного забезпечення у хмарних технологіях

Савчук Олена  
КПІ ім. Ігоря Сікорського  
Україна, Київ  
savchuk\_11@ukr.net

Казанцева Анастасія  
КНУ ім. Тараса Шевченка  
Україна, Київ  
lesnaja@bigmir.net

Пирожков Олексій  
КПІ ім. Ігоря Сікорського  
Україна, Київ  
a.y.pirozhkov@gmail.com

**Анотація.** В даній доповіді розглядаються моделі надійного функціонування програмного забезпечення. Зроблений вибір динамічної моделі, що гарантує конфіденційність та безпеку інформації навіть при компрометації провайдера хмарних послуг або його інфраструктури.

**Ключові слова:** моделі, програмне забезпечення, хмарні технології.

## ВСТУП

У міжнародному стандарті ISO 9001:2008 виділено характеристики, які дозволяють оцінювати програмне забезпечення (ПЗ) із позиції користувача, розроблювача й керуючого проектом. Рекомендуються 6 основних характеристик якості ПЗ, кожна з яких деталізується декількома (усього 21) субхарактеристиками.

Функціональна придатність деталізується придатністю для застосування, точністю, захищеністю, здатністю до взаємодії й погодженістю зі стандартами й правилами проектування.

Надійність рекомендується характеризувати рівнем завершеності (відсутності помилок), стійкістю до помилок і перезапусканням.

Застосовність пропонується описувати зрозумілістю, навченістю й простотою використання.

Ефективність рекомендується характеризувати ресурсною й тимчасовою економічністю.

Супроводжуваність характеризується зручністю для аналізу, змінюваністю, стабільністю й тестовністю.

Переносимість пропонується відображувати адаптованістю, структурованістю, заміненістю й запровадженістю.

Характеристики й субхарактеристики в стандарті визначені дуже коротко, без коментарів і рекомендацій з їхнього застосування до конкретних систем і проектів.

Технологія хмарних обчислень забезпечує нову концепцію плати за використання корисних обчислювальних ресурсів, що базується переважно на технологіях віртуалізації. Основними перевагами хмарних обчислювальних послуг є: самообслуговування, широкий доступ до мережі, об'єднання ресурсів, швидке масштабування. Незважаючи на ці переваги, широке використання цієї нової технології зіштовхуються з низкою перешкод, включаючи безпеку та конфіденційність.

На додаток до традиційних ризиків безпеки ризик як у будь-якої обчислювальної системи, підключеної до Інтернету, хмарні системи мають особливі проблеми безпеки і конфіденційності через віртуалізацію і свою багаторівневу природу [1]. Тому саме надійність та діагностика займає перше місце серед характеристик якості у хмарних технологіях.

Надійна програма насамперед повинна забезпечувати досить низьку ймовірність відмови в процесі функціонування в реальному часі. Швидке реагування на перекручування програм, даних або обчислювального процесу й відновлення працездатності за час, менше, ніж поріг між збоєм і відмовою, забезпечують високу надійність програм. При цьому некоректна програма може функціонувати абсолютно надійно. У реальних умовах по різних причинах вихідні дані можуть попадати в області значень, що викликають збої, не перевірені при випробуваннях, а також не задані вимогами специфікації й технічного завдання. Якщо в цих ситуаціях відбувається досить швидке відновлення, таке, що не фіксується відмова, то такі події не впливають на основні показники надійності – напрацювання на відмову й коефіцієнт готовності. Отже, надійність функціонування програм є поняттям динамічним, що проявляється в часі, і істотно відрізняється від поняття коректності програм.

Основним принципом класифікації збоїв і відмов у програмах при відсутності їхнього фізичного руйнування є поділ за часовим показником тривалості відновлення після будь-якого перекручування програм, даних або обчислювального процесу, що реєструється як порушення працездатності.

## МОДЕЛЬ ФАКТОРІВ, ЩО ВИЗНАЧАЮТЬ НАДІЙНІСТЬ ПРОГРАМНИХ ЗАСОБІВ

Неможливо забезпечити абсолютну відсутність дефектів проектування в складних ПЗ, внаслідок чого надійність їхнього функціонування має завжди кінцеве, обмежене значення. При цьому передбачається, що відомо правильний, еталонний стан об'єкта, стосовно якого може бути визначена наявність відхилення – дефекту або помилки. Для систематичної, координованої боротьби з ними необхідно досліджувати фактори, що впливають на надійність ПЗ із боку випадкових, існуючих і потенційно можливих дефектів у конкретних

програмах, а у надзвичайних обставинах виконувати діагностичні тести системи.

Наступний крок – виправлення помилок самою системою – плідний метод проектування надійних систем апаратного забезпечення. Якщо деякий програмний модуль містить помилку, ідентичні «запасні» модулі також будуть містити ту ж помилку.

Розглянемо класифікацію моделей надійності ПЗ (МНПЗ), що підрозділяються на аналітичні й емпіричні. Аналітичні моделі дають можливість розрахувати кількісні показники надійності, ґрунтуючись на даних про проведення програми в процесі тестування (вимірюючи й оцінюючи моделі). Емпіричні моделі базуються на аналізі структурних особливостей програм.

Розглянемо основні передумови, обмеження й математичний апарат аналітичних моделей. Аналітичне моделювання надійності ПЗ включає чотири кроки: 1) визначення припущень, пов'язаних із процедурою тестування ПЗ; 2) розробка або вибір аналітичної моделі, що базується на припущеннях про процедуру тестування; 3) вибір параметрів моделей з використанням отриманих даних; 4) застосування моделі - розрахунок кількісних показників надійності по моделі.

#### ДИНАМІЧНІ МОДЕЛІ НАДІЙНОСТІ

*Модель Шумана* може бути використана при певним чином організованій процедурі тестування. Виявлені помилки реєструються (збирається статистика про помилки), але не виправляються. Передбачається, що до початку тестування в ПЗ є  $E_T$  помилок. Протягом часу тестування  $t$  виявляється  $\epsilon_c$  помилок розраховуючи на команду в машинній мові. Розраховується середній наробіток на відмову.

*Модель La Padula*. По цій моделі виконання послідовності тестів виробляється в  $m$  етапів. Перевага моделі полягає в тому, що вона є прогнозою й, ґрунтуючись на даних, отриманих у ході тестування, дає можливість пророчити ймовірність безвідмовної роботи програми в наступних етапах її виконання.

*Модель Джелінського - Моранди* відноситься до динамічних моделей безперервного часу. Основне положення, на якому базується модель, полягає в тому, що значення інтервалів часу тестування між

виявленням двох помилок має експонентний розподіл із частотою помилок (або інтенсивністю відмов), пропорційною числу ще не виявлених помилок.

*Модель Шіку - Волвертона*. Це модифікація попередньої моделі. В основі моделі Шіку - Волвертона лежить припущення, відповідно до якого частота помилок пропорційна не тільки кількості помилок у програмах, але й часу тестування, тобто ймовірність виявлення помилок із часом зростає.

*Модель Муса* відносять до динамічних моделей безперервного часу. Вважається, що протягом усього життєвого циклу ПЗ може відбутися  $M_0$  відмов і при цьому будуть виявлені всі  $N_0$  помилки, які були присутні в ПЗ до початку тестування.

*Модель перехідних ймовірностей*. Ця модель заснована на марковському процесі, що протікає в дискретній системі з безперервним часом. Процес, що протікає в системі, називається марковським (або процесом без наслідків), якщо для кожного моменту часу ймовірність будь-якого стану системи в майбутньому залежить тільки від стану системи в цей час ( $t_0$ ) і не залежить від того, яким чином система прийшла в цей стан. Процес тестування ПЗ розглядається як марковський процес. У початковий момент тестування ( $t=0$ ) у ПЗ було  $n$  помилок. Передбачається, що в процесі тестування виявляється по одній помилці. Тоді послідовність станів системи ( $n, n-1, n-2, n-3$  і т.д.) відповідає періодам часу, коли попередня помилка вже виправлена, а нова ще не виявлена. Помилки виявляються з інтенсивністю  $\lambda$ , а виправляються з інтенсивністю  $\mu$ .

#### ВИСНОВКИ

У разі злочинів та випадкових, ненавмисних перекручувань обчислювального процесу, програм і даних, модель перехідних ймовірностей забезпечує найкраще надійне функціонування ПЗ. Саме ця модель вибрана для подальшого дослідження у хмарних технологіях.

#### ЛІТЕРАТУРА

1. Jose M. Alcaraz Calero, Nigel Edwards, Johannes Kirschnick, Lawrence Wilcock & Mike Wray: Clouds: Toward a multi-tenancy authorization system for cloud services//IEEE Security and Privacy, 2010, pp. 16-122



**СИСТЕМИ КЕРУВАННЯ**

**CONTROL SYSTEMS**





# Алгоритм покоординатного спуску в методі динамічного програмування для задач цифрового оптимального керування

Майєр І.В.

КПШ ім. Ігоря Сікорського  
Київ, Україна

**Анотація.** Запропоновано модифікацію алгоритму розв'язання задачі оптимального керування дискретними об'єктами методом динамічного програмування шляхом застосування методу покоординатного спуску.

**Ключові слова:** динамічне програмування, оптимальне керування, метод рівномірного пошуку, метод покоординатного спуску, моделювання, MATLAB /Simulink.

Значне число задач, що виникають в суспільстві, пов'язані з процесами прийняття рішень. Параметри цих процесів вибираються таким чином, щоб забезпечити екстремальне значення певного показника (вартість, маса, час роботи, тощо) при умові наявних обмежень на параметри процесу. Сьогодні більшість систем керування проектують на основі різного роду цифрових обчислювальних пристроїв, що забезпечує високу точність, швидкодію, гнучкість при налаштуванні та збалансовану вартість. Розроблення та вдосконалення алгоритмів, що реалізують (зокрема) методи оптимального керування для побудови високошвидкісних цифрових систем автоматичного керування є актуальною задачею сьогодення.

При проектуванні і оптимізації системи важливим моментом є формування цілі оптимізації, яка математично виражається як вимога забезпечення екстремального значення деякого критерію оптимальності.

Динамічне програмування – це підхід до розв'язання складних задач шляхом їх розбиття на більш прості підзадачі. Даний метод заснований на принципі оптимальності Беллмана, який дозволяє скоротити перебір рішень в багатоетапних нелінійних задачах [1].

Метод динамічного програмування для задачі оптимального керування полягає у тому що потрібно з початкової заданої точки траєкторії об'єкту керування оптимально потрапити в задану кінцеву точку, частіше всього це 0. Для цього задачу ділять на підзадачі і починають вирішувати її з кінця – це називається зворотній прохід. Він полягає в тому, що на кожному кроці знаходять відповідно до правил обчислень усі можливі оптимальні рішення. Далі застосовується прямий прохід, який з усіх можливих знайдених рішень для кожної підзадачі дозволяє сформулювати оптимальне рішення основної задачі.

Відомий підхід до розв'язання задачі оптимального керування об'єктом першого порядку методом динамічного програмування [2] для знаходження на кожному проміжку часу мінімального

Писаренко А.В.

КПШ ім. Ігоря Сікорського  
Київ, Україна

рішення використовує метод рівномірного пошуку. Хоча даний метод і є легким у реалізації, але він не є ефективним, оскільки витрачається багато часу та обчислювальних потужностей для вирішення задачі. Для позбавлення від вказаних недоліків замість методу рівномірного пошуку для функції однієї змінної було обрано метод покоординатного спуску для функції багатьох змінних.

Застосуємо метод покоординатного спуску для розв'язання підзадач зворотного проходу методу динамічного програмування.

На першому кроці методу покоординатного спуску в якості початкового наближення обираємо одну із координат  $M_0(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})$  на заданій області допустимих значень. Підставляємо в критерій оптимальності  $I$  усі точки початкового наближення крім першої. Далі отримуємо функцію однієї змінної  $I = \int(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})$ . Знайшовши для даної функції точку мінімуму, переходимо від точки  $M_0$  до точки  $M_1(x_1^{(1)}, x_2^{(0)}, \dots, x_n^{(0)})$ , в якій критерій оптимальності  $I$  приймає мінімальне значення по координаті  $x_1$ . Це перший крок оптимізації, що складається з спуску по координаті  $x_1$ . Для знаходження мінімуму можна використовувати метод золотого перетину, метод Фібоначі, тощо. Далі потрібно підставити в критерій  $I$  всі координати точки  $M_1$  крім  $x_2$  та розглянути функцію виду  $I = \int(x_1^{(1)}, x_2^{(0)}, \dots, x_n^{(0)})$ . Знову вирішити одновимірну задачу оптимізації та знайти нову точку  $M_2(x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(0)})$ , в якій критерій  $I$  приймає мінімальне значення по координаті  $x_2$ . Аналогічно проводиться пошук за координатами  $x_3, x_4, \dots, x_n$ . Після цього все повторюється від  $x_1$  до  $x_n$ . В результаті буде отримана послідовність точок  $M_0, M_1, \dots, M_n$  в яких значення цільової функції складають монотонно спадну послідовність  $I(M_0) \geq I(M_1) \geq I(M_2) \geq \dots$ . Також на будь-якому  $i$ -му кроці можна даний процес призупинити та в якості мінімального значення використовувати значення критерію в точці  $M_i$  [3].

Таким чином, метод покоординатного спуску зводить задачу про знаходження найменшого

значення функції багатьох змінних до багаторазового рішення одновимірних задач оптимізації по кожній з цих змінних.

В середовищі MATLAB було написано скрипт для розв'язання задачі оптимального керування методом динамічного програмування з використанням методу по координатного спуску.

Розглянемо приклад. Задано об'єкт другого порядку у вигляді дискретної моделі у просторі станів:

$$\begin{cases} x_1[k+1] = 0,1765x_1[k] - 0,7686x_2[k] + 0,25u[k], \\ x_2[k+1] = 0,5x_1[k] + 0,5x_2[k], \\ y[k] = x_1[k] + x_2[k]. \end{cases} \quad (1)$$

Необхідно знайти таку оптимальну керуючу послідовність  $u$ , що переведе об'єкт керування з початкового стану  $X_{start} = [5; -5]$  в кінцевий стан  $X_{end} = [0; 0]$  за  $N = 13$  кроків квантування, а значення критерію якості  $I = \int (x_1^2 + x_2^2 + u^2)$  буде мінімальним. Область допустимих керувань:  $\Omega(u) : |u| \leq 1$ . Области допустимих станів:  $V(x_1) : x_1 \in [-8; 12]$ ,  $V(x_2) : x_2 \in [-9; 12]$ .

В результаті роботи скрипту оптиміано послідовність керувань та станів об'єкту в кожний момент квантування :

- k=0; x1=-5.000; x2=5.000; U=-0.0345
- k=1; x1=-4.733; x2=0.000; U=-0.0345
- k=2; x1=-0.843; x2=-2.366; U=-0.0345
- k=3; x1=1.662; x2=-1.605; U=-0.0345
- k=4; x1=1.518; x2=0.000; U=-0.0345
- k=5; x1=0.238; x2=0.773; U=-0.0345
- k=6; x1=-0.561; x2=0.505; U=0.0345
- k=7; x1=-0.478; x2=0.000; U=-0.0345
- k=8; x1=-0.072; x2=-0.253; U=-0.0345
- k=9; x1=0.175; x2=-0.162; U=0.0345
- k=10; x1=0.175; x2=0.000; U=-0.0345
- k=11; x1=0.016; x2=0.085; U=-0.0345
- k=12; x1=-0.071; x2=0.0505; U=0.0345
- k=13; x1=0.000; x2=0.000; U=-0.0345

На рис. 1 представлено побудовану в MATLAB/Simulink модель дискретної системи керування. Отриману послідовність керувань подаємо на об'єкт (Discrete State-Space) за допомогою блоку From Workspace. На рис. 2 представлено графіки перехідних процесів за станами  $x_1$  (суцільна лінія) та  $x_2$  (штрихова лінія). На рис. 3 зображено оптимальну траєкторію об'єкту керування, а на рис. 4 – графік керування системи.

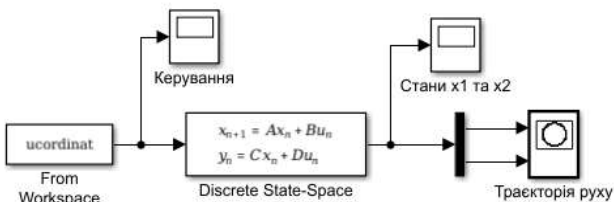


Рис. 1. Модель дискретної системи керування у MATLAB/Simulink

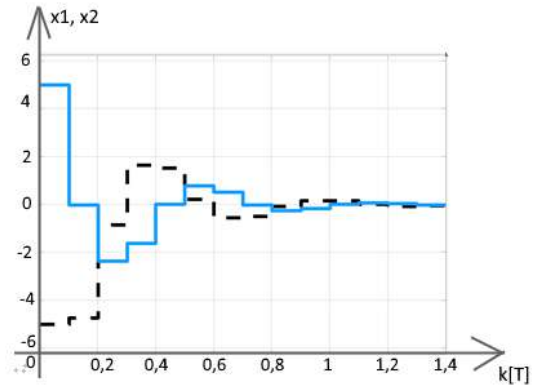


Рис. 2. Перехідні процеси за станами  $x_1$  та  $x_2$

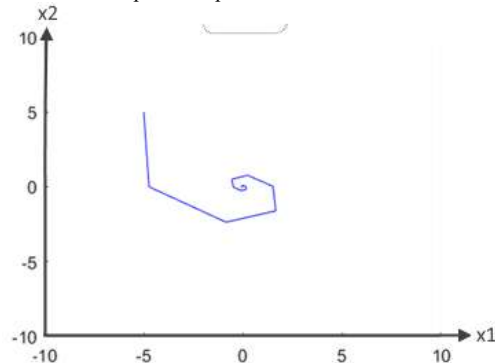


Рис. 3. Траєкторія руху об'єкту

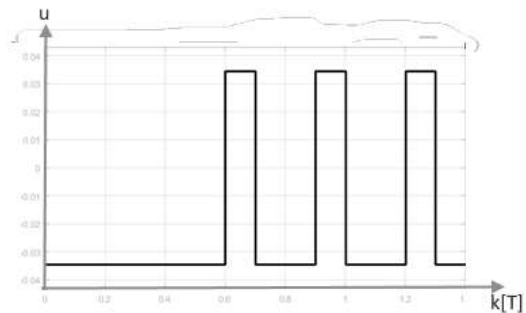


Рис. 4. Графік керування

З отриманих графіків видно, що знайдена оптимальна послідовність керування переводить його з заданого початкового стану у заданий кінцевий стан за визначену кількість тактів квантування.

ЛІТЕРАТУРА

1. Denardo E. V. Dynamic Programming: Models and Applications / Eric V. Denardo. – NY: Dover Publications, 2003.
2. Чураков, Е. П. Оптимальные и адаптивные системы : учебное пособие для вузов по специальности "Автоматика и телемеханика" / Е. П. Чураков . – М.: Энергоатомиздат, 1987 . – 256 с.
3. Акулич И. Л. Математическое программирование в примерах и задачах: Учеб. пособие для студентов эконом. спец. вузов. — М.: Высшая школа, 1986.

# Удосконалена формула розрахунку коефіцієнтів ПД-регуляторів для синтезу складних систем

Нагайко Дмитро  
КПІ ім. Ігоря Сікорського  
Київ, Україна  
nagayko.dima@gmail.com

Репнікова Наталія  
КПІ ім. Ігоря Сікорського  
Київ, Україна  
n.repnikova@gmail.com

**Анотація.** Розглянуті проблеми синтезу складних цифрових систем керування з ПД-регулятором. Пропонується удосконалена формула розрахунку коефіцієнтів ПД-регуляторів для цифрових систем керування, динаміка яких описується рівняннями високих порядків. Проведені експериментальні дослідження на моделях з використанням програмного пакета MATLAB/Simulink. Підтверджені адекватність та доцільність запропонованої формули.

**Ключові слова:** синтез, керованість, час перехідного процесу, ПД-регулятор.

## ВСТУП

Сьогодні інформаційні системи (ІС) займають одне з найважливіших місць в розвитку будь-якого промислового підприємства. Впровадження ІС на підприємстві та подальше її вдосконалення надає ряд значних переваг, таких як, автоматизація виробництва, підвищення конкурентоспроможності, підвищення якості та кількості продукції, зменшення витрат, збільшення прибутку тощо.

Одним з ключових етапів впровадження ІС на підприємстві є автоматизація процесів керування конкретними технічними об'єктами чи параметрами технологічного процесу. Тому, розроблення удосконалених алгоритмів цифрового керування різними об'єктами – задача актуальна та своєчасна.

Одним з найбільш поширених регуляторів в промисловості є ПД-регулятор [1]. Проте зі збільшенням порядку диференціальних рівнянь, що описують динаміку процесів керування, проблематичним стає отримання аналітичних залежностей для розрахунку коефіцієнтів ПД-регулятора. В літературі [1] запропоновано алгоритм розрахунку коефіцієнтів ПД-регулятора на базі білінійного перетворення. Проте в алгоритмі присутній недолік, який припускає перебір рішень.

В роботі пропонується вдосконалена формула розрахунку коефіцієнтів ПД-регуляторів для синтезу складних цифрових систем керування.

## ПОСТАНОВКА ТА ВИРІШЕННЯ ЗАДАЧІ

Розглянемо керований процес, що описується передавальною функцією n-го порядку виду:

$$W(s) = \frac{K_w}{s(T_1s + 1)(T_2s + 1) \dots (T_ns + 1)}, \quad (1)$$

де  $K_w$  – коефіцієнт підсилення передавальної функції керованого процесу.

Або якщо перейти до Z-перетворення, отримаємо передавальну функцію об'єкта керування у цифровому вигляді:

$$W(z) = \frac{Az^n + Bz^{n-1} + \dots + C}{(z-1)(z-z_1) \dots (z-z_n)} \quad (2)$$

Оскільки у виразі (2) присутня інтегрувальна ланка  $(z-1)$ , то для компенсації коренів  $z_1, z_2, \dots, z_n$  при умові збереження стійкості скомпенсованої системи буде використаний ПнД-регулятор, структурна схема якого зображена на рис. 1.

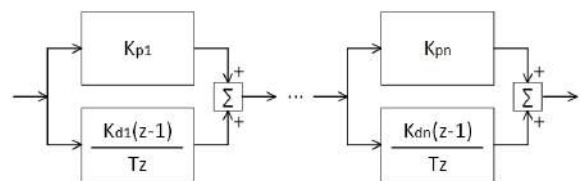


Рис. 1. Структурна схема ПнД-регулятора

Алгоритм визначення коефіцієнтів ПнД-регулятора:

1. Визначення граничного коефіцієнта підсилення передавальної функції керованого процесу  $K_b$ .

1.1 Визначення передавальної функції скоректованої системи  $W_c(z)$ :

$$W_c(z) = \frac{K_b A' z^n + K_b B' z^{n-1} + \dots + K_b C'}{T^n z^n (z-1)}, \quad (3)$$

$$\text{де } A' = \frac{A}{K_w}, B' = \frac{B}{K_w}, \dots, C' = \frac{C}{K_w}.$$

1.2 Визначення характеристичного поліному  $D(z)$ :

$$D(z) = z^{n+1} + az^n + bz^{n-1} + \dots + c \quad (4)$$

$$\text{З урахуванням білінійного перетворення } z = \frac{1 + \omega}{1 - \omega}$$

$$D(\omega) = a'\omega^{n+1} + b'\omega^n + c'\omega^{n-1} + \dots + d' \quad (5)$$

де коефіцієнти при степенях  $\omega$  ( $a', b', c', \dots, d'$ ) залежать від  $K_w$ .

1.3 Застосування критерію стійкості Гурвіца до (4) та розрахунок  $K_b$ .

2. Остаточна система рівнянь для розрахунку коефіцієнтів ПнД-регулятора матиме вигляд:

$$\left\{ \begin{aligned} &\left( K_{p1} + \frac{K_{D1}}{T} \right) \dots \left( K_{pn} + \frac{K_{Dn}}{T} \right) K_w = K_b \\ &\frac{K_{D1}}{K_{p1}T + K_{D1}} = z_1 \dots; \frac{K_{Dn}}{K_{pn}T + K_{Dn}} = z_n \\ &K_{p1} + \frac{K_{D1}}{T} = 1; \dots; K_{pn-1} + \frac{K_{Dn-1}}{T} = 1 \end{aligned} \right. \quad (5)$$

**ПРИКЛАД**

Нехай задана передавальна функція неперервної системи 3-го порядку виду:

$$W(s) = \frac{3,5}{0,25s^3 + 1,2s^2 + s}, T = 0,2c$$

Виконаємо перехід до Z-перетворення, та отримаємо передавальну функцію системи керування у цифровому вигляді:

$$W(z) = \frac{0,014784z^2 + 0,047124z + 0,009142}{(z-1)(z-0,806807)(z-0,47457)}$$

Розрахуємо граничний коефіцієнт  $K_b$  за формулами (3-5) та критерієм стійкості Гурвіца, отримаємо:

$$K_b = 2,5301$$

Розв'яжемо систему рівнянь (5), підставивши в неї  $K_b$ :

$$K_{p1} = 0.193193, K_{p2} = 0.37982584$$

$$K_{D1} = 0.1613614, K_{D2} = 0.06861197$$

Модель початкової цифрової системи керування та цифрової системи керування з ПнД-регулятором представлена на рис. 2. Графіки перехідних процесів зображені на рис. 3.

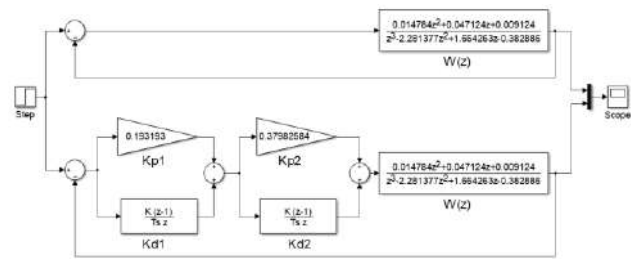


Рис. 2. Модель цифрової системи керування з ПнД-регулятором та без нього в програмному пакеті MATLAB/Simulink

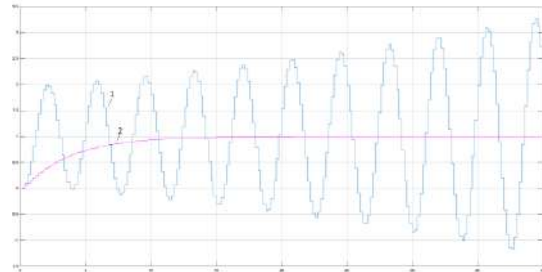


Рис. 3. Графіки перехідних процесів цифрової системи: 1 – без ПнД-регулятора, 2 – з ПнД-регулятором

Показники якості перехідного процесу системи з ПнД-регулятором наступні:  $t_{уст} = 23c, \delta = 0\%$ .

**ВИСНОВКИ**

Таким чином, запропонована удосконалена формула розрахунку коефіцієнтів ПД-регуляторів для синтезу складних цифрових систем керування. Це дозволить не тільки стабілізувати систему, але і покращить якість перехідного процесу (звести до нуля перерегулювання, зменшити час перехідного процесу).

**ЛІТЕРАТУРА**

1. Куо Б. Теория и проектирование цифровых систем управления : Пер. с англ. – М., Машиностроение, 1986. – С. 448.
2. Репнікова Н.Б. Синтез цифрових систем управління з ПнД – регуляторами / Н.Б. Репнікова, В.Ю. Федулова, Т.Я. Богодьорова. // Системи обробки інформації. – 2009. – №7(74). – С. 55-59

# Модель системи виявлення пішоходів та їх трекінгу з використанням рухомого джерела світла

Хлівненко Михайло  
КПІ ім. Ігоря Сікорського  
Київ, Україна

**Анотація.** На основі розробленого інструментарію для пакету MATLAB/Simulink запропоновано модель, що дозволяє виявляти пішоходів та відстежувати їх подальші пересування за допомогою фільтра Калмана. Розпізнавання пішоходів здійснюється за допомогою агрегованих функцій каналу (ACF). Отримані результати використовуються для побудови керуючого впливу на гібридний кроковий двигун, що позиціонує точкове джерело світла, яке освітлює рухомий об'єкт. Дана підсистема є частиною автомобільної системи нічного бачення.

**Ключові слова:** виявлення пішоходів, фільтр Калмана, відстеження рухомих об'єктів, безпека руху

## Вступ

Автомобільні системи нічного бачення призначені для забезпечення комфортного нічного водіння автомобіля, а також для навігації в умовах поганої видимості, що дозволяє підвищити безпеку водіння в нічний час.

З розвитком технологій все більше уваги приділяють безпеці пішоходів. Системи допомоги водіям забезпечують виявлення пішоходів у нічний час та знижують ризик дорожньо-транспортних пригод.

Автомобільні системи нічного бачення повинні використовувати швидкі та ефективні алгоритми пошуку пішоходів, а також надійні системи відслідковування подальшого руху знайдених об'єктів.

## ОПИС СИСТЕМИ

Запропонована система є частиною автомобільної системи нічного бачення. На рис. 1 зображено основні етапи, які виконує система.

Запропонована система показує, як здійснювати автоматичне виявлення та відстеження людей у відео з рухомої камери. Це демонструє гнучкість системи відстеження, адаптовану до рухомої камери, що ідеально підходить для застосування в автомобільній безпеці.

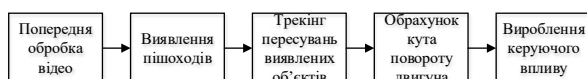


Рис. 1. Етапи роботи системи

Виявлення рухомих об'єктів відбувається за допомогою так званих агрегованих функцій каналу (aggregate channel features) [1]. Даний алгоритм

дозволяє точно та швидко виявляти людей на відео. Результатом функції детектування є координати центру, координати та розмір обмежувального прямокутника, а також ваговий коефіцієнт виявленої особи, що допомагає створювати певну пріоритетну чергу, адже система підсвічування пішоходів може одночасно підсвічувати лише один об'єкт [2]. За рахунок порівняння ваги виявлених об'єктів можливо обирати тих пішоходів, які становлять найбільшу загрозу для руху автомобіля.

Для збільшення роздільної здатності використовуються спеціальні кроки, що передбачають зміну розміру зображення. Приблизна висота пішохода обчислюється на основі даних про висоту ніг.

Після виявлення пішохода необхідно відстежити його пересування. Для цього використовується система стеження, що побудована на основі фільтра Калмана. Фільтр Калмана, відомий також як лінійно-квадратичне оцінювання — це алгоритм, що використовує послідовності вимірювання протягом часу, які містять шум (випадкові відхилення) та інші неточності, та видає оцінки невідомих змінних, що є потенційно точнішими за базовані на самих лише вимірюваннях. Формальніше, фільтр Калмана працює рекурсивно на потоках зашумлених вхідних даних, і видає статистично оптимальну оцінку базового стану системи [3].

Система стеження створює певний масив треків, де кожен трек є структурним представленням рухомого об'єкта на відео. Мета структури — це відслідковувати стан об'єкта. В свою чергу, стан об'єкта містить інформацію про час відслідковування, ваговий коефіцієнт об'єкта, а також передбачуване положення об'єкта у наступному кадрі [4].

Система виконує низьку функцій, серед яких: створення нових треків, видалення старих треків, оновлення існуючих треків, а також розрахунок положення рухомого об'єкта у кожному кадрі.

На рис. 2 представлено результат обробки кадрів.

Наступним кроком є розрахунок кута повороту крокового двигуна, що визначає напрямок променю світла. Знаючи кут огляду камери ми знаходимо ширину кадру, що припадає на один градус, а потім, знаючи ширину координати центру прямокутника, що обмежує пішохода, знаходимо кут повороту двигуна.

Знайдене значення подається для подальшої обробки в систему керування кроковим двигуном.



Використовується гібридний кроковий двигун з кутом повороту  $1,8^\circ$ . Використання мікрокрокового режиму роботи дозволяє знизити величину куту повороту до  $0,9^\circ$ . Таке значення дозволяє досягти високої точності у позиціонуванні джерела світла.

Модель системи позиціонування створена засобами MATLAB/Simulink і використовує модель управління кроковим двигуном [5]. Сигнал «step» управляє кроками двигуна, ширина імпульсу вказує на час роботи. Сигнал «dir» використовується для зміни напрямку обертання двигуна, значення «1» відповідає прямому напрямку обертання, а «0» - реверсному. На рис. 3 показано модель системи.



Рис. 2. Приклад роботи системи детектування

На рис. 4 представлено приклад графіку кута повороту крокового двигуна відповідно до розрахованого положення пішохода перед автомобілем.

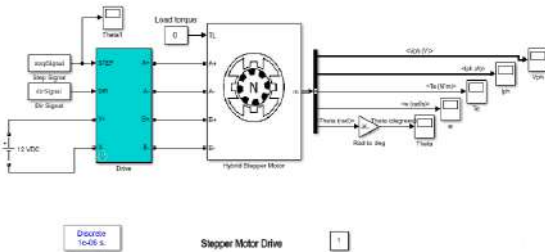


Рис. 3. Модель системи у MATLAB/Simulink

Кроковий двигун відповідає за позиціонування рухомого джерела світла, що являє собою напрямлений світловий потік, що підсвічує пішохода. Такий підхід є зручним та безпечним, адже водієві не потрібно дивитися в монітор для того, щоб побачити виявленого пішохода.

## ВИСНОВКИ

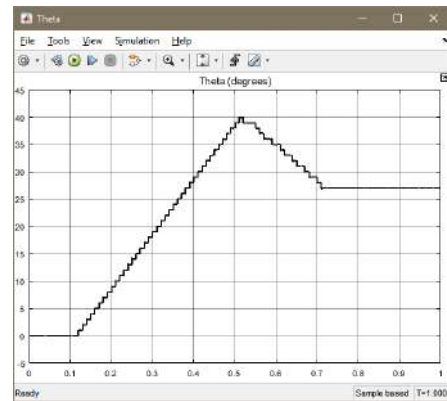


Рис. 4. Зміна кута повороту крокового двигуна у часі

Роблячи висновок, можна сказати, що запропонована модель виявлення пішоходів та відстеження їх пересувань з позиціонуванням точкового джерела світла є надійною та швидкою. Якість отриманих результатів свідчить про її високу точність.

Виявлення пішоходів відбувається з використанням швидкого алгоритму ACF. Завдяки додатковим крокам в алгоритмі відстежування переміщення об'єктів, система забезпечує безперервний та точний трекінг. На основі отриманих даних формується сигнал управління кроковим двигуном, що позиціонує рухоме джерело світла.

## ЛІТЕРАТУРА

1. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, 2005.
2. D P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. TPAMI, 2014.
3. [https://uk.wikipedia.org/wiki/Фільтр\\_Калмана](https://uk.wikipedia.org/wiki/Фільтр_Калмана) (Дата звернення 20.04.2018).
4. <https://www.mathworks.com/help/vision/examples/tracking-pedestrians-from-a-moving-car.html>. (Дата звернення 5.04.2018).
5. <https://www.mathworks.com/help/physmod/sps/examples/stepper-motor-drive.html> (Дата звернення 5.04.2018)

Рецензент: к.т.н., доц. КПІ ім. Ігоря Сікорського  
А.В. Писаренко

# Керування складськими роботами на базі алгоритму синтезу з бажаними коренями цифрової системи

Якимчук Владислав  
КПІ ім. Ігоря Сікорського  
Київ, Україна  
Vlad22109@gmail.com

Репнікова Наталія  
КПІ ім. Ігоря Сікорського  
Київ, Україна  
n.repnikova@gmail.com

**Анотація.** В роботі обґрунтовується можливість використання метода синтезу цифрових систем керування з заданим розташуванням полюсів для керування складськими роботами. Отримати практичні рішення для розв'язування задач синтезу для систем, динаміка яких описується рівняннями високих порядків.

**Ключові слова:** цифрові системи, полюси, керування.

## ВСТУП

Сучасні підприємства, чії склади спрямовані на конкурентоспроможність та ефективну роботу [1], повинні використовувати прогресивні методи організації управління – ІТ-технології, орієнтовані на чітке планування та координацію компонентів усіх торгово-технологічних процесів.

Автоматизація складу передбачає використання комплексів робото-технічних пристроїв, управління якими повинно виконуватися з заданою якістю. Тому розробка ефективних практичних методів синтезу цифрового керування складськими роботами задача актуальна.

## ПОСТАНОВКА ТА ВИРІШЕННЯ ЗАДАЧІ

Дослідити доцільність використання методу синтезу цифрового регулятора з заданим розташуванням полюсів для керування складськими роботами.

Для систем керування динаміка, яких описується у загальному вигляді моделлю передавальних функцій виду [2]:

$$W(s) = \frac{K}{a_0 s^n + a_1 s^{n-1} + \dots + a_n} \quad (1)$$

вирішити задачу синтезу цифрової системи.

Пропонується наступний алгоритм рішення задачі:

- 1) Перехід до моделі цифрової системи;
- 2) Перехід до моделі цифрової системи у просторі станів;
- 3) Визначення керованості;
- 4) Розрахунок коефіцієнтів зворотного зв'язку за допомогою формули (2) або бажаного характеристичного полінома за алгоритмом Check Step Response Characteristics;

$$\begin{aligned} N_n &= b_1 \prod_{i=2}^n a_{i(i-1)}, \\ V_n &= \sum_{i=1}^m (c_i N_i + \Delta_i (-1)^{i+1} - 1), \\ N_1 K_1 + N_2 K_2 + \dots + N_n K_n &= V_n, \end{aligned} \quad (2)$$

де  $a_{ij}, b_{ij}, c_{ij}$  - елементи матриці А, В та С.

- 5) Визначення матриці зворотнього зв'язку.
- 6) Моделювання системи керування з використанням Matlab\Simulink.

## ПРИКЛАД

Нехай задана динаміка роботи складського робота, що описується моделлю в передавальній функції:

$$W(s) = \frac{2}{0.06s^3 + 0.5s^2 + s}$$

- 1) Цифрова модель:  
>>W2=c2d(W,T)
- 2) Векторно-матрична модель цифрової системи:

```
>>ss(W2)
>>A=ans.a
```

$$A = \begin{bmatrix} 2.3231 & -0.8788 & 0.4346 \\ 2 & 0 & 0 \\ 0 & 0.5 & 0 \end{bmatrix}$$

```
>>B=ans.b
```

$$B = \begin{bmatrix} 0.125 \\ 0 \\ 0 \end{bmatrix}$$

```
>>C=ans.c
```

$$C = [0.0363 \quad 0.0592 \quad 0.0239]$$

- 3) Керованість:  
>>P=ctrb(A,B)  
>>rank(P)=3  
отже система керована.
- 4) Згідно формули (2) визначено необхідні коефіцієнти:

$$N_1 = 0.125,$$

$$N_2 = 0.25,$$

$$N_3 = 0.125,$$

$$V_1 = 1.328,$$

$$V_2 = -0.4153,$$

$$V_3 = 0.0223.$$

5) Матриця зворотнього зв'язку:

$$K = (10.6208 \quad -6.9715 \quad 3.5007).$$

6) Модель системи та перехідний процес представлено на рис. 1, 2 відповідно.

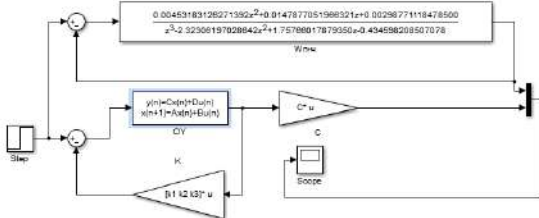


Рис. 1. Модель синтезованої та вихідної моделі

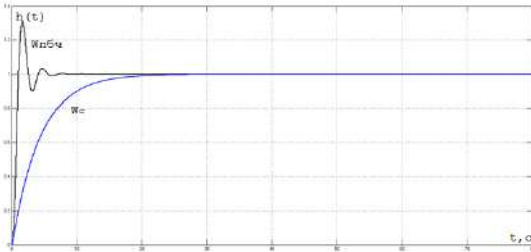


Рис. 2. Перехідний процес вихідної та синтезованої системи

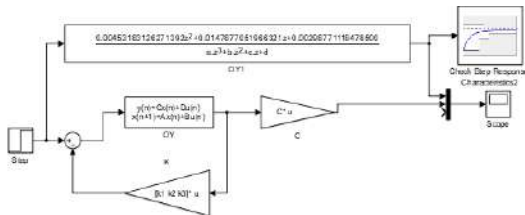


Рис. 3. Модель з Check Step Response Characteristics

За алгоритмом Check Step Response Characteristics (рис. 3) визначено бажаний характеристичний поліном.

$$z = (1.8528 \quad -0.9106 \quad -1.7872 \quad 0.8671).$$

Виконаємо порівняння результатів роботи кожного методу.

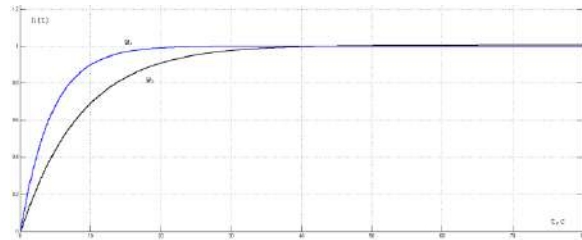


Рис. 4 – Перехідний процес з Check Step Response Characteristics(W2) та керування з заданим розташуванням полюсів(W1)

З отриманих результатів видно, що метод синтезу з використанням керування з заданим розташуванням полюсів(W1-верхня характеристика) забезпечує час регулювання в  $t_{nm} = 29,2c$  порівняно з методом синтезу за допомогою блоку Check Step Response Characteristics(W2), який надає  $t_{nm} = 59,7c$ .

ВИСНОВКИ

Таким чином доведено практичну доцільність використання при синтезу цифрових систем керування роботами аналітичної формули визначення бажаних коренів.

ЛІТЕРАТУРА

1. Эксклюзивные статьи про роботов и робототехнику [Електронний ресурс] – Режим доступу до ресурсу: <http://robotics.ua/shows/>.
2. Репнікова Н. Б. Теорія автоматичного управління: класика і сучасність/ Н. Б. Репнікова,– Київ: НТУУ (КПІ), 2012. – 238 с.

**ТЕХНОЛОГІЇ  
ПРОГРАМУВАННЯ**

**PROGRAMMING  
TECHNOLOGIES**



# UML діаграми для програмного комплексу дистанційного управління рухомими об'єктами

Кравінський Василь  
КПІ ім. Ігоря Сікорського  
Київ, Україна  
vasyl.kravinskyi@gmail.com

**Анотація.** Запропоновано варіант організації UML діаграм програмного забезпечення для програмних комплексів віддаленого управління рухомими об'єктами. Запропоновано варіант об'єднання різних типів діаграм.

**Ключові слова:** програмні комплекси для дистанційного управління, дистанційне управління, UML.

## ВСТУП

На сьогоднішній день все більшої популярності набувають програмні комплекси для дистанційного управління (ПКДУ) рухомими об'єктами (РО). Для розробки, налагодження і супроводження ПКДУ рухомими об'єктами необхідно створювати UML діаграми програмного забезпечення (ПЗ). Формалізація ускладнюється з двох основних причин. По-перше, через використання різних технологій ПЗ, що застосовуються в ПКДУ. По-друге, у ПКДУ поєднуються високотехнологічні елементи ПЗ пульта управління з графічними інтерфейсами і ПЗ виконавчих механізмів рухомих об'єктів. Отже, з'являється потреба формалізації декількох парадигм (ООП, процедурний підхід, функціональна парадигма) створення ПЗ у одній програмній системі.

Таким чином і перша і друга причина вимагають від дослідників формалізації програмних елементів ПКДУ у вигляді, що не залежить від технології ПЗ і парадигми його виконання.

У даній статті пропонується рішення з використанням загальновідомих засобів UML [1].

## АНАЛІЗ ВІДОМИХ РІШЕНЬ, ЗАГАЛЬНА ПОСТАНОВКА ЗАВДАННЯ

Відомі рішення формалізації аналогічних систем через UML [2-7]. Більша частина авторів даних джерел використовує UML як допоміжний елемент, після розробки вихідного коду для маркетингової мети. В цих джерелах відсутня узагальнена модель, що може стати прототипом ПКДУ, присутні виключно конкретні рішення.

Завдання формалізації ПКДУ також ускладнюються багатопоточністю. Програмне забезпечення ПКДУ не завжди зручно формалізувати у вигляді класичних мереж Петрі, через ускладнення моделі і не дуже широким її розповсюдженням у середовищі розробників ПЗ [8].

Таким чином потрібні дослідження для виявлення загальних підходів до формалізації ПКДУ у вигляді UML, що призначені для розробки, налагодження і вдосконалення ПЗ.

## ФОРМАЛЬНИЙ ОПИС ДІАГРАМИ ПРЕЦЕДЕНТІВ ПКДУ

На першому етапі опишемо формальну модель у вигляді UML діаграми прецедентів. При її створенні будемо дотримуватися стандартів UML [1].

Особливість побудованої діаграми полягає у тому, що вона є спробою узагальненої формалізації ПКДУ рухомих об'єктів.

Діаграма прецедентів ПК сайту, графічного інтерфейсу описується за допомогою функціональної моделі, що відображає системні прецеденти, системне оточення (діючі особи або актори) і зв'язки між прецедентами й акторами (діаграми прецедентів), що показана на рис. 1.

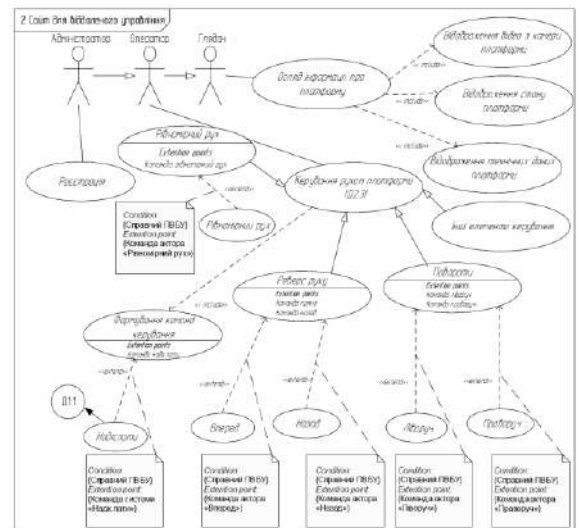


Рис 1. Діаграма прецедентів ПДУ

Основними дійовими особами (акторами) даної діаграми є «Адміністратор», «Оператор» та «Глядач». Актор «Глядач» має права тільки на огляд інформації про роботу і огляд загальної інформації. «Оператор» має усі можливості глядача, а крім того, він може керувати рухом платформи. «Адміністратор» має усі можливості як глядача так і оператора, а також він може реєструвати нових користувачів.

Діаграма прецедентів РО показана на рис. 2. Ці діаграми виконані у рамках стандарту і докладних пояснень не потребують за винятком особливостей ПКДУ рухомими об'єктами. Цією особливістю є взаємодія РО і користувачів – керівників РО через ПДУ. Опишемо процес взаємодії більш докладно.

Далі розглянемо діаграму прецедентів «Об'єкта управління» (рис. 2).



Програмний компонент РУ у загальній системі управління і ПКУ може бути реалізований у різні способи. Тому необхідно розкрити взаємозв'язок з іншими компонентами системи використовуючи UML діаграми.

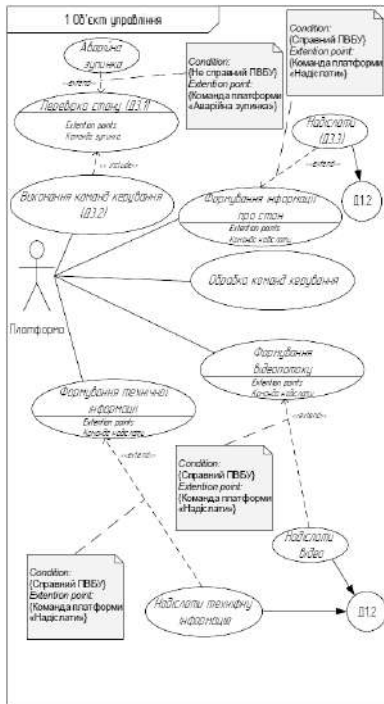


Рис 2. Діаграма прецедентів «Об'єкту управління»

Основною дійовою особою, що бере участь у даній діаграмі виступає «Платформа». У загальній системі ПКУ всі ПК працюють у єдиній системі. На рис.2, рис. 3 вони об'єднані, шляхом використання посилань, вигляду «Д\*.». Наприклад, варіант використання «Надіслати» (рис. 2) вказує на «Д1.1», що означає, що інформація надсилається до «Об'єкта управління».

#### ФОРМАЛЬНИЙ ОПИС ДІАГРАМИ СТАНІВ

При описі станів створеного прототипу, розроблено діаграми станів (рис. 3). Для кращого розуміння, розглянемо їх більш докладно.

Діаграма станів РО (рис. 3) містить 2 основні стани, а саме «Невизначений стан» та «Працездатний стан». При запуску платформи, вона перебуває у невизначеному стані. Якщо перевірка пройшла успішно, РО переходить у працездатний стан. У ньому виконуються усі команди управління, які надходять з ПДУ.

Покажемо зв'язок діаграми прецедентів і діаграми станів даних об'єктів. Наприклад варіант використання «Перевірка стану (Д3.1)» вказує на зв'язок зі станом «Перевірка стану платформи» діаграми станів (рис. 3).

#### ВИСНОВКИ

Результатом даної роботи є формальна модель, що побудована у стандартах UML [1]. Проведені дослідження показали можливість узагальнення підходів і побудови узагальнених моделей для ПКУД

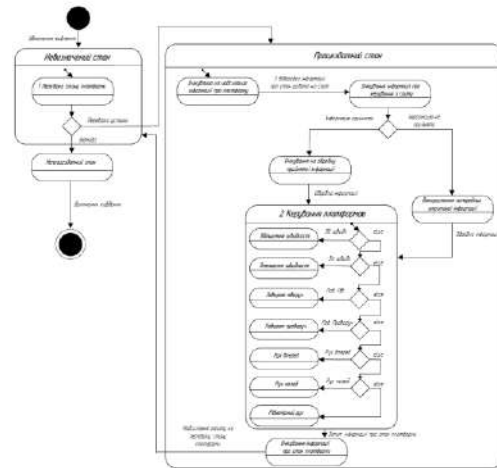


Рис 3. Діаграма станів РО

рухомих об'єктів у вигляді UML, що призначені для розробки, налагодження і вдосконалення ПЗ.

#### ЛІТЕРАТУРА

1. OMG Unified Modeling Language [Електронний ресурс] – режим доступу до ресурсу: <https://www.omg.org/spec/UML/2.5.1> (дата звернення 31.05.2018). – Назва з екрану.
2. Nilesh J. Uke, Ravindra C. Thool. Objects tracking in video: a object-oriented approach using Unified Modeling Language. International Journal of Computational Vision and Robotics (IJCVR), Vol. 5, No. 2, 2015.
3. Infantino, I., Cossentino, M. and Chella, A. An agent based multilevel architecture for robotics vision systems', Proceedings of the International Conference on Artificial Intelligence, IC-AI Las Vegas, pp.386–390,2002.
4. Shi-Xiang, T. and Wang, S. The conceptual design and simulation of mechatronic system base on UML, 2010 2nd International Conference on Computer Engineering and Technology, pp.V6-188–V6-192,2010.
5. Gavrilescu, M. Towards UML software models for cyber physical system applications, 2012 20th Telecommunications Forum (TELFOR), No. 2, pp.1701–1704,2012.
6. Evans, A.S. and Wellings, A.J. UML and the formal development of safety-critical real-time systems, Applicable Modelling, Verification and Analysis Techniques for Real-Time Systems, IEE Colloquium on, Vol. 2, No. 1, pp.2–5,1999.
7. Yuan, L., Tang, T. and Liu, Y. Research on third-party test bench of train control system using UML, 2011 IEEE International Conference on Service Operations, Logistics, and Informatics (SOLI), pp.560–564,2011.
8. Катін П. Ю. , Досенко Є. В., Іванів Р. Б. Моделювання програм з паралельними обчисленнями за допомогою мереж Петрі / П. Ю. Катін, Є. В. Досенко, Р. Б. Іванів // Економіка і управління. – 2013. – №4(60). – С.124-132

Рецензент: к.т.н., доц. КНУ ім. Ігоря Сікорського П.Ю. Катін

**БЕЗПЕКА ТА ЗАХИСТ  
ІНФОРМАЦІЇ**

**INFORMATION SECURITY**



## Розподіл ключів

Радецький Тимофій  
КПІ ім. Ігоря Сікорського  
Київ, Україна

**Анотація.** Управління ключами являє собою найважчу частину криптографії. Організація накопичення ключів це їхнє зберігання, облік та видалення. Для обміну ключами, як правило, використовують алгоритм RSA або більш ефективний алгоритм Діффі – Хелмана. Перед сучасною криптографією стоїть проблема підвищення стійкості алгоритмів і зменшення розмірів ключів та блоків даних. Очевидний спосіб рішення цієї проблеми – представлення блоків інформації в криптографічних алгоритмах точками еліптичних кривих.

**Ключові слова:** криптографічні системи, протоколи розподілу ключів, циклічні групи, еліптичні криві, RSA, ECC.

Забезпечення інформаційної безпеки є одним з найголовніших напрямів розвитку інформаційних технологій. Набір завдань, які розглядаються в цій галузі, постійно збільшується. Одним з основних засобів, що використовуються для захисту інформації в комп'ютерних системах, є криптографічні перетворення. Проблеми криптографії виходять за рамки задач забезпечення секретності. Сучасна криптографія включає в себе чотири основних розділи: криптосистеми з відкритим ключем, симетричні криптосистеми, системи електронного підпису, управління ключами. Сучасні криптосистеми побудовані на користуванні ключами і, як правило, управління ключами є слабким місцем у криптографічних додатках. Користування криптографічними технологіями не є складним, однак складним є безпечно користуватися ключами, зберігати та обмінюватися ними. Часто неякісне управління ключами підриває навіть виключно якісні системи, так як безпека алгоритму зосереджена в ключі. Управління ключами включає генерацію, накопичення і розподіл ключів.

У сучасному світі управління ключами є найважчим розділом криптографії. Проектувати безпечні криптографічні алгоритми і протоколи непросто, але наявний значний обсяг академічних досліджень. Зберегти секрет ключів набагато важче [1].

Загалом, усі методи генерації ключів можна розділити на апаратні та програмні. Основною вимогою при цьому є рівномірність розподілу по всьому простору можливих ключів. При генерації ключів апаратним способом використовуються генератори шуму – електронні пристрої, в яких протікає випадковий фізичний процес; при програмній генерації – генератори псевдовипадкових послідовностей [2].

Накопичення ключів пов'язане з їхнім зберіганням, обліком та видаленням. У складних інформаційних

системах один користувач може працювати з великим обсягом ключової інформації, внаслідок чого виникає необхідність організації міні-баз даних з ключовою інформацією. Такі бази даних відповідають за прийняття, зберігання, облік та видалення ключів, які використовуються. Інформація про ключі повинна зберігатися зашифрованою. Ключі, які шифрують ключову інформацію називаються майстер-ключами і, як правило, не зберігаються в комп'ютерній системі, або для їхнього перетворення використовують криптографічні алгоритми. Кількість ключів, що використовується, залежить від кількості абонентів, обсягів інформації і особливостей алгоритму шифрування. Сеансові ключі повинні знищуватися.

Питання оновлення ключів пов'язане з третім аспектом управління ключами – розподілом ключів. Розподіл ключів – одна з фундаментальних задач криптографії. При обслуговуванні  $n$  користувачів, що обмінюються закритою інформацією один з одним, необхідно  $n(n-1)/2$  різних секретних ключів. Зі збільшенням  $n$  з'являється проблема управління великим числом ключів. Існує кілька шляхів розв'язання цієї проблеми. Визначення найбільш оптимального з них залежить від конкретної ситуації [1]:

- Фізичний розподіл. За допомогою озброєної охорони або довірених кур'єрів ключі можуть розсилатися традиційним фізичним шляхом. Дана процедура використовується як в симетричних, так і в асиметричних криптосистемах. Передбачається, що творець ключів буде передавати асиметричний секретний ключ (або асиметричний відкритий ключ) фізично безпечним способом.

- Видача загального ключа учасникам взаємодії центром видачі ключів – схема «абонентського шифрування». В такій системі центр виготовлення і доставки ключів виступає як гарант справжності і автентичності повідомлень, так як він не тільки постачає користувачів ключами, але і несе відповідальність за їхню секретність при виготовленні та доставці. Якщо центр скомпрометований, то забезпечення безпеки наступних запитів на видачу ключів є проблемним, а безпека раніше виданих ключів залежить від криптосистеми.

- Забезпечення центром сертифікації ключів доступу до відкритих ключів користувачам і видача секретних ключів користувачу. Може використовуватися як симетричними, так і асиметричними криптосистемами. Так як при даному способі кожен користувач повинен якимось чином безпечно взаємодіяти з центром видачі ключів з самого початку роботи, то в цьому випадку

початковий обмін ключами є проблемним. Якщо довгострокові секретні ключі розподілені між користувачами і певним центром, який зазвичай називають центром розподілу ключів, то використовують спеціальні криптографічні протоколи. Цей спосіб розподілу передбачає, що користувачі та центр працюють в режимі онлайн.

- Мережа довіри. Використовується в асиметричних криптосистемах. Користувачі самі розповсюджують свої ключі і слідкують за ключами інших користувачів. Довіра полягає в неформальному способі обміну ключами. Одне з рішень є в тому, що за кожним користувачем закріплюється єдиний ключ, використовуючи який він може зв'язуватися з центром довіри. В цьому випадку система з  $n$  користувачами потребує лише  $n$  ключів. Коли два користувачі хочуть обмінятися секретними даними, вони генерують ключ, який буде використовуватися лише для передачі цього повідомлення. Його називають ключем сеансу. Ключ сеансу генерується за участі центру довіри за допомогою одного з протоколів.

- Протоколи обміну ключами. Видача і обмін секретним ключем відбувається через незахищені канали зв'язку між учасниками взаємодії, які до цього не мали спільного секретного ключа. Використовуючи криптосистему з відкритим ключем, партнери, які не довіряють посередникам і не мають можливості зустрітися, можуть домовитись про спільний секретний ключ в режимі онлайн відповідно до протоколу про обмін ключами. Спочатку сторони завчасно узгоджують секретний ключ. Потім для шифрування фактичної інформації застосовується симетричний шифр з узгодженим ключем.

Недолік методів, які використовують центр розподілу ключів, полягає в тому, що центру відомо, кому і які ключі призначені, що дозволяє читати всі повідомлення в інформаційній системі. При прямому обміні ключами виникає проблема автентифікації справжності суб'єктів. Вирішити її можна двома шляхами:

1) Механізм «запит – відповідь». Якщо користувач А бажає бути впевненим, що повідомлення, отримані від Б є істинними, він додає до повідомлення непередбачений елемент (запит). У відповіді абонент Б повинен виконати певну операцію над цим елементом. Окрім того, це неможливо зробити раніше, оскільки невідомо, яке випадкове число прийде в запиті. Після отримання відповіді з результатами дій користувач А може бути впевненим, що сеанс є справжнім. Недоліком цього методу є можливість встановлення – нехай складної – залежності між запитом і відповіддю.

2) Механізм позначки часу. Він передбачає фіксацію часу для кожного повідомлення. В цьому випадку кожен користувач інформаційної мережі може знати, наскільки «старим» є повідомлення. При використанні позначок часу постає проблема допустимого часового інтервалу затримки для підтвердження справжності сеансу, так як повідомлення з часовою позначкою не може бути переданим миттєво. Крім того, комп'ютерні

годинники відправника і отримувач повинні бути абсолютно синхронізованими.

Більшість відомих протоколів розподілу ключів використовуються для отримання загального ключа сеансу для двох учасників. Для обміну ключами, як правило, використовують алгоритм RSA або більш ефективний алгоритм Діффі – Хелмана, який дозволяє двом користувачам без посередників обмінюватись ключем, який потім може використовуватись для симетричного шифрування [3].

Сьогодні більш актуальною задачею є розподіл ключів для групи користувачів. Така задача виникає при автентифікації учасників групи, при організації безпечного зв'язку всередині групи абонентів, при формуванні групового цифрового підпису, при організації конференц-зв'язку. Криптографічні протоколи, в яких відбуваються добування і розповсюдження ключів, називаються протоколами розподілу ключів. Проблема розподілу ключів може бути вирішена в кілька шляхів. Спільний ключ може не добуватися у протоколі, а обиратися завчасно кимось з учасників і потім відправлятися усім іншим учасникам. Такі протоколи називаються протоколами розповсюдження ключів. Очевидно, в такому випадку рівень захисту невисокий. Для підвищення рівня безпеки можуть використовуватись протоколи обміну ключами, в яких кожен учасник групи робить свій вклад при генерації ключа. В обох випадках тільки діючі учасники групи мають доступ до секретного ключа.

Донедавна основним математичним апаратом, який використовували для реалізації протоколів обміну ключами були циклічні групи простого порядку  $p$ . Задача, що лежить в основі цих протоколів – проблема дискретного логарифмування (розрахунок показника  $x$  по відомим  $g^x \bmod p$  і твірному елементу  $g$ ). Однак перед сучасною криптографією стоїть проблема підвищення стійкості алгоритмів і зменшення розмірів ключів та блоків даних. Очевидний спосіб рішення цієї проблеми – представлення блоків інформації в криптографічних алгоритмах не тільки у вигляді чисел (елементів скінченних піль), але і у формі інших алгебраїчних об'єктів вищої складності. Одним з типів таких об'єктів є точки еліптичних кривих. Серед основних переваг, які дає еліптична крива – висока швидкодія і невелика довжина ключа.

Таблиця 1

Порівняння розмірів ключів RSA та ECC [2]

Розмір ключа RSA (біти)	Розмір ключа ECC (біти)
1024	160
2048	224
3072	256
7680	384
15360	521

За деякими оцінками експертів, криптосистеми на основі еліптичних кривих дають найвищу надійність на 1 біт ключа із усіх відомих систем з відкритим ключем. Це досягається завдяки важкості базового алгоритму – розрахунку дискретних логарифмів у групі точок еліптичної кривої (розрахунок множника  $k$  з відомими  $kG$  і твірної точки  $G$ ). Менша довжина

ключа призводить до пришвидшення розрахунків, зниженню потрібної потужності, меншому завантаженню пам'яті. Таким чином, еліптичні криві є досить привабливими для побудови більш стійких криптографічних алгоритмів.

В залежності від потрібної стійкості протоколи обміну ключами повинні володіти рядом криптографічних властивостей [3]:

- Контрибутивність: сформований секретний ключ залежить від секретних даних, внесених кожним учасником протоколу;

- Доскональна випереджальна секретність: компрометація довгострокових ключів не компроментує ключів сеансу;

- Неявна автентифікація ключа: кожен учасник протоколу впевнений, що жодна інша сторона не могла отримати доступ до сеансового ключа ( за винятком зловмисника);

- Автентичність: забезпечення неявної автентифікації ключа.

Можливі проблеми і труднощі в користуванні системами на основі еліптичних кривих:

- 1) Реальна стійкість таких систем до кінця не доведена. Дослідження показують, що частина еліптичних кривих, які використовуються в протоколах шифрування та розподілу ключів фактично для цього не підходять.

- 2) Важкість генерації кривих, які задовольняють умовам. Головна проблема – визначити кількість точок на кривій. Для цього необхідно обрати базову точку  $P$ , координати якої повинні мати досить великі значення для забезпечення стійкості. Окрім того, координати  $P$  повинні ділитись на кількість точок на кривій і скоріше за все, при визначенні числа точок на кривій, знаходження базової точки буде проблемним.

- 3) Використання ключів малих розмірів. Як не дивно, в цьому теж є проблема. З одного боку, потрібні менші потужності та час для підбору ключа. З іншого боку, менша довжина ключа знижує вимоги до потужності обчислювальних пристроїв, що впливає на швидкість обробки (наприклад, при створенні і перевірці електронного підпису)

Результати тестування Strytography Research Inc. показали, що електромагнітні хвилі, які випромінюються телефонами, можуть стати причиною втрати зашифрованої інформації: достатньо виміряти і проаналізувати кількість енергії, яку споживає пристрій або частоту його радіовипромінювання. Коли обробка додатку містить секретні елементи – такі як секретний ключ, який повинен залишатись в межах пристрою – конфіденційна інформація може стати незахищеною [4].

Simple Power Analysis (SPA) є одиничним виміром, який вимірює випромінювання пристрою або його енергоспоживання. Зокрема, випромінюванні пристроєм великі коливання можна оброблювати для виявлення індивідуальних бітів ключа.

Differential Power Analysis (DPA) є збором серії вимірів. Такий статистичний аналіз використовується для отримання ключових даних, отриманих шляхом надточних вимірів перепадів в енергоспоживанні пристрою. В результаті тестування SPA- та DPA-методів було встановлено, що для викрадення інформації мобільний пристрій достатньо наблизити до спец-обладнання, на відстань, меншу 3 метрів. Дослідження привели до висновку, що будь-які додатки, які не вживають активних контрзаходів захисту, знаходяться під загрозою втрати шифрувальної інформації.

Таким чином, ряд недоліків ставить під сумнів однозначність у перевагах використання еліптичних кривих у протоколах управління ключами. Ситуація вимагає посилення вектору дослідження у напрямі вивчення математичного апарату побудови криптографічних систем. Зокрема, на вістрі сучасної криптографії розглядається застосування поліному замість простого числа. Окрім того, існують рішення, що дозволяють знизити ризик злому – впровадження контрзаходів на апаратному рівні та на рівнях операційних систем і додатків – такі як встановлення вимог до розробників в галузі написання більш крипто-стійких шифрувальних модулів, а також впровадження в протоколи пристрою інструкцій з розміщення ключів таким чином, щоб зменшити ймовірність їхнього викрадення. Значної уваги вимагає розвиток соціальної інженерії та підвищення загального рівня знань з інформаційної безпеки користувачів.

#### ЛІТЕРАТУРА

1. Шнайер Б. Прикладна криптографія. Протоколи, алгоритми, вихідні тексти мовою Сі. - М.: Вид-во «Тріумф», 2003, - 816 с.
  2. Кнут Д.Е. Мистецтво програмування. - Т. 2. Напівчисленні алгоритми. - М.: Видавничий дім «Вільямс», 2004, - 832 с.
  3. Сمارт Н. Криптографія. - М.: Вид-во «Техносфера», 2006, - 528 с.
- Security Week. Утечка криптографических ключей посредством электромагнитного излучения / Information Security/Информационная безопасность. – 2016. - №4

Рецензент: к.т.н., доц. КПП ім. Ігоря Сікорського  
В.П. Полторак





**ПІДГОТОВКА КАДРІВ У  
ГАЛУЗІ ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ**

**TRAINING IN INFORMATION  
TECHNOLOGIES FIELD**



# Особенности формирования программ подготовки магистров информационной безопасности

Гришин И.Ю., Тимиргалеева Р.Р.

Институт компьютерных систем и информационной безопасности

Кубанский государственный технологический университет

Краснодар, Россия

igugri@gmail.com, renatimir@gmail.com

**Аннотация.** В работе рассмотрен подход к формированию программы магистратуры, направленной на обучение и воспитание магистров в области информационной безопасности и базирующейся на компетентном подходе с учетом требований профессиональных стандартов.

**Ключевые слова:** программа, магистратура, учебный план, информационная безопасность, компетенция, профессиональный стандарт.

## ВВЕДЕНИЕ

В настоящее время организации все чаще сталкиваются с ситуацией, когда вследствие различного вида атак на информационную инфраструктуру возникает ситуация прерывания своей обычной деятельности либо нарушения такой деятельности. Например, успешные атаки на информационные системы банковских и торговых организаций в 2016-2017 годах принесли убытки в миллиарды долларов по всему миру [1–2].

В соответствии с ГОСТ Р 53114-2008 «Защита информации. Обеспечение информационной безопасности организации. Общие рекомендации» объектом защиты являются бизнес-процессы организации, осуществляемые с использованием информационных технологий.

Опыт лучших практик показывает (например, ISO/IEC 27002), что выполнение требований по информационной безопасности служит общей цели развития бизнеса и обеспечения его непрерывности, поскольку защита информационных активов от различного вида угроз способна минимизировать риски и максимизировать возврат инвестиций [3–4].

Здесь прослеживается постепенный переход от командных методов управления информационной безопасностью к более гибким методам, учитывающим экономическую эффективность применяемых методов, основанных на конвергенции методов системного анализа, информационных технологий, экономической эффективности и защиты информации.

В соответствии с указанными тенденциями, опираясь на новые стандарты высшего образования поколения 3+ (ФГОС ВО), а также проекты стандартов поколения 3++ (ФГОС ВО с учетом профессиональных стандартов) в Кубанском государственном технологическом университете разработана и реализуется программа магистратуры «Обеспечение информационной безопасности и непрерывности бизнес-процессов» в рамках

направления 10.04.01 – Информационная безопасность.

Миссией данной программы является обеспечение государства и организаций высококвалифицированными кадрами, способными противостоять большим вызовам для общества, государства и науки, обладающих компетенциями в области информационной безопасности, кибернетики для решения задач эффективного противодействия киберугрозам, создания передовых цифровых технологий, систем обработки больших объемов данных и искусственного интеллекта.

## ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

Область профессиональной деятельности выпускников, освоивших программу магистратуры, включает сферы науки, техники и технологии, охватывающие совокупность проблем, связанных с обеспечением информационной безопасности и защиты информации.

В соответствии с профилем и спецификой магистерской программы уточнённая область профессиональной деятельности: сферы науки, техники и технологии, охватывающие совокупность проблем, связанных с обеспечением непрерывности и информационной безопасности основных бизнес-процессов организаций (процессов государственного и муниципального управления), в условиях существования угроз в информационной сфере.

По профилю подготовки выпускники могут работать в научно-исследовательских и проектных учреждениях, органах государственной власти и местного самоуправления, на предприятиях, в кредитно-финансовых учреждениях и вузах.

Объектами профессиональной деятельности выпускников, освоивших программу магистратуры, являются:

- фундаментальные и прикладные проблемы информационной безопасности;
- объекты информатизации, информационные ресурсы и информационные технологии, компьютерные, автоматизированные, телекоммуникационные, информационные и информационно-аналитические системы, обеспечивающие основные бизнес-процессы организации;
- средства и технологии обеспечения непрерывности и информационной безопасности бизнес-процессов;

- экспертиза, сертификация и контроль защищённости информации и объектов информатизации;

- методы и средства проектирования, моделирования и экспериментальной отработки систем, средств и технологий обеспечения информационной безопасности объектов информатизации;

- организация и управление обеспечением непрерывности и информационной безопасности бизнес-процессов;

- образовательный процесс в области информационной безопасности.

На основе изучения рынка труда, товаров и услуг региона были выбраны виды профессиональной деятельности. Видами профессиональной деятельности, к которым готовятся выпускники, освоившие программу магистратуры, являются: проектная, научно-исследовательская, организационно-управленческая.

#### ПЕРЕЧЕНЬ ФОРМИРУЕМЫХ КОМПЕТЕНЦИЙ

Компетенции ФГОС ВО, соответствующие выбранным видам профессиональной деятельности соответствуют рекомендованным ФГОС ВО направления 10.04.01 – Информационная безопасность.

Компетенции, самостоятельно разработанные образовательной организацией в дополнение ФГОС ВО и соответствующие профилю магистерской программы, профессионально-специализированные компетенции:

- способность на практике применять стандарты обеспечения информационной безопасности и непрерывности бизнес-процессов;

- способность проводить оценку рисков непрерывности бизнес-процессов на основе угроз информационной безопасности;

- способность к стратегическому управлению обеспечением непрерывности бизнес-процессов и информационной безопасности организации.

Учебный план имеет общую трудоемкость 240 зачетных единиц.

Общее руководство научным содержанием и образовательной частью магистерской программы осуществляется штатным научно-педагогическим работником кафедры компьютерных технологий и информационной безопасности, имеющим учёную степень доктора технических наук и учёное звание профессора соответствующего профиля. Руководитель магистерской программ ведёт

самостоятельные научно-исследовательские проекты, имеет публикации в реферируемых журналах.

Перечень формируемых компетенций выпускника магистерской программы соответствует профессиональным стандартам: «Специалист по технической защите информации», «Специалист по технической защите информации», «Специалист по безопасности компьютерных систем и сетей» [5].

#### ВЫВОДЫ

Разработанная магистерская программа используется для обучения студентов магистратуры, а в перспективе возможно применение основных положений ее для организации переподготовки и повышения квалификации.

#### ЛИТЕРАТУРА

1. Тимиргалеева Р. Р. Обеспечение информационной безопасности и непрерывности бизнес-процессов при использовании мобильных технологий / Р. Р. Тимиргалеева, И. Ю. Гришин // Цифровая экономика и «Индустрия 4.0»: проблемы и перспективы. труды научно-практической конференции с международным участием, 2017. – С. 489–493.

2. Тимиргалеева Р. Р. Проблемы обеспечения информационной безопасности и непрерывности бизнеса / Р. Р. Тимиргалеева, И. Ю. Гришин. // Научно-методический электронный журнал Концепт. – 2017. – №9. – С. 131–137.

3. Тимиргалеева Р. Р. Организационно-экономические аспекты формирования системы обеспечения информационной безопасности и непрерывности бизнеса / Р. Р. Тимиргалеева, И. Ю. Гришин. // Конкурентоспособность в глобальном мире: экономика, наука, технологии. – 2017. – №8. – С. 115–119.

4. Grishin I. Y. The application of artificial intelligence methods for forming industry management systems / I. Y. Grishin, R. R. Timirgaleeva. // CEUR Workshop Proceedings. 1. Сер. "Selected Papers of the 1st International Scientific Conference Convergent Cognitive Information Technologies, Convergent 2016". – 2016. – С. 115–120.

5. Кучер В. А. Простой подход к применению технологий электронного обучения при подготовке студентов направления «информационная безопасность» / В. А. Кучер, И. Ю. Гришин, Р. Р. Тимиргалеева. // Филологические и социокультурные вопросы науки и образования. Сборник материалов I Международной научно-практической конференции. – 2016. – С. 261–271.

# Моделювання типових технологічних процесів засобами CoDeSys

Драбинко Віталій Павлович

КПІ ім. Ігоря Сікорського  
Київ, Україна  
v.drabyenko@gmail.com

Жеребко Валерій Анатолійович

КПІ ім. Ігоря Сікорського  
Київ, Україна  
valery.zherebko@gmail.com

**Анотація.** В даній роботі було розглянуто моделювання процесів та систем в інструментальному програмному комплексі CODESYS. Було змодельовано одні з найпоширеніших процесів, такі як гідродинамічні, термодинамічні та інші. Було зроблено порівняння з іншими програмними комплексами такого ж типу. Було зроблено висновок, що використання програмного комплексу CODESYS призводить до оптимізації підготовки кадрів, адже простота програмування процесів і систем є очевидною складовою цього комплексу.

**Ключові слова:** моделювання, оптимізація, процеси та системи.

## ВСТУП

В наш час дуже важливо, щоб студенти отримували якісну освіту і ставали висококваліфікованими спеціалістами. Особливо це стосується студентів різних технічних спеціальностей. Одним із важливих етапів їхнього навчання є програмування. В цьому і може допомогти інструментальний програмний комплекс CODESYS. Він був розроблений компанією 3S-Smart Software Solutions GmbH [1]. Основою комплексу CODESYS є середовище розробки прикладних програм для програмованих логічних контролерів (ПЛК) [2].

Було розглянуто актуальність та ефективність використання інструментального програмного комплексу CODESYS у вищих навчальних закладах. Актуальність залучення CODESYS в навчальний процес є неймовірно великою, адже це програмний комплекс який постійно оновлюється і до того ж є безкоштовним. В ході даної роботи нами було розроблено декілька технологічних процесів та систем на різних мовах, що є доступними в програмному комплексі CODESYS, такі як термодинамічний, гідродинамічний, рух механічних систем та були промодельовані електричні схеми та системи.

## РЕАЛІЗАЦІЯ CODESYS В ПРАКТИЧНИХ ЦІЛЯХ

Чому ж було обрано саме CODESYS в порівнянні з іншими інструментальними програмними комплексами? По-перше, в CODESYS реалізована можливість створювати програми на всіх п'яти мовах програмування, визначених міжнародним стандартом IEC 61131-3. Це такі мови як IL, ST, LD, FBD та SFC. Структура проекту не є складною адже вона є деревовидною та зрозумілою. Вона складається з менеджера бібліотек, головного тіла програми, трейсу (графічного представлення

результатів) і само собою головною програмою (кодом). Часткова реалізація гідродинамічного процесу виглядає так:

```
PROGRAM PLC_PRG
VAR
    lab1_model : model1_fb; // вибір схеми для
запуску: model1_fb, model2_fb, model3_fb
    valve_in1:BOOL:=FALSE; //Значення
клапана A. false - закритий, true - відкритий
    valve_out2:BOOL:=FALSE; //Значення
клапана B. false - закритий, true - відкритий
    i:INT;
    dt1:INT:=20; //час виконання завдання, один
цикл - 1 с, час одного циклу задається в Task
END_VAR
VAR_OUTPUT
    LE1:REAL:=0.0; //поточний рівень рідини, м
END_VAR
Реалізація:
sq_tank1:=3.14 * r_tank1 * r_tank1;
IF NOT valve_in1 AND valve_out2 THEN
    q1:= q1 - q_out1;
    LE1:= q1 / (density * sq_tank1);
END_IF
IF valve_in1 AND valve_out2 THEN
    q1:=q1 + q_in1 - q_out1;
    LE1:= q1 / (density * sq_tank1);
END_IF
```

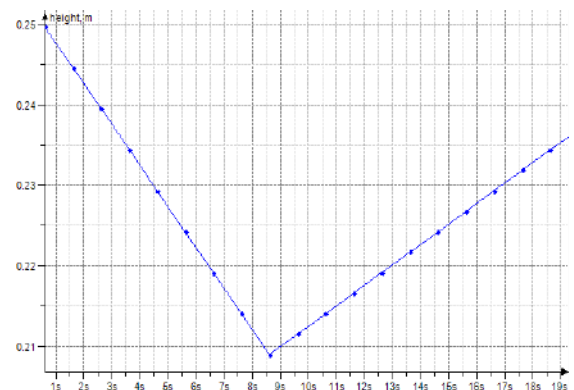


Рис. 1. Залежність висоти води в баку від часу

В даному випадку технологічним об'єктом керування виступає бак з рідною певної форми. В залежності від вибору моделі його фізичні характеристики змінюються.

### РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ В СЕРЕДОВИЩІ CODESYS

Для початку реалізації проекту, необхідно обрати скільки буде виконано кроків дискретизації та самий час виконання цих кроків. Для цього процесу було обрано гідродинамічний процес з кроком дискретизації в 1 секунду та часом виконання в 19 секунд, і отримано такі результати (рис. 1).

Даний графік описує один з гідродинамічних процесів, такий як залежність витоку води з баку при включенні подачі води, як тільки рівень води досягне рівня трубки. Як видно на графіку початковий рівень води становив 0.25 м і повільно падав вниз, але як тільки рівень води досягнув позначки трубки, то одразу ж бак почав наповнюватись і паралельно стікати, що і видно на даному графіку.

### ПЕРЕВАГИ CODESYS

Надзвичайно важливим елементом в CODESYS є інструмент графічного трасування значень змінних. Дуже зручним в програмному комплексі є наявність вбудованої системи візуалізації, що дозволяє виконувати на практиці досить складні проекти автоматизації.

Незамінною в освітньому процесі є можливість додавання власних бібліотек і програмних розширень нарівні зі стандартними. Це відкриває чимало можливостей для створення навчальних програм з вивчення автоматизації та моделювання процесів і систем.

Використання цього комплексу надає студентам не тільки теоретичні навички, але й практичні та професійні. Студенти, які мають практичні навички користування ним, з легкістю увійдуть на робочі місця.

### ПРАКТИЧНА ЕФЕКТИВНІСТЬ CODESYS

Для перевірки ефективності використання даного програмного комплексу на практиці було проведено моделювання термодинамічного процесу в комплексі CODESYS. У цій програмі було змодельовано процес нагрівання деякого об'єкта з певними фізичними властивостями (сталеві кулька) та розраховано залежність температури від часу при відомій залежності підвідної потужності від часу.

Як бачимо графік дуже точний, адже на ньому видно початкову температуру досліджуваного

об'єкта і крок дискретизації був заданий в 0.01 секунди. В CODESYS було використано мову

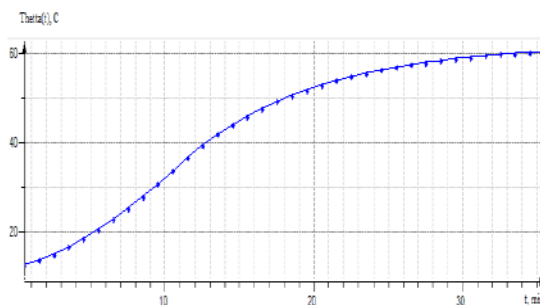


Рис. 2. Залежність температури від часу в CODESYS

програмування ST, що зайняло небагато часу і дозволило покроково спостерігати за проходженням процесу.

### ВИСНОВКИ

В ході написання цієї статті був розглянутий інструментальний програмний комплекс CODESYS для моделювання технологічних процесів та систем.

Використовуючи всі переваги надані CODESYS було змодельовано декілька процесів та систем. Було зроблено висновок, що наш вибір є найбільш актуальним та ефективним, в порівнянні з іншими варіантами. Тому є доцільним використання цього комплексу у навчальному процесі як безкоштовний універсальний комплекс. Він є простим для складання програмних моделей й імітаційного моделювання систем керування на прикладі розроблених моделей технологічних процесів та систем.

Результати даної розробки впровадженні в навчальний процес (дисципліна «Моделювання процесів та систем») кафедри автоматики та управління в технічних системах, факультету інформатики та обчислювальної техніки КПІ ім. Ігоря Сікорського.

### ЛІТЕРАТУРА

1. Jochen Petry IEC 61131-3 mit CoDeSys V3: Ein Praxisbuch für SPS Programmierer. – 2011. – 839с.
2. Петров И. В. Програмируемые контроллеры. Стандартные языки и приёмы прикладного проектирования / Под ред. проф. В. П. Дьяконова. – М.: СОЛОН-Пресс, 2004. – 256 с.

# **ABSTRACTS**





# INFORMATION SYSTEMS AND TECHNOLOGIES

Page 11

## *The Interactive Web Application With Cloud Data Storage Usage*

Volodymyr Samotyy  
Cracow University of Technology  
Cracow, Poland  
Lviv State University of Life Safety  
Lviv, Ukraine  
vsamotyj@gmail.com

Ulyana Dzelendzyak, Vitaliy Palyushok  
Lviv Polytechnic National University  
Lviv, Ukraine  
u.dzelendzyak@gmail.com

*Abstract.* The interactive system "SHELTER" which consists of database, data warehouse, a progressive mobile application and basic web interface has been developed. The system gives an opportunity to create, to send, to work out user applications and review, change, delete, transfer the system's elements, manage additional features of the application, which are not inaccessible to the ordinary user.

*Keywords:* non-relational data base, data warehouse, progressive mobile application, web interface, cloud technology.

---

Page 13

## *Insults Detection On Short Texts*

Andrii Popko  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
andrew.popko.j@gmail.com

Yaroslav Dorohyi  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
cisco.rna@gmail.com

*Abstract.* Theses of the report contain a description of the choice of a mathematical model for solving the problem of classification of short texts for the presence of insults in them. The paper briefly describes the research of methods for their detection. The authors proposed their own model of classification with a description of improvements that increase the efficiency of its work.

*Keywords:* classification of texts, machine learning, methods of machine learning, logistic regression.

---

Page 15

## *Development Of Technology For 3D-Modeling Of Geological Objects*

Ualkhanova Ainur, Denisova Natalya  
D. Serikbaev East Kazakhstan State Technical University  
Ust-Kamenogorsk, Kazakhstan  
AUalkhanova@ektu.kz, NDenisova@ektu.kz

Krak Iurii  
Taras Shevchenko National University  
Kyiv, Ukraine  
Yuri.krak@gmail.com, Krak@univ.kiev.ua

*Abstract.* The paper is devoted to researching of 3D visualization technologies for various geological objects, small and occupying extensive territories. In the paper, features, problems and technologies for constructing 3D object models based on photogrammetry are investigated. The possibilities of using the software product Agisoft Photoscan, allowing in semi-automatic mode to receive a point cloud and 3d-models of objects on the basis of their two-dimensional photos are considered.

*Keywords:* photogrammetry, 3D-model, 3D-modeling, point cloud, orthomosaic.

---

Page 17

# *The Use Of Amazon Connect For Call Center Automation*

Rodion Kharabet  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
pensioneer4ik@gmail.com

Yaroslav Dorogoy  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine

*Abstract.* The article considers automation of modern call centers and offers the use of Amazon Connect as the solution for this issue. The article also examines the components of Amazon Web Services that can help to operate a powerful and flexible customer service system on the basis of Amazon Connect.

*Keywords:* call center, Amazon Connect, automation, customer service, Amazon Web Services, call.

---

## **Page 19**

# *Analysis Of Algorithms For Optimizing The Calculation Of The Rosenbrock's Function Local Minimum*

Oleh Pysarenko  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
oa.pisarenko@gmail.com

*Abstract.* The work demonstrates an analysis of optimization algorithms by solving Rosenbrock function using the LabView tool. A Downhill Simplex method and Genetic Algorithm were chosen. The principle of algorithms work is described. The local minimum of Rosenbrock function is calculated. The number of iterations performed for each algorithm was compared. The best rate of performance has been shown by using Genetic Algorithm.

*Keywords:* Rosenbrock function, Genetic Algorithm, Downhill Simplex, LabView, optimization.

---

## **Page 21**

# *Automation Of Instrumental Study Of Parameters Of Cable Lines*

Oleg Morgal  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine

Serhii Shykhutskyi  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine

*Abstract.* Certification tests of lines of structured cable systems (SCS) to meet the requirements of certain categories or classes do not give a complete map of the current state of the SCS and do not have tools for comparative analysis to detect changes in the state of the systems. The authors propose to supplement the test program with measurements of parameters TDR and TDX and to automate the method of comparative analysis of measurements in order to detect deviations in the state of the system.

*Keywords:* SCS, TDR, TDX

---

---

**Page 23**

## *Chatbot Based On Neural Networks*

Boiaryn Oles  
Igor Sikorsky Kyiv Polytechnic  
Institute  
Kyiv, Ukraine  
boiaryn.o@gmail.com

Diachuk Ivan  
Igor Sikorsky Kyiv Polytechnic  
Institute  
Kyiv, Ukraine  
diachukivan96@gmail.com

Dorogiy Yaroslav  
Igor Sikorsky Kyiv Polytechnic  
Institute  
Kyiv, Ukraine  
argusyk@gmail.com

**Abstract.** The analysis and comparison of different architectures based on neural networks for the implementation of the chatbot. The most successful architecture was chosen to fulfill the task.

**Keywords:** neural networks, chatbot, memory networks, LSTM, data science.

---

**Page 26**

## *Index Structure Optimizations For In-Memory Data Storage Systems*

Oleksandr Podrubailo, Vasyl Saverchenko  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
alex.podrubailo@gmail.com

**Abstract.** In-memory data grid systems uses RAM as main storage device, which takes the performance of such systems on the new level. However, new in-memory systems often uses old data structures and algorithms which were proposed for relational database management systems and are not suitable for main memory, in particular, they do not take into account number of cache misses. This paper describes methods of index structure optimizations for in-memory storage solutions, which will allow to significantly reduce the number of comparatively slow reads from main memory.

**Keywords:** in-memory data grid, deleting key fields, rearrange fields.

---

**Page 29**

## *Comparison Of The Machine Learning Frameworks*

Yaroslav Dorohyi  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
cisco.rna@gmail.com

Kseniia Levchenko  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
ksulevchenko95@gmail.com

**Abstract.** Theses of the report contain a comparison of the existing machine learning frameworks for Python language. The paper briefly describes pros and cons of every chosen framework. The authors gave their own recommendations for using each of the chosen framework.

**Keywords:** image recognition, machine learning, convolutional neural networks, machine learning frameworks.

---

**Page 32**

## ***Situational Management Of Urban Transport With Intelligent Support For Dispatch Solutions***

Stenin A., Pasko V., Lemeshko V., Rusakova A.  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
alexander.stenin@yandex.ua

*Abstract.* The article proposes an algorithm of fuzzy situation control, which is the basis of an intelligent ADSS for managing a specific type of public transport. This algorithm is based on the method of the PROBA, which makes it possible to select the optimal control alternatives, which were generated automatically on the basis of the expert survey method.

*Keywords:* urban transport management, knowledge base of the subject area, fuzzy situation network, ADSS.

---

**Page 34**

## ***Extending Of Stirling Engine Usage Area For Supporting Alternative Residential Power Supply***

Fedir Ohulchansky  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
fedor.ohul@gmail.com

Natalya Repnikova  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
n.repnikova@gmail.com

*Abstract.* The work focuses on the rationale of solar energy and the Stirling engine combination possibility for using it as efficient residential energy supply.

*Keywords:* stirling engine, residential, renewable energy.

---

**Page 36**

## ***Design Of User Activity Of Human-Machine Interface For Process Control Systems***

Leonid Yurchuk  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
leonidyu0@gmail.com

*Abstract.* The questions of designing of the user activity of the human-machine interface for the control systems of technological processes in accordance with the requirements and recommendations of the international standard ANSI / ISA-101.01-2015 - Human Machine Interfaces for Process Automation Systems are considered.

*Keywords:* man-machine interface, user, activity, process control systems, navigation.

---

---

**Page 38**

## *Non-blocking Database With ACID Transactions*

Kytsmen Dmytro  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine

*Abstract.* In order to achieve non-blocking transaction processing, it is proposed to use such transaction processing algorithms and data structures to further store new data that significantly contributes not only to optimizing and speeding up the database but also ensures the presence of ACID properties.

*Keywords:* atomicity, consistency, isolation, durability, availability, stability, transaction, concurrent access.

---

**Page 40**

## *Implementation And Research Of Artificial Neuron On Different Families Of FPGA*

Yurii Berdnyk  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
berdniky@gmail.com

*Abstract.* This paper presents the results of FPGA implementation of artificial neuron for the developed method of implementation of nonlinear activation functions and artificial neuron implementation algorithms. The obtained results of the research, which are presented in the table, are FPGA source and speed.

*Keywords:* neural networks, artificial neuron, activation functions, FPGA, VHDL.

---

**Page 42**

## *Research Of The Color Models Influence In Determining The Color Characteristics Of Recognition Objects On The Image*

Andrii Antosev  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
gibson12ant@gmail.com

Yaroslav Dorohyi  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
cisco.rna@gmail.com

*Abstract.* Analysis of images color peculiarities determination methods is carried out. The popular color models are considered and their features are outlined.

*Keywords:* color model, image processing, computer vision, RGB, HSV, Lab, recognition.

---

**Page 44**

## *On Choosing Model of Reliable Software Operation in Cloud Technologies*

Olena Savchuk  
Igor Sikorsky Kyiv Polytechnic  
Institute  
Kyiv, Ukraine  
savchuk\_11@ukr.net

Anastasia Kazantseva  
Taras Shevchenko KNU  
Kyiv, Ukraine  
lesnaja@bigmir.net

Oleksiy Pyrozhkov  
Igor Sikorsky Kyiv Polytechnic  
Institute  
a.y.pirozhkov@gmail.com

*Abstract.* In this article, models of reliable software functioning are considered. A dynamic model has been chosen to guarantee the confidentiality and security of information even when compromising the provider of cloud services or its infrastructure.

*Keywords:* models, software, cloud technologies.

## **CONTROL SYSTEMS**

**Page 49**

## *Algorithm of Coordinate Descent in the Method of Dynamic Programming for Digital Optimal Control Problems*

Iryna Mayer  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine

Andrii Pysarenko  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine om

*Abstract.* The modification of the algorithm for discrete optimal control problem by the dynamic programming method applying a coordinate descent method is proposed.

*Keywords:* dynamic programming, optimal control, method of uniform search, method of coordinate descent, simulation, MATLAB/Simulink.

**Page 51**

## *An Improved Formula For Calculating The Coefficients Of PD Controllers For The Synthesis Of Complex Systems*

Dmytro Nahaiko  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
nagayko.dima@gmail.com

Natalya Repnikova  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
n.repnikova@gmail.com

*Abstract.* In this paper the problems of synthesis of complex digital control systems with PD controller have been considered. An improved formula for calculating the coefficients of PD controllers for digital control systems, the dynamics of which is described by high-order equations, has been proposed. Experimental studies on models using the MATLAB / Simulink software package have been carried out. Approved adequacy and feasibility of the proposed formula.

*Keywords:* synthesis, controllability, response time, PD controller.



---

**Page 53**

## *Model Of Pedestrian Identification And Tracking System Using A Moving Light Source*

Mykhailo Khlivnenko  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine

*Abstract.* Based on the developed toolkit for the MATLAB/Simulink package, a model is proposed pedestrian detection and tracks their subsequent movement with the Kalman filter. Pedestrian identification is performed by using aggregated channel functions (ACF). The obtained results are used to construct a control influence on a hybrid stepper motor, which positions the source of light that illuminates a moving object. This subsystem is part of the automotive night vision system.

*Keywords:* pedestrian detection, Kalman filter, tracking of moving objects, traffic safety.

---

**Page 55**

## *Controlling Warehouse Robots Based On Synthesis Algorithm With Desirable The Roots Of The Digital System*

Vladislav Yakymchuk  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
Vlad22109@gmail.com

Natalya Repnikova  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
n.repnikova@gmail.com

*Abstract.* The paper substantiates the possibility of using the method of synthesis of digital control systems with a given pole position for controlling warehouse work. Get practical solutions for solving synthesis problems for systems whose dynamics are described by equations of high order.

*Keywords:* digital systems, poles, management.

---

## **PROGRAMMING TECHNOLOGIES**

**Page 59**

## *UML Diagrams Of Software For Moving Objects Remote Control Programm Complexes*

Vasyl Kravinskyi  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine

*Abstract.* A variant organization of UML diagrams for software moving objects remote control programm complexes were offered. The variant of combining different types of diagrams is proposed.

*Keywords:* remote control software complexes, remote control, UML.

---

# INFORMATION SECURITY

Page 63

## *Key Distribution*

Tymofii Radetskyi  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine

*Abstract.* Key management is the most difficult part of cryptography. The organization of the accumulation of keys is associated with their storage, recording and deletion. As a rule, key exchange is performed using the RSA algorithm or the more efficient Diffie-Helman algorithm. Before modern cryptography there is a problem of increasing the stability of algorithms and reducing the size of keys and data blocks. The obvious way of solving this problem is to present blocks of information in cryptographic algorithms using points of elliptic curves.

*Keywords:* Cryptographic systems, key distribution protocols, cyclic groups, elliptic curves, RSA, ECC.

---

# TRAINING IN INFORMATION TECHNOLOGIES FIELD

Page 69

## *Features Of The Formation Of Programs For The Education Of Masters Of Information Security*

Grishin Igor, Timirgaleeva Rena  
Kuban State Technological University  
Krasnodar, Russia  
igugri@gmail.com, renatimir@gmail.com

*Abstract.* The paper considers the approach to the formation of a master's program aimed at education and training masters in the field of information security and based on a competence approach, taking into account the requirements of professional standards.

*Keywords:* program, master's program, curriculum, information security, competence, professional standard.

---

Page 71

## *Modeling Of Typical Technological Processes By Means Of CODESYS*

Drabynko Vitalii  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
v.drabynko@gmail.com

Zherebko Valerii  
Igor Sikorsky Kyiv Polytechnic Institute  
Kyiv, Ukraine  
valery.zherebko@gmail.com

*Abstract.* In this paper, the simulation of processes and systems in the CODESYS instrumental software system was considered. We have been modeling some of the most common processes, such as hydrodynamic, thermodynamic, and others. A comparison was made with other software systems of the same type. It was concluded that the use of the CODESYS software system leads to optimization of training, since the ease of programming of processes and systems is an obvious component of this complex.

*Keywords:* simulation, optimization, processes and systems.