

И.В. Пономарев, Ю.М. Рыбка

АВТОМАТИЗИРОВАННОЕ РАЗМЕЩЕНИЕ ДАННЫХ НА ИНФОРМАЦИОННЫХ САЙТАХ

Аннотация. Предложена программа автоматического размещения данных на интернет ресурсах.

Ключевые слова: автоматизация, послойная архитектура, коммуникация с интернет ресурсами, асинхронная модель взаимодействия, XML.

Введение. В связи с бурным развитием рекламы в интернете большую популярность приобрели сервисы бесплатного размещения рекламы – интернет- доски объявлений.

Возникла проблема размещения информации на возможно большем количестве сайтов и быстрого ее обновления по мере необходимости. Таким образом,

актуальной является разработка программы автоматического размещения данных на информационных сайтах.

Постановка задачи. Необходимо разработать основу программы реализующей базовый функционал системы автоматического размещения данных на информационных интернет ресурсах.

Основная часть. В результате анализа внутренней структуры типовых досок объявлений была найдена формальная схема взаимодействия с интернет-ресурсами такого рода.

В основе архитектуры продукта используется шаблон послойной архитектуры (Layer Architecture Pattern) [1]. При этом разработанное программное решение состоит из трех слоев:

1. Слой взаимодействия с интернет ресурсами.
2. Планировщик.
3. Графический интерфейс пользователя.

В качестве платформы для реализации выбран .Net Framework – мощный, современный, удобный в использовании, инструмент разработки прикладных программ, предоставляющий обширную стандартную библиотеку [2].

Взаимодействие с интернет ресурсами. Слой взаимодействия с интернет ресурсами определяет общую схему взаимодействия с некоторой абстрактной доской объявлений. Разработанная стандартная схема состоит из следующих шагов:

- проверка необходимости входа в систему,
- вход в систему с указанием пароля и логина,
- переход на страницу управления списками объявлений,
- сбор ссылок на существующие списки объявлений,
- актуализация существующих объявлений,
- добавление новых объявлений.

Реализация слоя коммуникации с интернет-ресурсами представляет иерархию классов, состоящую из базового абстрактного класса `AdsPlacer` на вершине и наследников. Базовый класс описывает каркас коммуникации с абстрактной доской объявлений, при этом, каждый шаг стандартной схемы реализован при помощи виртуального метода, так что конкретные наследники могут вносить произвольные коррективы в схему взаимодействия. Производные классы реализуют коммуникации с каждым конкретным интернет ресурсом, посредством перегрузки методов базового класса.

Диаграмма класса `AdsPlacer` представлена на рисунке 1.

Для непосредственной коммуникации с веб-страницами базовый класс использует стандартный компонент `WebBrowser` из библиотеки `Windows.Forms`. Этот компонент инкапсулирует возможности браузера `Internet Explorer` и предоставляет удобный интерфейс навигации по страницам и работы с содержимым страниц. Для работы с содержимым веб-страниц используется компонент `HtmlDocument`.

Для задания информации взаимодействия с конкретными досками был спроектирован простой XML формат [2]. Согласно этому формату теги верхнего уровня являются уникальным идентификатором поля, а их содержимое задает критерии, по которым программа сможет найти это поле на странице. Пользовательские объявления, которые также хранятся в XML-файле, могут задавать значения, которые нужно поместить в поля.

Взаимодействие с интернет-ресурсами реализовано согласно асинхронной модели. Для этого в классе используется очередь объектов-действий.

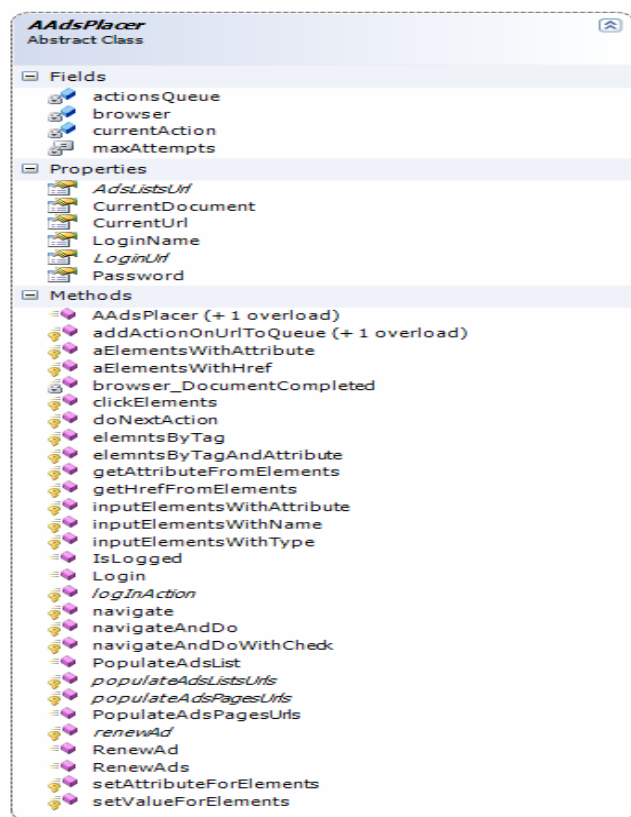


Рисунок 1 - Диаграмма класса AdsPlacer

Метод `Navigate` загружает документ по указанному адресу. По окончании загрузки документа, генерируется событие `DocumentCompleted`, которое обрабатывается во втором потоке. Вспомогательный класс `ActionOnUrl` представляет действия, которые должны быть выполнены при переходе компонента `WebBrowser` к заданному адресу, и используется при реализации очереди действий.

Таким образом, класс `AdsPlacer` создает и управляет очередью необходимых действий и выполняет все навигации. Предоставляется широкий набор методов для удобной обработки HTML-документов.

Планирование и графический интерфейс. Слой планирования обновлений определяет запуск заданий с определенным интервалом, указанным пользователем. Реализован класс `Scheduler` на основе компонента `Timer` стандартной библиотеки.

Графический интерфейс пользователя представляет собой окно, в котором выводится интерфейс браузера, отображающий содержимое обрабатываемой в данный момент страницы. Браузер используется, чтобы пользователь мог проследить за автоматизированным процессом обновления и при необходимости внести коррективы. Поверх

браузера в левом верхнем углу размещены элементы управления – поля логина и пароля, поле задания интервала времени для повтора сеанса обновления рекламных данных, а также элемент управления планированием.

Пример реализации. В качестве примера реализации рассмотрим автоматическое обновление объявлений на сайте `gorod.dp.ua`.

Работу с этой доской объявлений выполняет класс `GorodDpUaAdsPlacer` — наследник класса `AdsPlacer`. Он переопределяет виртуальные методы следующим образом:

`LoginUrl` — возвращает адрес строки авторизации `"http://gorod.dp.ua/user/login.php"`

`AdsListsUrl` — возвращает адрес строки, содержащей ссылку на списки рекламных блоков `"http://gorod.dp.ua/gazeta/user.php"`

`loginAction` — выполняет вход в систему: устанавливает значение атрибутов «value» элементов с тегом «input» и значением атрибутов «name» равным «username» и «password» имени пользователя и пароля соответственно.

`populateAdsPagesUrls` - возвращает коллекцию значений атрибута «href» элементов с тегом «a», содержащих в значении атрибута «href» строку «?edit&» - список ссылок на конкретные рекламные блоки.

`renewAd` — обновляет блок рекламной информации:

- инициирует нажатие на всех элементах с тегом «input» и значением атрибута «name» равным «Submit» - нажатие кнопки «редактировать» на странице.

Выводы. В работе рассмотрена основа программного решения для автоматического размещения данных на интернет-ресурсах. Определена архитектура продукта, разработана общая схема взаимодействия с некоторым абстрактным информационным сайтом, позволяющая применять это решение для любых досок объявлений.

ЛИТЕРАТУРА

1. Мартин Фаулер. Шаблоны корпоративных приложений. Пер. с англ. – М.: Издательский дом «Вильямс», 2012-544 с.
2. Эндрю Троелсен. Язык программирования C# 2010 и платформа .NET 4. Пер.с англ. – М.: Издательский дом «Вильямс», 2011-1054 с.

В.В. Герасимов, І.В. Пономарьов, І.О. Щетінін
**ОПТИМІЗАЦІЯ ПРОЦЕСУ РОЗРОБКИ ІНТЕРНЕТ-ДОДАТКІВ
НА ПЛАТФОРМІ JAVA**

Анотація. Розглянуто Play Framework, який значно спрощує та пришвидшує процес розробки Інтернет-додатків на платформі Java. Описано особливості використання та переваги цієї технології. Представлено результати порівняння Play Framework з альтернативними технологіями.

Ключові слова: Play Framework, MVC, HTTP, View, Controller, JPA, log4j, Groovy, Selenium, JSP, Spring Framework, HTML, JSON, XML, MySQL.

Постановка проблеми. Активний розвиток глобальної мережі Internet призвів до виникнення та популярності web-додатків. Існує безліч технологій для їх створення, проте не завжди процес розробки є простим та швидким. Наприклад, часто вирішення таких типових задач як взаємодія з базами даних або тестування додатків вимагає значних зусиль від розробника та є причиною написання «надлишкового» коду. Розробники нової технології Play Framework спробували вирішити цю проблему.

Основна частина. Play framework є альтернативою «роздутим» Java EE технологіям. Він фокусується на продуктивності розробки та цілях RESTfull архітектури. Play є дуже ефективним засобом при гнучкій розробці програмного забезпечення. Мета Play framework — дотримуючись Java, зробити розробку web додатків простішою. За думкою його розробників простішими речами можна зробити складніші речі.

Play — це чистий Java фреймворк, який дозволяє користуватись улюбленими засобами розробки та бібліотеками. При наявності досвіду користування платформою розробки Java не має необхідності переходити на іншу мову, інше IDE та бібліотеки. Достатньо просто перейти на продуктивніші технології Java.

Розглянемо більш прискіпливо головні особливості Play framework'у.

HTTP маршрутизація (HTTP-to-code mapping). Інші Java Web фреймворки такі як Servlet API або Struts framework змушують мати справу з абстрактним відображенням HTTP протоколу та дивними API концепціями.

Play побудований інакше: фреймворк web-додатків має повний доступ до HTTP та його концепцій. Це одна з фундаментальних різниць між Play та іншими Java web-фреймворками.

HTTP, модель запит/відповідь, архітектурний стиль REST, URI — це головні ідеї Play framework.

Наприклад, зв'язування URI з викликом Java виконується одним рядком:

```
GET /clients/{id} Clients.show
```

Ефективна шаблонна система. Ідея JSP та мови виразів JSTL досить непогана. Проте необхідно багато конфігураційних файлів для створення бібліотеки тегів. Також не має можливості мати доступ до базової моделі об'єкта. JSP має багато обмежень і це дійсно розчаровує. Play використовує шаблонну систему на кшталт JSP, яка позбавлена цих недоліків.

Наприклад, в JSP та JSTL маємо такий програмний код:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
<c:choose>
  <c:when test="${emails.unread != null &&
fn:size(emails.unread)}">
    You have ${fn:size(emails.unread)} unread email(s)!
  </c:when>
  <c:otherwise>
    You have no unread emails!
  </c:otherwise>
</c:choose>
```

Еквівалент в шаблонній системі Play буде виглядати наступним чином:

```
You have ${emails.unread ?: 'no'}
${emails.unread?.pluralize('email')} !
```

Мова виразів, що використовується шаблонною системою Play, називається Groovy, та є послідовником мови Java. В той час поки Play головним чином використовує шаблонну систему для побудови HTML відповідей, розробники мають можливість використовувати її для генерації будь-яких інших документів, таких як e-mail повідомлення, сторінки JSON формату та ін.

Більш потужне JPA. JPA — це в чистому вигляді засоби об'єктно-реляційного представлення даних (object-relational mapping — ORM), доступні для мови Java. Робота з JPA в Play стала ще простішою та ефективнішою. Без спеціального налаштування Play автоматично запускає менеджер сутностей (JPA Entity Manager) та чарівним образом синхронізується, коли перезавантажується програмний код.

Більш того, якщо використовувати суперкласс `play.db.jpa.Model`, він стане у нагоді в прагненні зробити програмний код правильнішим та кращим. Наприклад:

```
public static void messages(int page) {  
    User connectedUser = User.find("byEmail", connected()).first();  
    List<Message> messages = Message.find("user = ? and read =  
false order by date desc", connectedUser).from(page * 10).fetch(10);  
    render(connectedUser, messages);  
}
```

Розробка через тестування. Інтегрований запуск тестів в Play значно полегшує їх розробку. Є можливість писати всі види тестів, від простих до функціональних, та запускати їх спеціальними play командами в командному рядку за допомогою середовища Selenium. Керувати виконанням тестів можна через браузер. Підтримуються розрахунки тестового покриття. Якщо підключити спеціальний модуль, наприклад Cobertura, то можна отримати детальніші результати тестування.

Повноцінний фреймворк додатків. Play framework має всі засоби для створення сучасного web додатку:

- Підтримка реляційних баз даних через JDBC.
- Об'єктно-реляційне представлення даних через Hibernate (з JPA API).
- Інтегрована підтримка кешу з легким використанням системи розподіленого кешу в пам'яті.

- Пряме використання web-сервісів або в JSON, або в XML (маються на увазі реальні web-сервіси, не SOAP).
- Підтримка OpenID для розширеної аутентифікації.
- Web додаток може бути розгорнутим де завгодно (серверах додатків, сервісах Google App Engine, хмарах тощо).
- API для роботи із зображеннями.
- Модульна архітектура дозволяє комбінувати додаток з іншими додатками та технологіями.

Чому саме Play Framework. Фреймворк розроблений для того, щоб якомога швидше почати розробляти і при цьому відразу бачити проміжний результат.

Фреймворк має вбудований (embedded) web-сервер і настраюваний classloader. Які можливості це дає? У першу чергу запуск проекту без попереднього налаштування серверів Tomcat або подібних та перекомпіляція вихідних кодів «на льоту». Тобто при додаванні нового обробника запиту достатньо перезапустити сторінку в браузері. Винятки треба зробити для прекомпільованих ресурсів: плагінів, бібліотек тощо. При додаванні бібліотеки додаток необхідно перезапустити.

- Фреймворк вже включає в себе інструменти першої необхідності і дозволяє не витрачати часу на їх початкову настройку.

Наприклад, в Play Framework реалізовані власні бібліотеки для спрощення роботи з JPA та log4j. Не потрібно реалізовувати підключення до бази даних в програмному коді за допомогою JDBC, це можна легко зробити в файлі конфігурації додатку application.conf. Для того, щоб Java клас став моделлю JPA в Play Framework він має бути нащадком класу play.db.jpa.Model, його налаштування здійснюється за допомогою анотацій визначених в пакеті play.db.jpa. Для логування Play надає стандартний клас play.Logger, налаштувати рівні логування можна в файлі conf/log4j.properties. Розробниками Play створено безліч модулів для вирішення типових задач. Наприклад, модуль Secure реалізовує авторизацію та реєстрацію користувачів. Модулі легко підключаються, налаштовуються та мають відкритий код.

- Зручний template-engine.

За умовчанням використовується Groovy, проте можна використовувати і іншу технологію, наприклад Japid. Groovy надає багато

можливостей. Це і побудова логічних умов в шаблонах, робота зі списками та масивами, оголошення змінних, шаблонне наслідування, багато зручних тегів, наприклад, `{form}`, та ін.

- Фреймворк має відкритий код та підтримує сторонні плагіни.

Вже існує безліч сторонніх плагінів, що дозволяють, наприклад, інтегруватися з MongoDB, GAE тощо.

- TDD (керування запуском тестів).

За допомогою команди `play test` додаток запускається в режимі виконання всіх unit, функціональних та selenium тестів.

MVC в Play Framework. Модель MVC (рис. 1) розділяє додаток на окремі прошарки: прошарок презентації та прошарок моделей (model). Прошарок презентації далі розділяється на представлення (view) та прошарок контролерів (controller).

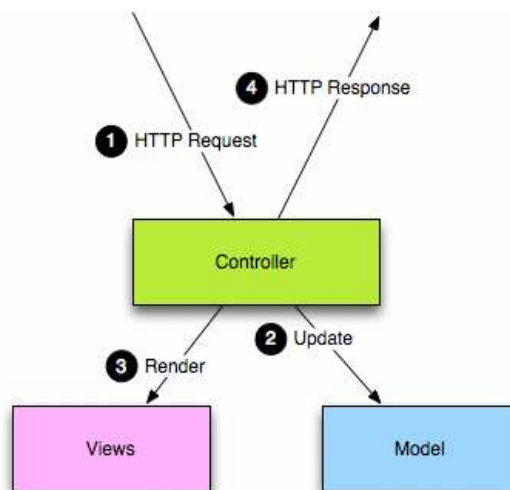


Рисунок 1 - Модель MVC в Play framework

Модель (model) є предметно-орієнтоване подання інформації, з якою працює додаток. Предметна логіка додає сенс даним — наприклад, розрахунки днів народжень користувачів, кошиків покупців тощо. Більшість додатків використовують постійний механізм зберігання даних, такий як база даних.

Представлення (view) надає моделі форми, придатні для взаємодії. Зазвичай це інтерфейс користувача. Кілька уявлень можуть існувати для однієї моделі, але для різних цілей. В web додатках представлення зазвичай надаються у web форматах, таких як HTML, XML або JSON. Однак є деякі випадки, коли подання може бути виражено в двійковому вигляді, наприклад, динамічні графіки діаграм.

Контролер (controller) відповідає на події (зазвичай дії користувача), обробляє їх та може також змінювати логіку моделей. В веб-додатках подіями зазвичай є HTTP запити: контролер приймає HTTP запит, витягує необхідні дані з події, такі як рядок параметрів запиту, заголовок запиту тощо. За допомогою логіки модель обробляє подію та надсилає результат роботи в представлення (view), тобто в інтерфейс користувача.

В Play ці три прошарки визначені в директорії проекту, кожен в окремому пакеті.

app/controllers. Контролер — це Java клас, де кожен статичний, відкритий метод є дією (action). Управління передається в такі методи, коли надходить відповідний запит. Код в контролерах нагадує швидше процедурно-орієнтоване програмування, ніж об'єктно-орієнтоване. Бо складається з набору методів, кожен з яких відповідає за обробку певного запиту. Ці методи витягують необхідні дані з HTTP запитів, працюють з об'єктами моделей та відправляють назад результати, які упаковуються в HTTP відповідь.

app/models. Прошарок моделей являє собою набір Java класів, який використовує всі об'єктно-орієнтовані особливості, доступні мові Java. Вони містять структуру даних та операції, на яких функціонує додаток. Кожного разу, коли об'єкти моделей повинні бути збережені в постійному сховищі, вони можуть включати деякі артефакти, такі як JPA анотації або SQL вирази.

app/views. Більшість представлень (або шаблонів) додатку генеруються ефективним шаблонним движком, який надається Play. Щоб відобразити необхідну інформацію контролер використовує шаблон. Цей пакет підтримує HTML, XML, JSON та інші формати шаблонів зі спеціальними директивами для динамічного створення моделі подання.

Результати порівняння Play framework з популярними технологіями розробки Інтернет-додатків такими як JSP (Java Server Pages) та Spring Framework наведено нижче. Кожна характеристика оцінювалась за п'ятибальною шкалою.

Порівняння Play Framework з альтернативними технологіями

	Java Server Pages	Spring Framework	Play Framework
Тестування	++	++++	+++++
	JSP не має вбудованих засобів тестування. Для цього використовуються сторонні програмні продукти. В Spring Framework є спеціальні бібліотеки для тестування, за основу яких взято JUnit, проте запускаються тести за допомогою програм збірки проектів Java (наприклад maven). Play Framework може запускати додаток в режимі виконання всіх тестів. За допомогою бібліотеки play.test створювати unit, функціональні та selenium тести набагато простіше.		
Сервер web додатків	+++	+++	++++
	Додатки Java Server Pages та Spring Framework розгортаються та функціонують на серверах web додатків Java від сторонніх виробників. Наприклад Tomcat, GlassFish, JBoss, Jetty, WebLogic і т.д. Play Framework містить вбудований web сервер. Також play додатки можуть бути розгорнуті на сторонніх серверах, як і Java Server Pages та Spring Framework додатки.		
Робота з базами даних.	+	++++	+++++
	Java Server Pages. Взаємодія з серверами баз даних є досить кропіткою роботою. Треба створювати джерело даних JDBC та пул підключень, пишучи при цьому програмний код. Spring Framework. Можна використовувати Spring JDBC, DAO та ORM. Spring забезпечує повне абстрагування питань підключення до баз даних та управління підключеннями. Також надає безліч допоміжних класів DAO, підтримує декілька технологій ORM, таких як Hibernate, JDO, iBatis. Play Framework має власний SQL сервер. В Play Framework визначені бібліотеки ORM JPA. За допомогою цих бібліотек можна реалізувати моделі баз даних та необхідну логіку роботи з ними, а тому немає потреби використовувати JDBC.		
Модульність додатків	++	+++++	+++++
	В додатку, побудованому суто на JSP та сервлетах-контролерах, напевно найважче використовувати допоміжні програмні модулі. З іншого боку існує велика кількість маленьких фреймворків, які входять до складу Spring Framework і можуть виконувати такі допоміжні функції як авторизація, аутентифікація, управління транзакціями, повідомленнями і т.д. Будь-яку складову частину цієї технології під час розробки можна замінити на іншу або створити власний аналог. Play Framework — модульно-орієнтовна технологія, розробниками якої вже створено досить багато готових модулів. Також досить просто написати власний модуль. Керування модулями в Play Framework дуже зручне.		

Висновки. В сучасному світі Internet технологій створено досить багато засобів розробки web-додатків, проте більшість з них є досить складними та незручними у використанні, що призводить до збільшення часу розробки та складності програмного коду. Play Framework — це молода технологія, проте вона вже набула популярності. Її розробники активно працюють над усуненням недоліків та випускають нові версії цієї технології.

Завдяки Play Framework процес створення web-додатків стає простим та швидким. Для функціонування play додатків не потрібне додаткове програмне забезпечення, на кшталт сервера web-додатків Tomcat і подібних, можна використовувати власні засоби даного фреймворку. Значно спрощена робота з базами даних завдяки підтримці JPA. Допоміжні модулі надають можливості для вирішення багатьох задач, з якими повсякчас зіштовхуються розробники. Модулі мають відкритий програмний код та по суті є звичайними play додатками, які встановлюються та налаштовуються досить просто.

ЛІТЕРАТУРА

1. James Ward. Tutorial: Play Framework, JPA, JSON, jQuery, & Heroku. — Режим доступу:
<http://www.jamesward.com/2011/12/11/tutorial-play-framework-jpa-json-jquery-heroku>
2. Play Framework. Manual, tutorials & references. — Режим доступу:
<http://www.playframework.org/documentation/1.2.4/home>
3. Isa Goksu. Writing Testable Code // "Methods & Tools" Software Development Magazine — Programming, Software Testing, Project Management, Jobs — Режим доступу:
<http://www.methodsand-tools.com/archive/archive.php?id=103>
4. Alexander Reelsen. Play Framework: Data Validation Using Controllers // Packt Publishing — Режим доступу:
<http://www.packtpub.com/article/play-framework-data-validation-controllers>
5. Ray Gomez. Develop using the Play! Framework: The JPA Model // The Buzz Media — 2009 — Режим доступу:
<http://www.thebuzzmedia.com/develop-using-the-play-framework-the-jpa-model/>
6. Groovy. User Guide. — Режим доступу:
<http://groovy.codehaus.org/User+Guide>

В.В. Герасимов, А.П. Пеньков, Г.О. Пушков

**АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ НАСИЧЕНИХ
ІНТЕРНЕТ-ДОДАТКІВ НА ПЛАТФОРМІ JAVA**

Анотація. Проведено аналіз технології JavaFX 2, яка активно розвивається в останній час. Наведено особливості цієї технології, розглянуто перспективи її розвитку. Представлені результати порівняння сучасних технологій розробки насичених Інтернет-додатків на різних платформах.

Ключові слова: RIA, JavaFX, Flash, Silverlight, FXML, Java, Groovy, JRuby, Scala, JVM, AJAX, інтерактивність, платформа, технологія, інтерфейс.

Вступ. З розвитком комп'ютерних технологій виділився окремий напрямок — розробка насичених (або багатих) Інтернет-додатків (Rich Internet Application, RIA). Це особливий клас додатків, який містить в собі симбіоз настільних (desktop) та веб-додатків. Він має кращі властивості цих зовсім різних напрямків: легкість в установці, безпека роботи, відсутність необхідності в підтримці програми, багаті можливості інтерфейсу, інтерактивність. Основним напрямком використання цієї технології є розробка клієнтської частини великих проєктів, яка зв'язується з сервером і виконує певні бізнес-функції. Також можна реалізовувати й інші окремі додатки — невеликі ігри, презентації та ін.



Рисунок 1 - Rich Internet Applications серед технологій створення програмних систем

Існує конкуренція між технологіями, за допомогою яких розробляються RIA-додатки. До останнього часу з боку Java-технології не було достойної відповіді для Flash від Adobe, Silverlight від Microsoft, однак з релізом JavaFX 2.0 ситуація змінилася і у Java-розробників з'явився конкурентоспроможний інструмент. Тепер всі частини проекту можуть бути повністю реалізовані на одній платформі Java, що забезпечує швидкодію, підвищує безпеку та скорочує час розробки, а звідси й ціну.

Метою роботи є дослідження технології JavaFX 2 та проведення порівняльного аналізу її з іншими технологіями розробки RIA-додатків.

Основна частина. Якщо звернутися до історії, то Java-аплети були найпершою RIA-технологією, яка стала доступною ще в 1995 р. Проте аплети не отримали широкого розповсюдження. Згодом з'явилась технологія JavaFX 1.x, проте й вона повторила історію з Java-аплетами. 10 жовтня 2011 року корпорація Oracle випустила JavaFX 2.0 — вдосконалену платформу для створення інтерфейсу користувача в бізнес-додатках корпоративного класу і наступного етапу еволюції Java в якості головної клієнтської повнофункціональної платформи.

На конференції JavaOne 2010 Oracle продемонстрували довгостроковий план для JavaFX (рис. 2). Початок схеми — грудень 2011 року.

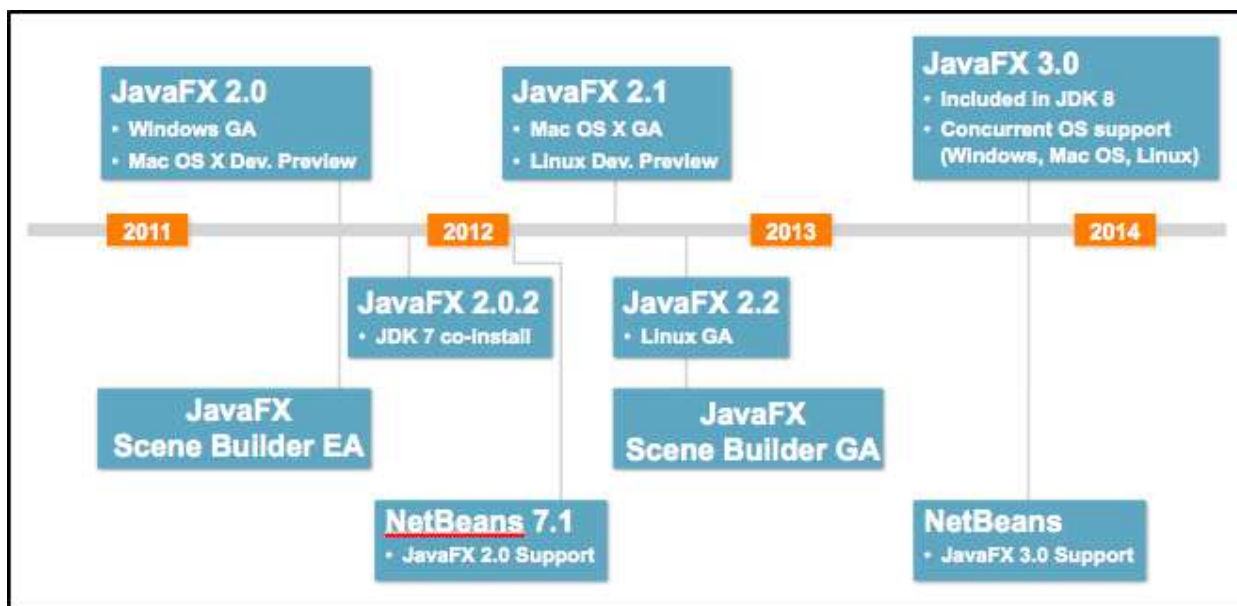


Рисунок 2 - Схема розвитку технології JavaFX

JavaFX 2.0 дає розробникам гнучкі можливості для створення додатків з використанням бажаних інструментів та мов програмування. Так в ній вводиться FXML — мова з підтримкою сценаріїв, основана на мові розмітки XML і призначена для опису користувацьких інтерфейсів. Розробники, знайомі з Web-технологіями або іншими мовами розмітки на базі XML, легко освоюють FXML і його можливості.

JavaFX 2.0 дозволяє застосовувати популярні мови сценаріїв, підтримувані віртуальною машиною Java (JVM), такі як Groovy, JRuby і Scala, та об'єднує простоту динамічних мов з функціональністю платформи Java. JavaFX 2.0 надає Web-компонент, який працює на базі движка web-рендерингу Webkit і дозволяє розробникам комбінувати і поєднувати традиційні можливості Java і динамічні можливості Web-технологій. При створенні додатків JavaFX розробники можуть використовувати бажані інструменти Java-розробки, наприклад, NetBeans і Eclipse. При проектуванні інтерфейсу користувачів розробники можуть використовувати спеціальну програму JavaFX Scene Builder.

Майбутні релізи JavaFX будуть основними етапами для розвитку JavaFX не тільки з точки зору функціональності, але й розглядаються як інтеграція в Java SE. Ось деякі з ключових нових функцій, які, як планується, будуть частиною майбутніх релізів JavaFX:

- Тісна інтеграція з Java SE;
- Покращення UI Controls та Charts;
- Підтримка сервісів даних;
- Покращення WebView;
- Модульність;
- Підтримка доступності;
- Краща підтримка 3D графіки;
- Multi-Touch та підтримка жестів;
- Підтримка датчиків.

Порівняємо технологію JavaFX з основними її конкурентами. По кожній властивості технології буде виставлятися кількість набраних балів («+») від 1 до 5. Властивості для порівняння наступні:

- Досвід користувачів — цей критерій показує рівень інтерактивності та швидкість відгуку: human-computer interaction (HCI). На основі цього будується інтерактивна модель додатку;

- Розгортання — описує рівень комфортності при розгортанні додатку (створення релізу, інсталяція, активація, оновлення, адаптація, деінсталяція та ін.);
- Забезпечення широкої доступності — критерій описує такі речі як кількість користувачів, поширеність та доступність;
- Багатотехнологічність — критерій взаємодії або її відсутності з іншими технологіями;
- Час запуску — час, необхідний для старту роботи додатка;
- Клієнт – Сервер — порівнюються процеси на сервері або клієнті;
- Платформонезалежність;
- Підтримка розробників — цей критерій стосується інструментів розробки та мов програмування;
- SEO, доступність — можливість автоматизованого індексування та доступність для людей-інвалідів;
- Роздільна здатність екрану — критерій незалежності від розділювальної здатності екрану;
- Насиченість UI — використання стандартних компонентів, кастомізація, скіни, стилі, анімація та ін.;
- Вартість — критерій вартості розробки, ліцензії, вартість або безкоштовність використання технології.

Таблиця 1

Порівняння технологій

Властивості	AJAX	Flash	Silverlight	JavaFX
Досвід користувачів	++	++++	+++	+++ +
	AJAX, відносно досвіду користувачів, революційно замінив звичайний DHTML, однак він є дуже залежним від парадигми браузера. Silverlight порівняно з іншими підтримує меншу кількість браузерів. Flash ще не досконало взаємодіє з Adobe Integrated Runtime (AIR) для переносу веб-додатків у десктоп-додатки. Під час запуску Java-додатків дещо погіршується швидкість відгуку додатку.			
Розгортання	++++	++++	+++	+++
	Центральне розгортання серверу пропонують всі технології. AJAX не потрібно оновлювати, тому що він поставляється з браузерами. Тому він має найменший бар'єр для використання користувачами веб-додатку. Flash і JavaFX пропонують автоматичний оновлювач. Але Java при онов-			

	ленні перериває на довший час потік робіт (workflow) користувача.			
Забезпечення широкої доступності	+++++	++++	+++	+++ +
	AJAX вбудований в усі браузери. Flash є доступним у 97 % браузерів, іноді інсталюється з операційною системою. Java також в більшості випадках інсталюється. Silverlight менш поширений.			
Багатотехнологічність	+	+++	++++	+++ +
	AJAX не вистачає взаємодії між різними стандартами та платформами і різними мовами програмування. Flash, JavaFX та Silverlight — поодинокі технології, які пропонують єдине середовище виконання, яке забезпечує однакову поведінку на багатьох платформах. Silverlight дозволяє обирати мову для реалізації. В JavaFX також це доступно через використання таких мов JVM, як Groovy і Scala.			
Час запуску	+++++	+++++	++++	+++
	Головною рисою AJAX є те, що він одразу підтримується браузерами і запуск додатків на ньому миттєвий. Flash також дуже швидкий. Silverlight трохи повільніший. Час запуску Java потребує декількох кроків і тому він більший за інших в цьому списку.			
Клієнт – Сервер	++	++++	+++	+++ ++
	AJAX пропонує асинхронну комунікацію і зменшення навантаження на сервер, але не має сокет-з'єднання. Flash має сокет-з'єднання з AIR. Silverlight та Flash забезпечують швидку взаємодію з користувачем, але ця технологія націлена на виконання на боці клієнта. Java технологія добре підходить для серверної сторони та пропонує однорідну комунікацію між сервером та RIA-клієнтом.			
Платформонезалежність	+++++	++++	+++	+++ +
	Серед всіх Silverlight підтримує меншу кількість платформ, Flash та JavaFX є кращими за нього, але в той же час AJAX має ширші можливості майже на усіх мобільних пристроях.			
Підтримка розробників	+++	++++	++++	+++ ++
	Для AJAX розробки існують допоміжні інструменти, але їх не багато і вони не є кросплатформними. Допоміжні інструменти розробки Silverlight та Flash є платними, а інструменти розробки JavaFX є безкоштовними.			
SEO, доступність	+++	+	+	+

	AJAX пропонує найкращі показники SEO та доступності. JavaFX, Silverlight та Flash потребують використання HTML-альтернатив.			
Роздільна здатність екрану	+	++++	++++	+++ +
	AJAX може використовувати тільки браузерну піксельну технологію. JavaFX, Silverlight та Flash використовують векторну графіку.			
Насиченість UI	+	+++++	++++	+++ +
	AJAX впроваджує стилювання за допомогою CSS (каскадні таблиці стилів), але не має детального стилювання компонент та недостатньо реалізує концепцію «Look and Feel» (вигляд і поведінку). Flash порівняно з JavaFX та Silverlight в цьому плані пропонує більше.			
Вартість	+++	++	++	+++ +
	Для деяких браузерів потрібна інша реалізація додатку на AJAX і це потребує додаткових витрат. Flash та Silverlight не є безкоштовними технологіями, як і засоби для їх розробки. JavaFX та інструменти для розробки таких додатків є безкоштовними, з відкритим кодом.			
Результат	35	44	38	45

Таким чином, не дивлячись на відносно невелику швидкість запуску додатків, написаних за допомогою JavaFX, ця технологія одержує найбільшу кількість балів завдяки безкоштовності технології та інструментів розробки, великої кількості Java-спільноти та підтримки різних мов програмування. Silverlight не отримав широкої популярності і через проблеми цю технологію більше не будуть розвивати. Не дивлячись на невелику кількість набраних балів AJAX є дуже популярним, але його більш використовують для підвищення інтерактивності та комфортності при роботі з веб-ресурсами (соціальні мережі, сайти, інтернет-магазини), ніж для розробки RIA додатків. Більшість RIA-додатків розроблені на Flash: це склалося історично — інші альтернативні технології не могли на той час бути конкурентоспроможними, але на даний момент ситуація змінюється, популярність Flash падає, технологія JavaFX знаходить своїх прихильників і кількість розроблених додатків за допомогою цієї технології нестримно росте.

Висновки. Корпорація Oracle зробила великий крок вперед в напрямку RIA, що є дуже доброю новиною для Java-розробників. Технологія має багато різних реалізованих елементів інтерфейсу та

класів, які дуже прискорюють розробку додатків. Цю технологію також можуть використовувати розробники Ruby, Scala та інших мов, основаних на JVM. Великою перевагою JavaFX є те, що додаток, написаний за допомогою цієї технології, може мати декілька режимів розгортання і працювати на різних типах пристроїв, що робить цей додаток дуже універсальним і при цьому заради цієї універсальності не потрібно нічого робити додатково.

Однак JavaFX також має свої недоліки. Головним серед них є відносно великий час запуску додатку. Також через те, що версія 2.x кардинально відрізняється від версії 1.3, JavaFX 2.x можна розглядати як тільки-но створену технологію. І тому поки що дуже багато заплановано, але реалізовано не так багато, інструменти графічної розробки не дуже комфортні, мало досвідчених розробників та відносно мало інформації — тобто наявні всі риси, присутні новим технологіям. На даний момент доступною є версія JavaFX 2.2.3, в якій виправлена велика кількість недоліків та помилок, і розробники цієї технології можуть перейти до наступного кроку — впровадження нових ідей та можливостей до наступних версій JavaFX.

ЛІТЕРАТУРА

1. Florian Moritz. Rich Internet Applications (RIA): A Convergence of User Interface Paradigms of Web and Desktop Exemplified by JavaFX — Diploma Thesis / University of Applied Science Kaiserslautern — Saarbruecken, 2008 — P. 140.
2. JavaFX Documentation Home. Getting Started with JavaFX — Режим доступу:
http://docs.oracle.com/javafx/2.0/get_started/jfxpub-get_started.htm
3. JavaFX Documentation Home. JavaFX Architecture and Framework — Режим доступу:
<http://docs.oracle.com/javafx/2.0/architecture/jfxpubarchitecture.htm>
4. JavaFX Documentation Home. JavaFX Overview — Режим доступу:
<http://docs.oracle.com/javafx/2.0/overview/jfxpub-overview.htm>
5. Алексеев А. А., Савельев А. О. Технологии Microsoft для создания RIA-приложений — Режим доступу:
<http://www.intu-it.ru/departament/se/tmsria/>

В.М. Григорьев

ВЫБОР ПЛАТФОРМЫ ДЛЯ РЕАЛИЗАЦИИ ЛАБОРАТОРНОГО ПРАКТИКУМА ПО КОМПЬЮТЕРНЫМ СЕТЯМ

Аннотация. Исследуется время загрузки большого числа виртуальных машин Qemu под управлением операционной системы RouterOS фирмы Mikrotik. в рамках хост-машины Ubuntu. Показано, что операционная система RouterOS является адекватным выбором для организации виртуальной лаборатории по компьютерным сетям.

Ключевые слова: время загрузки, виртуальная машина, RouterOS, Mikrotik, Qemu, GNS3, Ubuntu .

Актуальность темы. Большинство сетевых устройства (например, маршрутизаторы) работают под управлением операционных систем. В ряде случаев эти операционные системы можно запустить внутри виртуальных машин. Сетевые устройства в сети соединены каналами связи. Для организации виртуальной сетевой лаборатории надо соединить между собой виртуальные машины, в которых запущены операционные системы сетевых устройств.

Виртуальная лаборатория применима как испытательный стенд на начальной стадии проектирования реальных сетей, так и для обучения сетевым технологиям и подготовке к сертификации.

При комплектации реальной сетевой учебной лаборатории возникают трудности при определении производителя сетевого оборудования, подборе и обновлении сетевых устройств. Неприемлемы сетевые устройства, обладающие ограниченной функциональностью и Веб-интерфейсом для настройки. В виртуальной лаборатории подбор и смена «оборудования» не приводит к закупке новых сетевых приборов, а сводится к установке новых виртуальных машин.

Поместив лабораторию на общедоступном сетевом ресурсе, с помощью удалённого рабочего стола можно обеспечить совместную работу над сетевым проектом. Для коллективной работы не требуется

организация одновременного физического присутствия людей в одном месте. Появляется возможность давать студентам домашние задания по проектированию сетей, что в случае реального оборудования невозможно.

Виртуальная лаборатория обладает следующими преимуществами по сравнению с реальной:

- отсутствие затрат на оборудование,
- нет необходимости в помещении для развёртывания сетевого оборудования,
- высокая скорость сбора и коммутации сетевой топологии,
- каналы связи между сетевыми устройствами в виртуальной топологии имеют 100% надёжность и заявленную пропускную способность.

Анализ последних исследований Идея виртуальных сетевых лабораторий не нова, и для их реализации существуют специализированные решения. Для моделирования устройств фирмы Cisco, работающих под управлением операционной системы IOS, используются программы Cisco Packet Tracer и Boson NetSim, имеющие удобный графический интерфейс, позволяющий быстро создавать достаточно сложные сетевые топологии. Однако встроенная в них урезанная версия IOS позволяет изучать лишь сетевые технологии начального уровня

Для лабораторий Netsim фирмы Bosson и PacketTracer фирмы Cisco разработаны лабораторные практикумы для подготовки к сертификации CCNA и CCNP.

Большой интерес представляет использование операционных систем реальных сетевых устройств. Сетевая общественность для подготовки к сертификациям Cisco широко использует менеджер виртуальных машин Dynamips, поддерживающий широкий спектр операционных систем IOS. Виртуальная машина под управлением Dynamips запускается из командной строки и требует к себе бережного отношения. Для предотвращения 100%-й утилизации процессора необходимо подбирать параметр idlePC. Без должных настроек Dynamips использует все ресурсы компьютера уже для топологии из трёх маршрутизаторов.

Dynamips входит в состав контейнера виртуальных машин GNS3. GNS3 – графическая оболочка, позволяющая работать с визуальным представлением сетевой топологии в режиме Drag-and-Drop.

В связке Dynamips-GNS3 сетевая топология хранится в текстовом файле, а конфигурации сетевых устройств присутствуют в файлах в виде команд IOS. Настроенная в одном GNS3 топология легко переносима в другой GNS3.

Есть ряд платных курсов по сертификациям CCNP и CCIE, которые базируются на Dynamips и GNS3:

1. CCIE Service Provider Workbook фирмы ieMentor;
2. CCIE Lab Workbook Internetworkexpert (ine.com) .

Эти курсы снабжены файлами конфигураций и топологий для Dynamips и GNS3.

Есть огромные репозитории, хранилища топологий и конфигураций под Dynamips и GNS3.

<http://www.gns3-labs.com>

<http://gns3vault.com>

<http://7200emu.hacki.at>

gns3 workbench на <http://sf.net>

Для практически любой сетевой технологии можно найти реализацию в Dynamips и GNS3. Всё это положительно влияет на продвижение продуктов Cisco.

Казалось бы связка Dynamips и GNS3 – идеальный выбор для организации учебной лаборатории. Однако, опыт показал, что рядовой студент не относится к настройке параметра IdlePC в Dynamips с должной ответственностью и даже небольшая топология занимает всё процессорное время.

Нельзя не упомянуть менеджер виртуальных машин IOU для операционных систем Cisco IOS. В ней можно запустить пару операционных систем Cisco IOS с весьма мощной функциональностью, и она не требует такой настройки, как Dynamips. К сожалению, IOU не обладает графическим интерфейсом.

Для создания сетевых топологий в GNS3 используется технология Drag-and-Drop: зацепил устройство мышью и перетащил его на рабочее поле. GNS3 поддерживает три менеджера виртуальные машины: Dynamips, VirtualBox и Qemu. Выбор именно этих менеджеров машин для включения в GNS3 обусловлен наличием в их составе раз-

витых средств для соединения между собой виртуальных машин (в VirtualBox - с помощью API).

Под Qemu работает весьма широкий класс сетевых, встроенных и мобильных операционных систем: Juniper JunOS (FreeBSD), файерволы Cisco, множество встроенных Linux для сетевых и мобильных устройств (Vyatta, *WRT, Google Android, Mikrotik RouterOs и др).

Под Qemu можно запустить виртуальные машины не только для Intel-платформы.

VirtualBox также поддерживает множество ОС, но в составе GNS3 требует настройки отдельной виртуальной машины для каждого устройства сетевой топологии. Qemu для всех устройств с одинаковой ОС может использовать единые настройки.

Наш выбор при организации виртуальной лаборатории был сделан в пользу менеджера виртуальных машин Qemu в составе GNS3. Связка Qemu - GNS3 позволяет быстро создать сложную сетевую топологию из десятков устройств, соединив устройства топологии друг с другом.

Постановка задачи. Предполагается, что в виртуальной лаборатории по компьютерным сетям присутствует десяток пользователей, и каждый из них работает с сетевой топологией, состоящей из нескольких сетевых устройств. Одновременно работают десятки виртуальных машин. Возникает задача такого выбора операционных систем хост-машины и сетевых устройств, чтобы обеспечить пользователям комфортную одновременную работу при условии ограниченности ресурсов хост-компьютера. Удобство работы определяется временем запуска и скоростью работы операционной системы внутри виртуальной машины, что в свою очередь зависит от того, сколько ресурсов хост-машины потребляет виртуальная машина.

Следовало определиться, какую операционную систему использовать для хост-машины: Windows или Linux. GNS3 и Qemu задуманы, сделаны и развиваются в Linux. Qemu под Linux поддерживает аппаратную виртуализацию KVM (Kernel-based Virtual Machine) на базе процессоров Intel VT либо AMD SVM. Сам по себе KVM не выполняет эмуляции и используется совместно с виртуальными машинами Qemu. Мы будем использовать KVM без оптимизатора памяти ksm.

Qemu под Windows не поддерживает KVM и при запуске нескольких экземпляров Qemu используется только одно ядро центрального процессора, что существенно замедляет работу с большими сетевыми топологиями. Разработчики Qemu винят в этом планировщик процессов Windows.

Возникает вопрос выбора дистрибутива Linux. GNS3 написан на Python и требует библиотеки Qt4. После ряда экспериментов с различными дистрибутивами Linux по установке GNS3 из исходных кодов выбор пал на настольную версию Ubuntu.

Обоснование полученных результатов. Определим операционную систему сетевого устройства для запуска под Qemu. Если потребовать, чтобы устройство поддерживало сетевую технологию MPLS, то выбор сразу сократится: это либо операционная система JunOS фирмы Juniper, либо RouterOS фирмы Mikrotik.

По объёму потребляемых ресурсов JunOS существенно превосходит RouterOS. Например, на компьютере с двухядерным процессором Intel Core2 6600 с частотой 2.4 ГГц время загрузки RouterOS версии 5.21 в Qemu под Ubuntu составляет несколько секунд (см. ниже), а JunOS версии Olive12.1R1.9 грузится 60-80 секунд. RouterOS требует минимум 64 Мб памяти, JunOS — 512 Мб. Образ диска RouterOS - 60 Мб, JunOS - 600 Мб.

Рассмотрим влияние KVM на время загрузки в Qemu (версия 13.0) в GNS3 нескольких экземпляров RouterOS версии 5.21 с памятью 64 Мб на компьютере с двухядерным процессором 2.4 ГГц и 4Гб памяти под управлением Ubuntu (табл. 1). Графики зависимости времени загрузки от числа экземпляров RouterOS приведены на рис.1. RouterOS считается загруженной, если она выдала в консоли слово «MikroTik». Это проверяется с помощью следующего скрипта на языке Python

```
tn = telnetlib.Telnet()
for p in range(3000,3079):
    while 1:
        try:
            tn.open("127.0.0.1",p)
            tn.write("\r")
            tn.read_until("MikroTik")
            break
        except socket.error:
```

```
print p
time.sleep(1)
continue
```

Из табл.1 видно, что KVM уменьшает время загрузки RouterOS более, чем в два с половиной раза и это время составляет около двух секунд. Для обычного бытового компьютера имеем приемлемое время загрузки 80-ти экземпляров RouterOS равное 3 минутам.

Загрузим одновременно 32 экземпляра RouterOS (62с). Не останавливая загруженные RouterOS, запустим в новом GNS3 одновременно ещё 32 экземпляра RouterOS. На это уйдёт 75с. Итого 137с., что сравнимо с временем одновременного старта 64-х RouterOS (140с., согласно, табл.1).

Таблица 1

Время загрузки нескольких экземпляров RouterOS в GNS3 в Ubuntu

Кол-во эк- земпляров RouterOS N	Qemu без KVM		Qemu с KVM	
	Общее время загрузки Тсек.	Время загрузки одного экземп- ляра N/Т сек.	Общее вре- мя загрузки Т сек.	Время загруз- ки одного эк- земпляра N/Т сек.
16	82	5.125	30	1.875
32	166	5.1875	62	1.9375
48	260	5.416667	100	2.083333
64	470	7.34375	140	2.1875
80	Нет памяти		180	2.25

Число одновременно работающих экземпляров RouterOs под Qemu определяется свободной памятью хост-машины Ubuntu. Анализ показал, что каждый экземпляр RouterOS с памятью 64 Мб, запущенный в Qemu с KVM, требует у Ubuntu 32 Мб памяти, а каждый экземпляр RouterOS с памятью 128 Мб запущенный в Qemu с KVM, требует у Ubuntu 54 Мб памяти.

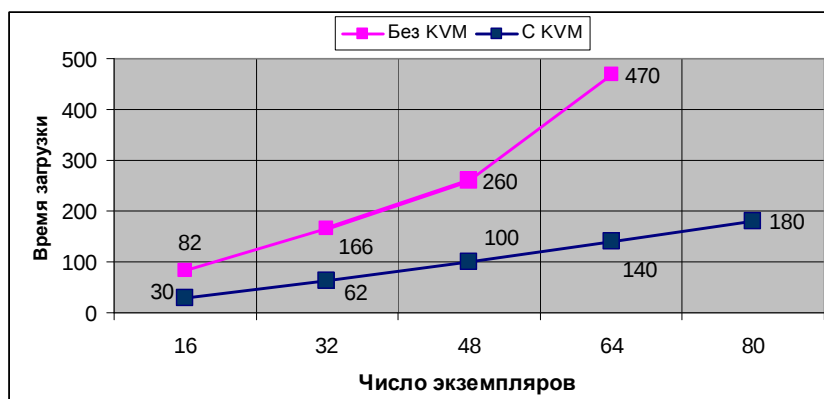


Рисунок 1 - Время загрузки нескольких экземпляров RouterOS в GNS3 в Ubuntu (2 ядра)

Операционную систему Ubuntu можно запускать также под управлением виртуальной машины, например HYPER-V фирмы Microsoft. Qemu в Ubuntu под управлением HYPER-V работает несколько медленнее, чем Qemu в Ubuntu на реальном компьютере. Это обусловлено, помимо прочего, и тем, что KVM не работает на виртуальной аппаратуре HYPER-V.

Хост-компьютер для HYPER-V имеет 4-ядерный процессор Intel Core i7 950 частотой 3066 МГц и 24Гб памяти. В HYPER-V запущена система Ubuntu с 24Гб памяти и 4-мя виртуальными ядрами. Рассмотрим время загрузки в Qemu в GNS3 нескольких экземпляров RouterOS версии 5.21 с памятью 64 Мб на виртуальной Ubuntu (табл.2). График зависимости времени загрузки от числа экземпляров RouterOS приведен на рис.2.

Таблица 2

Время загрузки нескольких экземпляров RouterOS в Ubuntu под HYPER-V

Кол-во экземпляров RouterOS N	Общее время загрузки T сек.	Время загрузки одного экземпляра N/T сек.
16	39	2.4375
32	75	2.34375
48	118	2.458333
64	155	2.421875
80	195	2.4375

Из табл.2 видим, что время загрузки RouterOS под Qemu в Ubuntu под HYPER-V составляет около двух с половиной секунд, что

сравнимо по времени с бытовым компьютером. Для виртуального компьютера имеем приемлемое время загрузки 80-ти экземпляров RouterOS равное 195 секундам.

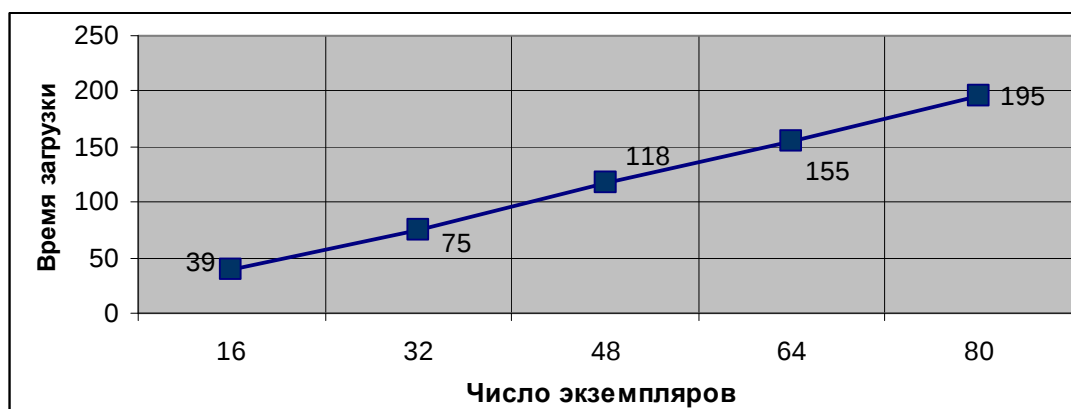


Рисунок 2- Время загрузки нескольких экземпляров RouterOS в GNS3 в Ubuntu под HYPER-V (4 ядра)

Анализ показал, что каждый экземпляр RouterOS с памятью 64 Мб, запущенный в Qemu, требует у Ubuntu под HYPER-V 57 Мб памяти, а каждый экземпляр RouterOS с памятью 128 Мб запущенный в Qemu, требует у Ubuntu HYPER-V 79 Мб памяти. То есть RouterOS в Qemu с KVM потребляет приблизительно вдвое меньше памяти, чем RouterOS в Qemu без KVM.

Время загрузки одного экземпляра RouterOS на бытовом 2-х ядерном компьютере без поддержки KVM в два раза больше, чем на виртуальном 4-х ядерном процессоре HYPER-V (см. табл.). Много-ядерность процессоров распараллеливает загрузку нескольких экземпляров RouterOs. Поэтому реальное ядро бытового компьютера имеют приблизительно одинаковую мощность по сравнению с ядром виртуального процессора HYPER-V.

Рассмотрим влияние KVM на время загрузки в Qemu версии 13.0 в GNS3 нескольких экземпляров RouterOS версии 5.20 с памятью 64 Мб на более мощном компьютере с четырёхядерным процессором Intel i5-3550 под управлением Ubuntu (табл.3). Графики зависимости времени загрузки от числа экземпляров RouterOS приведены на рис.3.

Время загрузки нескольких экземпляров RouterOS в GNS3 в Ubuntu
для компьютера с 4-х ядерным Intel i5-3550

Кол-во экземпляров RouterOS N	Qemu без KVM		Qemu с KVM	
	Общее время загрузки T сек.	Время загрузки одного экземпляра N/T сек.	Общее время загрузки T сек.	Время загрузки одного экземпляра N/T сек.
16	35	2.1875	23	1.4375
32	60	1.875	40	1.25
48	90	1.875	58	1.208333
64	120	1.875	73	1.140625
80	152	1.9	90	1.125

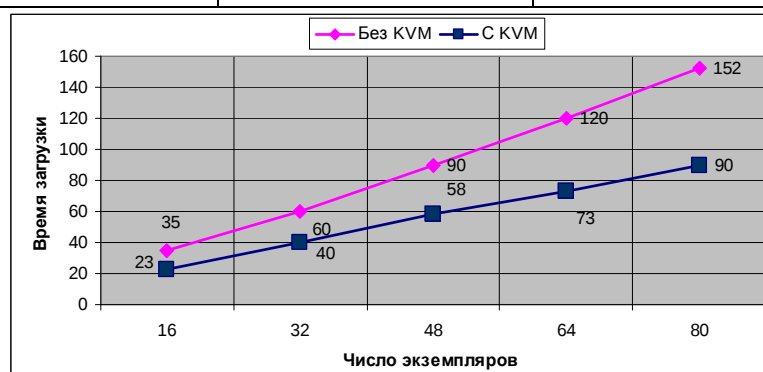


Рисунок 3 - Время загрузки нескольких экземпляров RouterOS в GNS3 в Ubuntu с 4-х ядерным Intel i5-3550

Из табл. 3 видим, что KVM уменьшает время загрузки RouterOS в полтора раза и это время составляет менее двух секунд. Время загрузки 80-ти экземпляров RouterOS с поддержкой KVM составляет полторы минуты.

Выводы. Операционная система RouterOs поддерживает практически все современные сетевые технологии: виртуальные частные сети 2-го и 3-го уровня, IPSec, MPLS, VPLS, VRF. С учётом функциональности, быстродействия и потребляемым ресурсам Mikrotik RouterOs под управлением виртуальной машины Qemu в составе GNS3 под управлением Ubuntu является лучшим выбором для организации виртуальной сетевой лаборатории.

ЛИТЕРАТУРА

1. Григорьев В.М. Виртуализация лабораторного практикума по компьютерным сетям // Системные технологии. Региональный межвузовский сборник научных трудов. - Выпуск 1 (78). - Днепропетровск, 2012. - С. 104–112.

ПРОИЗВОДИТЕЛЬНОСТЬ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА В РЕЖИМЕ ТРОТТЛИНГА

Аннотация. Выполнен анализ производительности процессора в режиме троттлинга при его аппаратной реализации, с помощью критических терморезисторов на основе VO_2 . Показано, что если средняя затрата энергии на одну машинную команду E_{in} не превышает минимальную энергию $E_{in\ min}$, вызывающую режим троттлинга, процессор работает с производительностью, отвечающей нормальному тепловому режиму. При $E_{in} \geq E_{in\ min}$ производительность процессора уменьшается обратно пропорционально значению E_{in} .

Ключевые слова: производительность процессора, режим троттлинга, защита процессора от перегрева.

Введение. Для современной вычислительной техники характерна тенденция роста степени интеграции микросхем и увеличения количества команд, выполняемых за один такт. В связи с этим актуальной проблемой является обеспечение нормального теплового режима процессора [1, 2]. При выполнении блоков программы, требующих энергоемких аппаратных ресурсов, возникает перегрев, следствием которого может стать выход из строя СБИС процессора. Эффективным средством борьбы с перегревом, в таком случае, является перевод процессора в режим троттлинга [1]. В этом режиме уменьшается количество „полезных” тактов, в которых процессор выполняет машинные команды и увеличивается число „холостых” тактов, в которых он простаивает. Как следствие энергопотребление снижается и процессор остывает.

Простой способ аппаратной реализации режима троттлинга для защиты процессора от перегрева предложен в работах [3,4]. Этот способ использует критический терморезистор на основе пленки диоксида ванадия, который в области температуры $T_t = 68^\circ\text{C}$ скачком изменяет электрическое сопротивление не менее чем в 100 раз. Такой терморезистор создается непосредственно на кристалле процессора, и

переключается в состояние с низким сопротивлением, когда температура процессорного чипа достигнет 68 °С.

При реализации режима троттлинга возникает вопрос о том, насколько будет снижена производительность процессора по сравнению с нормальным тепловым режимом его работы.

Постановка задачи. Задачей настоящей работы является анализ производительности процессора в режиме троттлинга при его аппаратной реализации, использующей критический терморезистор на основе VO_2 .

Основная часть. При анализе производительности процессора будем исходить из тактовой частоты f и количества машинных команд n , выполняемых за один такт. Тогда, в нормальном тепловом режиме, производительность равна:

$$PT_0 = nf. \quad (1)$$

Если при обработке блока программы средняя энергия, затрачиваемая на выполнение одной машинной команды E_{in} , то в процессоре выделяется мощность nfE_{in} . Для упрощения анализа примем, что температура T_{CU} одинакова во всех точках процессорного чипа, а температура окружающей среды равна Q . Тогда уравнением теплового баланса процессора имеет вид:

$$C_{CU} \frac{d\Delta T}{dt} + K_{CU} \Delta T = nfE_{in}, \quad (2)$$

где C_{CU} – теплоемкость процессорного чипа; t – время; $\Delta T = T_{CU} - Q$; K_{CU} – коэффициент теплообмена процессора с окружающей средой.

Решение (2) при начальном условии $t = 0$, $\Delta T = 0$ имеет вид:

$$T_{CU} = Q + \frac{nfE_{in}}{K_{CU}} \left(1 - \exp\left(-\frac{t}{\tau_{CU}}\right) \right), \quad (3)$$

где $\tau_{CU} = C_{CU}/K_{CU}$ – тепловая постоянная времени процессора.

Как вытекает из (3), при отсутствии энергозатрат на выполнение машинных команд ($E_{in} = 0$), температура процессора не изменяется во времени и равна температуре окружающей среды Q . Если $E_{in} \neq 0$, то при выполнении блока команд, температура процессора возрастает, стремясь к постоянному значению T_{CUmax} . Это значение, согласно (3) равно:

$$T_{CUmax} = Q + \frac{nfE_{in}}{K_{CU}}, \quad (4)$$

При аппаратной реализации режима троттлинга, использующей терморезистор с критической температурой T_t [3, 4], возможны два случая.

В первом случае максимальная температура нагрева процессора T_{CUmax} не превышает значение T_t . Поэтому режим модуляции тактовой частоты f не включается и процессор работает с производительностью PT_0 , определяемой (1).

Во втором случае $T_{CUmax} \geq T_t$ критический терморезистор, переключается в состояние с низким сопротивлением и запускает режим модуляции частоты. Процессор переходит в режим троттлинга. В этом режиме температура T_{CU} испытывает колебания в интервале $T_0 \leq T \leq T_t$ (рисунок 1), что исключает перегрев процессора. Для терморезисторов на основе диоксида ванадия VO_2 значение $T_t = 68^\circ C$ ниже предельной температуры перегрева процессора $90^\circ C$ [2], что обеспечивает его надежную защиту при выполнении энергоемких участков программы.

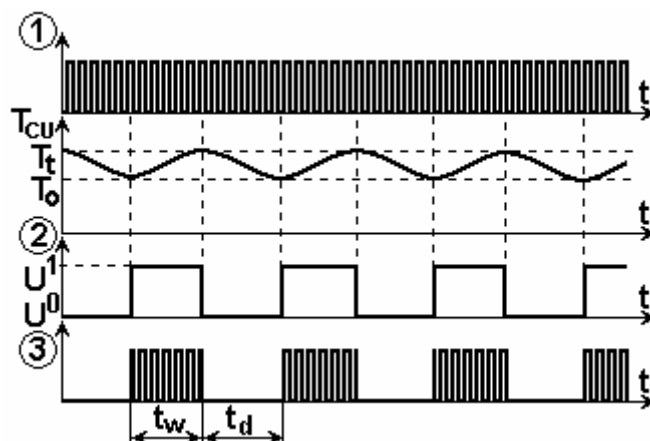


Рисунок 1 – Временные диаграммы для режима троттлинга.

1 – тактовые импульсы; T_{CU} – температура процессора,

2 – модулирующий сигнал;

3 – тактовые импульсы после модуляции

Приравняв в (4) температуру T_{CUmax} значению критической температуры терморезистора T_t , можно найти энергию $E_{in\ min}$. Это минимальная энергия, затрачиваемая на выполнение одной машинной команды, при которой процессор переходит в режим троттлинга:

$$E_{in\ min} = \frac{K_{CU}}{nf} (T_t - Q) = \frac{K_{CU}}{PT_0} (T_t - Q). \quad (5)$$

Процессор работает с производительностью $PT_0 = nf$, если средняя энергия E_{in} , на одну машинную команду удовлетворяет усло-

вию $E_{in} < E_{in \min}$. При условии $E_{in} \geq E_{in \min}$ электронный узел, схема которого представлена в [3, 4], переводит процессор в режим троттлинга. Для этого режима характерна последовательность “холостых” тактов длительностью t_d (рисунок 1), когда процессор не выполняет команды программы. Энергопотребление снижается и процессор остывает до температуры T_o , при которой терморезистор переключается в состояние с высоким сопротивлением. Подача тактовых импульсов возобновляется. Процессор выполняет программу, пока не нагреется до температуры T_t , при которой вновь переходит в режим холостых тактов. Описанные процессы периодически повторяются до завершения выполнения блока команд с $E_{in} \geq E_{in \min}$. Поскольку в режиме троттлинга не все такты используются для выполнения команд, производительность процессора $PT < PT_0 = nf$.

Таким образом, в интервале времени t_w процессор выполняет команды программы, а в интервале t_d простаивает (рисунок 1). Кинетика температуры процессора $T_{CU}(t)$ в интервале времени t_w описывается уравнением (2) с начальным условием $t = 0$, $T_{CU}(0) = T_o$, а в интервале t_d , когда $E_{in} = 0$ – уравнением

$$\frac{d\Delta T}{dt} + \frac{\Delta T}{\tau_{CU}} = 0, \quad (6)$$

с начальным условием $t = 0$, $T_{CU}(0) = T_t$.

За время t_w процессор выполнит $t_w fn$ команд. Поэтому его производительность PT в режиме троттлинга описывается соотношением:

$$PT = \frac{t_w fn}{t_w + t_d} = \frac{PT_0}{1 + t_d/t_w}. \quad (7)$$

Решая уравнения (2) и (6) при указанных выше начальных условиях получим уравнения для интервалов времени t_d и t_w :

$$T_o = Q + (T_t - Q) \exp\left(-\frac{t_d}{\tau_{CU}}\right), \quad (8)$$

$$T_t - Q - \frac{nfE_{in}}{K_{CU}} = (T_o - Q - \frac{nfE_{in}}{K_{CU}}) \exp\left(-\frac{t_w}{\tau_{CU}}\right). \quad (9)$$

Из (8) и (9) вытекает следующая связь между значениями t_d и t_w :

$$t_w = \tau_{CU} \ln \left(\frac{\frac{nfE_{in}}{K_{CU}} - (T_t - Q) \exp\left(-\frac{t_d}{\tau_{CU}}\right)}{\frac{nfE_{in}}{K_{CU}} - T_t + Q} \right). \quad (10)$$

Как показано в [4], длительность “холостых” тактов t_d много меньше тепловой постоянной времени процессора τ_{CU} ($t_d \ll \tau_{CU}$). Это позволяет в (10), разложить в ряд Тейлора экспоненциальную и логарифмическую функции, ограничившись двумя членами разложения. В результате, с учетом (5) получим выражение для отношения длительностей “холостых” t_d и рабочих тактов t_w :

$$\frac{t_d}{t_w} = \frac{E_{in}}{E_{in \min}} - 1. \quad (11)$$

Подстановка (11) в (7) позволяет найти зависимость производительности процессора в режиме троттлинга от E_{in} . Обобщая результат для $E_{in} < E_{in \min}$, получим следующее выражение для производительности процессора:

$$PT = \begin{cases} PT_0 = fn, & \text{если } E_{in} < E_{in \min} \\ PT_0 \frac{E_{in \min}}{E_{in}}, & \text{если } E_{in} \geq E_{in \min} \end{cases}. \quad (12)$$

График зависимости (12) показан на рисунке 2.

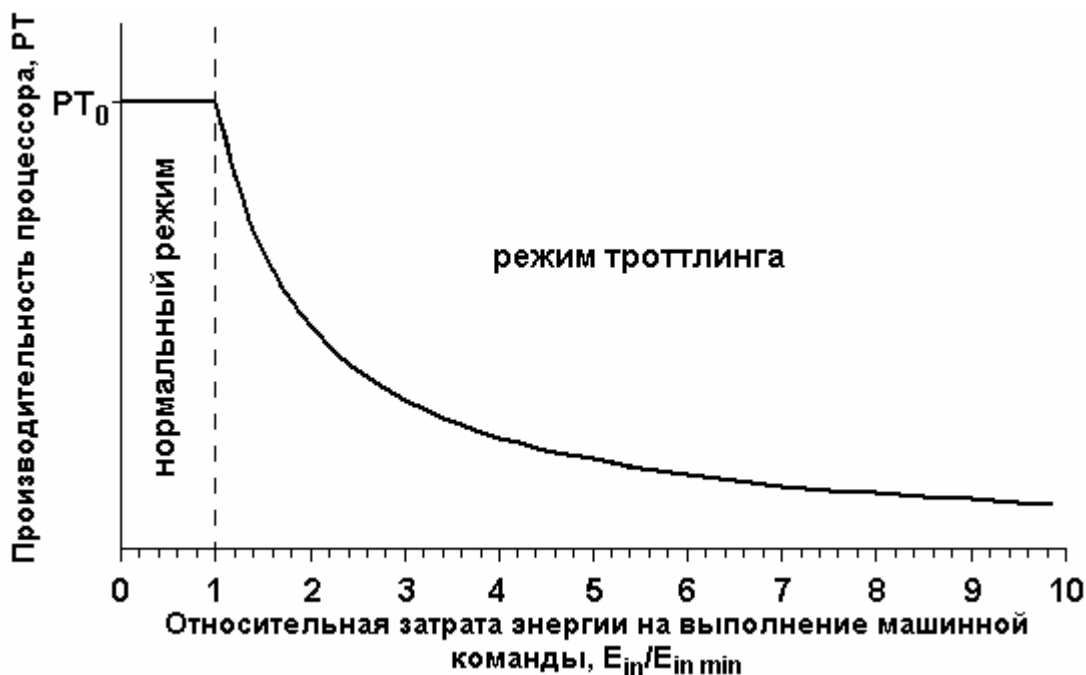


Рисунок 2 – Зависимость производительности процессора от относительной затраты энергии на одну машинную команду

Выводы. Нагрев процессора не превышает 68 °С и его производительность отвечает нормальному тепловому режиму, если средняя энергия, затрачиваемая на выполнение одной машинной команды E_{in} , меньше значения $E_{in \min}$, определяемого соотношением (5). При $E_{in} \geq E_{in \min}$ критический терморезистор на основе VO_2 , переводит процессор в режим троттлинга. В этом режиме производительность процессора уменьшается обратно-пропорционально значению E_{in} .

ЛИТЕРАТУРА

1. Функции управления производительностью и энергопотреблением процессоров Intel Pentium 4 и Intel Xeon. Часть 2: новые процессоры, новые технологии [Электронный ресурс] / Д. Беседин // iXBT – 2004 – Режим доступа: <http://www.dvgu.ru/meteo/PC/IntelThermal-2.html>. – Загл. с экрана.
2. Защита процессоров AMD от перегрева (Обзор) [Электронный ресурс] / А. Карабуто // FERRA.RU – 2002 – Режим доступа: <http://www.ferra.ru/online/system/s19535/> – Загл. с экрана.
3. Ивон, А.И. Использование критических терморезисторов для аппаратной реализации режима троттлинга центрального процессора [Текст] / А.И. Ивон, А.А. Андреев, Л.А. Можаровский. // Системні технології: сб. наукових праць. – Дніпропетровськ, 2009. – В. 1 (60). – С. 87 – 92.
4. Ивон, А.И. Расчет длительности „холостых” тактов при использовании режима троттлинга для защиты процессора от перегрева [Текст] / А.И. Ивон, В.Ф. Истушкин // Системні технології: сб. наукових праць. – Дніпропетровськ, 2012. – В. 1 (78). – С. 41 – 46.

.

НЕЙРОННЫЕ СЕТИ В ЗАДАЧАХ КЛАССИФИКАЦИИ ДЕФЕКТОВ

Аннотация. Исследована возможность применения нейронной сети Кохонена для классификации сигналов дефектоскопии. Обученная нейронная сеть промоделирована на проверочном множестве зашумленных сигналов.

Ключевые слова: нейронная сеть Кохонена, классификация сигналов, самообучение.

Введение. Задача определения дефектности изделий и принятие решения о годности связана с интенсивным использованием человеческого интеллекта и квалифицированного труда. Перспективным является использование методов искусственного интеллекта для решения этой задачи в автоматическом режиме без участия человека. Среди таких методов высокую эффективность продемонстрировали искусственные нейронные сети, и, прежде всего, самоорганизующиеся карты Т. Кохонена (*SOM - self organizing maps*)[1], положенные в основу автоматической классификации различных видов информации. Эффективность карт Кохонена определяется, прежде всего, их вычислительной простотой и возможностью работы в реальном времени путем последовательной обработки информации по мере ее поступления.

Целью данной работы является использование нейронных сетей Кохонена в задачах классификации зашумленных сигналов дефектоскопии.

Теоретическое обоснование. Стратегия метода построения *SOM* заключается в представлении множества экспериментальных данных в виде топологической карты – решетки размерностью $n \times m$, состоящей из нейронов. Каждый нейрон представляет собой вектор, определяющий в d -мерном пространстве признаков некоторую характерную точку. В результате реализации алгоритма обучения векторы нейронов распределяются в пространстве признаков, аппроксимируя

статистические взаимосвязи между данными обучающего множества [1,2].

В процессе работы алгоритма настраиваются синаптические веса нейронов. Для настройки сети определяется мера соседства нейронов (мера близости). Зоны соседства нейронов уменьшаются с течением времени. В начале обучения весовым коэффициентам сети присваиваются малые случайные значения

Общее число синаптических весов равно $n \times m$. Для всех нейронов вычисляется Евклидово расстояние до входного вектора и выбирается нейрон-

-победитель с наименьшим расстоянием. Затем производится подстройка весов для нейрона-победителя и всех нейронов из его зоны соседства. Модификации весов рассчитываются по формуле:

$$Wv(s + 1) = Wv(s) + \Theta(u, v, s) \alpha(s)(D(t) - Wv(s)), \quad (1)$$

где $Wv(s)$ – значение весового коэффициента связи входного нейрона и выходного нейрона в момент времени s , u – нейрон-победитель для $D(t)$, $\alpha(s)$ – шаг обучения, уменьшающийся с течением времени и $D(t)$ – входной вектор. Функция соседства $\Theta(u, v, s)$, зависящая от расстояния решетки между нейроном -победителем (нейрон u) и нейроном v . Далее предъявляется новое наблюдение и процесс повторяется.[1].

Основная часть. При сканировании изделий из композитных материалов пространственная база получаемых сигналов ограничена снизу диаметром зоны электромагнитного контроля. Поскольку амплитуда сигналов в каждой точке абсциссы определяется проекцией дефекта изделия на плоскость, перпендикулярную направлению вихревых токов, то интегральной информативной характеристикой таких сигналов есть их площадь.

Плавное изменение формы сигнала от унимодального с максимальной амплитудой (дефекты превышают зону контроля) до бимодального с наибольшим провалом вершины (точечные дефекты) моделируется с помощью выражения[3]:

$$y(x) = (2^n - 1) \cdot \exp(-1,5x^2) - k \cdot \exp(-3x^2), \quad (2)$$

где $(2^n - 1)$ – это максимальное значение сигнала n -разрядного АЦП.

В результате получаются сигналы различной формы: при $k = 0 \div 0.35$ – узкий унимодальный сигнал, описывающий длинную трещину, длина которой превышает зону контроля. При изменении $k = 0.35 \div 0.55$ получаем пологий унимодальный сигнал, характерный для трещин меньшей размерности; при $k = 0.6 \div 1$ получаем бимодальный сигнал, который характерен для маленьких трещин (при $k = 1$ – точечный дефект).

Площадь получаемых модельных сигналов вычисляется путем интегрирования выражения (2) по абсциссе от $-\infty$ до $+\infty$, и является разностью известных [4] площадей под двумя ненормированными гауссовыми кривыми со значениями параметра $\sigma = 1/\sqrt{3}$ и $\sigma' = 1/\sqrt{6}$:

$$Q = (2^n - 1) \sqrt{\frac{2\pi}{3}} \left(1 - \frac{\sqrt{2}}{2} k\right), \quad (3)$$

где Q измеряется в квадратных относительных единицах длины, приведенной к эквивалентному диаметру катушки вихретокового преобразователя (ВТП).

В случае АЦП с разрядностью $n=8$ площади соответствующих сигналов ВТП изменяются от минимального значения $Q_n = 110.71$ до максимального $Q_m = 369.036$, после чего увеличение размеров дефектов не регистрируется.

Практическая реализация. Для моделирования использовалась вычислительная среда Matlab R2010b. В качестве обучающего множества для нейронной сети Кохонена использовались значения площадей модельных сигналов (3).

Для проверки качества классификации выбирались значения площадей зашумленных сигналов. Шум со средним значением 0 и стандартным отклонением от 0 до 0,3 с шагом 0,05 добавлялся к векторам входа. Для вычисления площади таких сигналов использовалась формула трапеций [4].

Самоорганизующаяся сеть Кохонена формировалась в виде одномерного слоя из 3 нейронов, в соответствии с числом классов дефектов. Выполнялось обучение на протяжении 200 итераций (рис. 1).

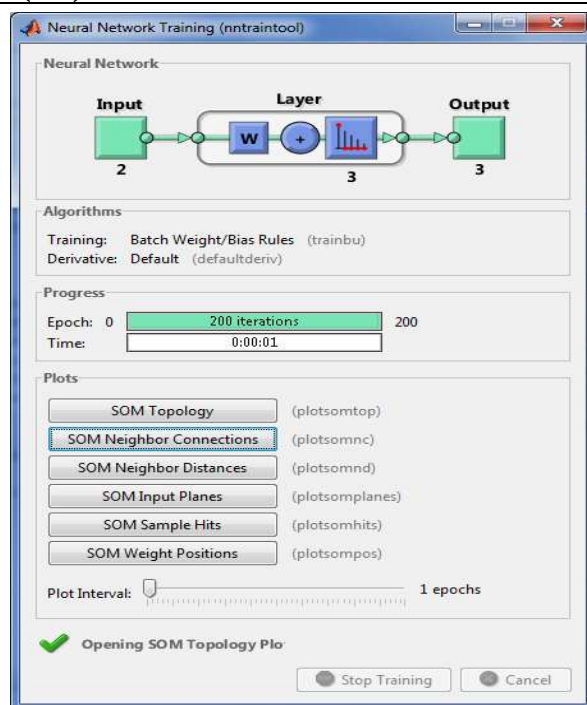


Рисунок 1- Моделирование сети Кохонена в Neural Networks Toolbox

Обученную нейронную сеть использовали для классификации входных векторов из проверочного множества (рис. 2).

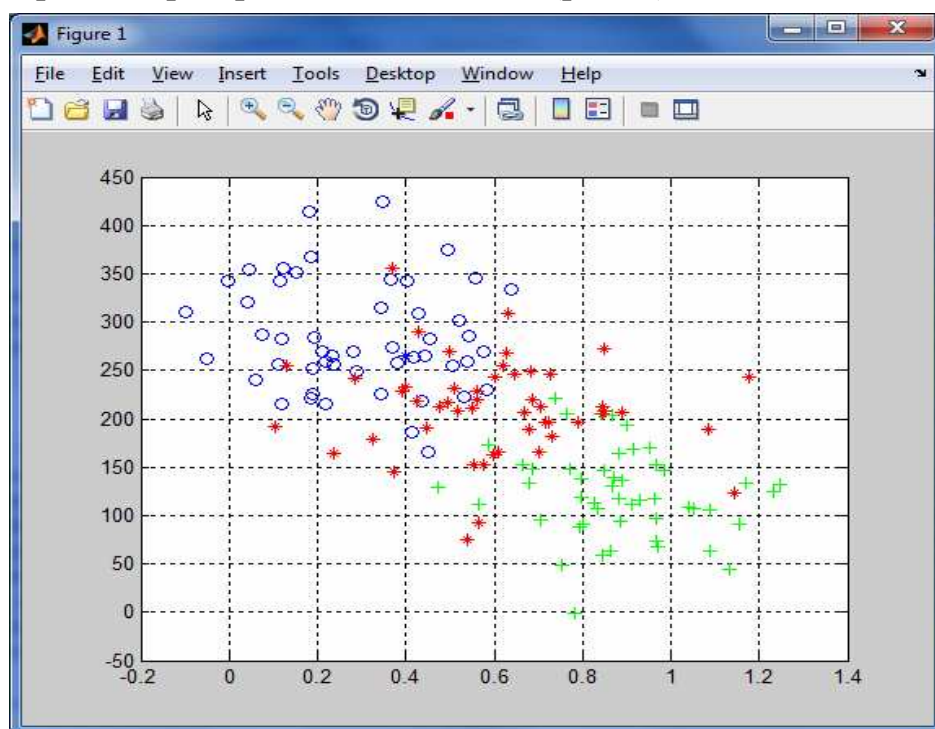


Рисунок 2 - Распределение зашумленных сигналов

После классификации нейронная сеть выдавала такие результаты (рис.3):

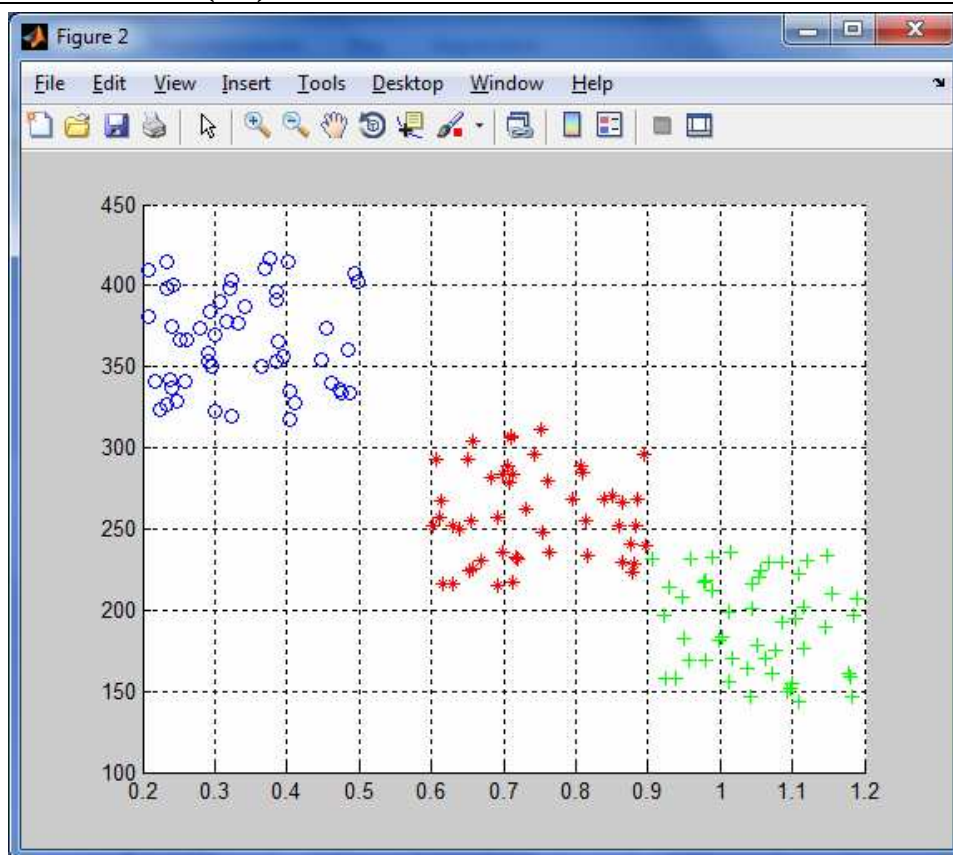


Рисунок 3 - Распределение кластеров нейронной сети Кохонена

Для оценивания эффективности функционирования созданной нейронной сети проводилось ее тестирование на 1000 векторах входа для различных уровней шума и определялась доля неправильно классифицированных сигналов:

$$P = \frac{n}{N} \times 100, \%$$

где n – число неправильно классифицированных сигналов, N – их общее количество.

Таблица 1

Количество неправильно классифицированных сигналов

Значение шума, σ	0	0,05	0,1	0,15	0,2
P	0	3,4	8	15	25,4

Выводы. В данной работе проанализированы возможности применения нейронной сети Кохонена для классификации зашумленных сигналов при проведении неразрушающего контроля. Эти сети характеризуются тем, что настройка синаптических весов происходит в режиме самообучения, при этом каждый поступающий на вход сигнал вызывает адаптацию

тех или иных параметров искусственной нейронной сети. Это процесс может протекать непрерывно, обеспечивая возможность решения задачи в реальном времени.

ЛИТЕРАТУРА

1. Kohonen T. Self-Organizing Maps / T. Kohonen // Berlin :Springer-Verlag. – 1995. – 362 p.
2. Хайкин Саймон. Нейронные сети: полный курс, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2006
3. Хандецкий В.С. Спектральная идентификация сигналов в дефектоскопии композитов с использованием теории статистических испытаний / Хандецкий В.С., Герасимов В.В. // Вісник ДНУ: Фізика. Радіоелектроніка. – Дніпропетровськ: – 2003. № 10. – С. 128 – 132
4. Бронштейн И.Н. Справочник по математике для инженеров и учащихся втузов/ И.Н. Бронштейн, К.А. Семендяев – М.: Наука, 1986.

І.І. Атаманенко, Д.П. Сівцов

ВИКОРИСТАННЯ МОДУЛІВ НА АСЕМБЛЕРІ ДЛЯ ЗБІЛЬШЕННЯ ПРОДУКТИВНОСТІ C / C++ ПРОГРАМ СИМЕТРИЧНОГО ШИФРУВАННЯ

Анотація. У статті обговорюється можливість збільшення продуктивності C/C++ програм для шифрування інформації на основі симетричних криптографічних алгоритмів. Було встановлено, що істотний ефект може бути досягнутий шляхом реалізації частин таких програм на асемблері.

Ключові слова: симетричні криптоалгоритми, асемблер, продуктивність.

Введення. На теперішній час симетричні криптоалгоритми використовуються в основному для забезпечення конфіденційності інформації на локальних комп'ютерах, робочих станціях і серверах мереж (шифрування), генерації кодів перевірки цілісності мережних повідомлень і електронних документів (технологія MAC), віддаленої мережної автентифікації (протокол Kerberos), забезпечення безпеки мережних застосувань і протоколів стеку TCP/IP (PGP, IPsec, SSL/TLS, SET і т.д.). Окрім цього, симетричні криптоалгоритми можуть використовуватися в якості генераторів псевдовипадкових послідовностей для потокового шифрування (DES в режимі OFB, ГОСТ 28147-89 в режимі гамування) для генерації сеансових ключів (стандарт ANSI X9.17 на основі 3-DES), а також для інших цілей [1,2].

У зв'язку з тим, що потоки інформації в сучасних комп'ютерних системах і мережах характеризується великим об'ємом інформації і швидкістю передачі, криптографічне програмне забезпечення та апаратні засоби повинні задовольняти зростаючі потреби не тільки щодо криптостійкості, але і щодо експлуатаційних характеристик.

Постановка задачі. Мета цієї роботи - визначення і експериментальне дослідження способу поліпшення експлуатаційних особливо-

стей реалізації програм симетричних шифрувальних алгоритмів в застосуванні до блокових шифрів.

Основна частина. Однією з важливих експлуатаційних характеристик програмних реалізацій симетричних криптоалгоритмів є продуктивність.

Дійсно, продуктивність є однією з основних причин апаратної реалізації симетричних криптоалгоритмів (криптопроцесори, криптокарти) [3].

З тих же міркувань програмні реалізації алгоритмів симетричного шифрування виконуються на мовах програмування C або C++.

Наприклад, у [2,4] наведені численні вихідні коди програм блокових і потокових шифрів, створені на мовах C/C++.

Програмна реалізація симетричних криптографічних алгоритмів лише мовами високого рівня C або C++ на основі програмних платформ, наприклад, таких як MS Visual Studio, дозволяє отримати виконуваний код досить високої якості. Але це не гарантія того, що виконуваний код буде оптимальним в плані продуктивності.

Це обумовлено тим, що продуктивність залежить від ряду факторів, які викликані особливостями і характеристиками алгоритму, середовища програмування, використовуваної операційної системи та апаратних засобів.

Розглянемо деякі особливості симетричних криптоалгоритмів в контексті вказаних факторів.

Для визначеності обговоримо тільки алгоритми блокового шифрування / розшифрування (або скорочено - блокові шифри).

Довжина блоків (або внутрішніх субблоків) і ключів шифру (або внутрішніх субключів) зазвичай коливається від 16 до 512 біт, що порівнянне з розміром форматів даних мов програмування C або C++, а головне - з розрядністю регістрів загального призначення сучасних процесорів.

Алгоритми блокових шифрів будуються на використанні переважно таких бітових операцій, які можуть бути достатньо просто реалізовані на регістровому рівні процесора, а саме:

- побітова операція *XOR* над n -бітними послідовностями;
- додавання або множення за модулями 2^n або 2^n-1 ;
- операції простого і циклічного зсувів на 2^n біт;
- обмін вмістом між блоками або між субблоками;

- операції n -бітних підстановок, перестановок, конкатенації;
- деякі операції на кінцевих полях $GF(2n)$;

Примітка: суттєво, що в усіх випадках де n – ціле, яке порівнянне з розрядністю регістрів загального призначення сучасних процесорів.

Тому можна зробити висновок, що для програмування розділів алгоритму блочного шифрування, які потребують таких операцій, асемблер буде найкращим інструментальним засобом. У такому випадку ці розділи будуть виконуватися на рівні машинних команд, а значить, з максимальною швидкістю.

Окрім операцій, зазначених вище, в блокових шифрах також використовуються і такі операції, які проблематично реалізувати тільки на регістровому рівні. Наприклад, це читання даних з масиву цілочисельних констант, операції перетворення на основі S-блоків, і так далі.

Збереження подібних масивів або блоків часто не може бути виконано тільки за підтримки масиву регістрів загального призначення процесора, а тому потребує розміщення їх в пам'яті.

Тим не менше, і в цьому випадку, принаймні, частина програмного коду відповідного розділу алгоритму може бути реалізовано на асемблері.

Таким чином ми можемо припустити, що вірне рішення в напрямку покращення швидкості програмного продукту, яка здійснює алгоритм блокового шифру, полягає в тому, щоб використовувати модулі асемблера разом з вихідним кодом на C/C++. Це твердження буде справедливим не тільки для блочних шифрів, а й для інших застосувань симетричних криптоалгоритмів. Подібне рішення підтримується також тим, що сучасні компілятори мов високого рівня мають, як правило, вбудований асемблер (у тому числі компілятори мов C/C++).

Для практичної реалізації запропонованого рішення, можна застосувати наступні варіанти сумісного використання модулів на асемблері і програмного коду на мові програмування високого рівня [4,5,6].

1. Компіляція об'єктного модулю, який містить одну або декілька процедур обробки даних, виконується окремо від основної програми. Зовнішній об'єктний модуль на асемблері приєднується до ос-

новної програми на етапі компоновки. Асемблерна процедура, оголошена відповідним чином, викликається з будь якого місця основної програми. Переваги:

- можливість використання асемблерних процедур в програмах, написаних на різних мовах і навіть в різних операційних системах;

- незалежність процесу розробки кінцевого продукту програмного забезпечення і процедур на асемблері;

Недолік - необхідність враховувати механізм виклику зовнішніх процедур і передачі параметрів у процедуру, яка викликається.

2. Використання вбудованого асемблера. Переваги:

- для процедури, яка була розроблена в основній програмі не потрібно спеціальних дій для зв'язування цієї процедури з викликаючою програмою.

- порядок передачі параметрів у процедуру, яка викликається, та відновлення стеку забезпечується автоматично.

- більш простіша і швидка відладка програмного продукту в цілому;

- Недоліки:

- певні обмеження, які накладає середовище програмування на виконання асемблерних модулів;

- процедури, розроблені за допомогою вбудованого асемблера, не можна перетворити у зовнішні модулі, які можуть використовуватися окремо.

З метою практичної перевірки запропонованого підходу було обрано шифр Віженера, який був реалізований як C++ програма, яка включала асемблерні модулі. Платформа програмування – MS Visual Studio 2010. Для створення асемблерних процедур використовувався вбудований асемблер (тобто варіант 2, описаний вище). Програма шифрує або розшифровує ASCII-текст з ключем, довжина L якого менша за довжину відкритого тексту. Таким чином, обробка тексту чи криптограми виконується по блокам довжиною L . Для проведення порівняння були розроблені дві версії програмної реалізації шифру Віженера. Перша версія виконана виключно на C++. Друга версія виконана в контексті запропонованого підходу, тобто на основі сумісного використання C++ і асемблера.

На рис.1 наведено вихідний код процедури шифрування за першою версією програми (тобто тільки на C++). Ця процедура разом з процедурою розшифрування визначає функціональність програми в цілому. Ділянка процедури, в якій виконується основна робота - розрахунок символів криптограми, відзначена курсивом:

```

.....
class crypt
{
public:
    string content;
    string crypt_content;
    string uncrypt_content;
    string key;
    bool doCrypt(string content="", string key=""){
        if(content.empty() && this->content.empty() ||
        key.empty() && this->content.empty()) return false;
        if(!content.empty()) this->content = content; else
content = this->content;
        if(!key.empty()) this->key = key; else key = this->key;
        int a=0;
        string crypt_content;
        crypt_content.clear();
        for(int i=0; i<content.length(); i++)
        {
            crypt_content+=content[i]+key[a];
            if(a<key.length()-1) a++; else a=0;
        }
        this->crypt_content = crypt_content;
        return true;
    }
}
.....

```

Рисунок 1 - Лістинг програмного коду процедури шифрування, виконаної виключно на C++

На рис.2 наведено вихідний код процедури шифрування другої версії програми, у якому та ж ділянка реалізована у вигляді асемблерного модулю. Для наочності цю ділянку виділено також курсивом:

```

.....
int a=0, b=0;
    int keyL;
    char buf[256], bufk[256];

    keyL=key.length();
    b=content.length();
    this->crypt_content.clear();
    // addition of ASCII-plain text with cipher key
    while(content.length()-a>0)
    {
        (content.length()-
a>256)?(b=256):(b=content.length()-a);
        for(int i=0;i<b;i++)
            buf[i]=content[i+a];
        for(int i=0;i<keyL;i++)
            bufk[i]=key[i];
        _asm{
            xor EBX, EBX
            mov BL, 1
            mov ECX, b
            lea ESI, buf
            lea EDX, bufk
next_ch:  mov AL, [EDX]
                                add [ESI], AL

            inc ESI
            inc EDX
            inc EBX
            cmp EBX, keyL
            jg decK
            dec ECX
            jnz next_ch
            jmp ex
decK:    sub EDX, keyL

```

```

xor EBX, EBX
mov BL,1
jmp decc

ex:
};

for(int i=0;i<b;i++)
    this->crypt_content+=buf[i];
    (content.length()-
a>256)?(a+=256):(a+=content.length()-a);
}
return true;
}

.....

```

Рисунок 2 – Лістинг програмного коду процедури шифрування на мові C++ і на асемблері

Для оцінки і порівняння продуктивності у тому і другому випадку були обрані наступні числові показники:

- значення середнього часу виконання програмного коду, тобто обробки текстового файлу в режимі шифрування;
- кількість тактів процесора, витрачених на виконання даного програмного коду.

Перша версія програми підраховує значення зазначених числових показників використовуючи виключно C++ програмний код, друга – використовуючи блоки асемблера всередині C++ програмного коду.

Відповідний фрагмент вихідного коду програми наведений на рис. 3. Для наочності асемблерні фрагменти виділені курсивом так, як і вище:

```

.....
start=clock();
long long t1,t2;
long long delta_t;
asm{
    CPUID
    RDTSC
}

```

```

mov DWORD PTR[t1], eax
mov DWORD PTR[t1 + 4], edx
}
crypt.doCrypt(crypt.content, key);
_asm{
CPUID
RDTSC
mov DWORD PTR[t2], eax
mov DWORD PTR[t2 + 4], edx
}
stop=clock();
.....

```

Рисунок 3 – Вихідний код для обчислення значень показників продуктивності з використанням асемблерних команд

Внаслідок симетричності криптоалгоритму Віженера, величини середнього часу і кількості тактів процесора практично однакові як для шифрування так і для розшифрування (і це було підтверджено в ході тестувань). З тих же міркувань не наведено програмного коду процедури розшифрування. Тому ми далі обговорюємо результати експериментів тільки шифруванням.

Наведемо конкретний приклад. Зашифровувався ASCII-текст з загальною кількістю символів 533 498. Довжина ключа шифрування - 8 символів.

Отримані такі результати характеристик продуктивності. Для першої версії програми (що написана виключно на C++) середній час її виконання дорівнює 0,3691 с. Для другої версії програми (написана на C++ з використанням асемблерних модулів) цей час становить 0,2072 с.

Загальна кількість тактів процесора на виконання першої версії програми дорівнює $\approx 1265 \cdot 10^6$, а для другої версії - відповідно $\approx 641 \cdot 10^6$.

Таким чином за допомогою модулів асемблера ми досягли зменшення середнього часу виконання шифрування приблизно на 43%. Крім того загальна кількість циклів процесора зменшилася приблизно на 51%.

ЛІТЕРАТУРА

1. Stallings W., Cryptography and Network Security: Principles and Practice (5th Edition) [Електронний ресурс] / W. Stallings – NY.: Prentice Hall, 2011. – 721 pp. // WORDPRESS.COM – 2012 – Режим
доступа: <http://mrajacse.files.wordpress.com/2012/01/cryptography-network-security-5th-edition.pdf>
2. Шнайер, Б. Прикладна криптографія. Протоколи, алгоритми, вихідні тексти на мові Сі [Текст] / Б. Шнайер – М.: Видавництво Триумф, 2002. – 816 с.
3. Коркішко Т., Алгоритми та процедури симетричного блокового шифрування [Текст] / Т. Коркішко, А. Мельник, В. Мельник – Львів: Бак, 2003. – 168 с.
4. Аграновский А.В., Практическая криптография: алгоритмы и программирование Сі [Текст] / А.В. Аграновский, Р.А. Хади – М.: СОЛОН-Пресс, 2009. – 256 с.
5. Магда Ю.С., Використання асемблера для оптимізації програм на С++ [Текст] / Ю.С. Магда – СПб.: БХВ-Петербург, 2004. – 496 с.
6. Кулаков, В. Програмування на апаратному рівні: спеціальний довідник [Текст] / В. Кулаков – СПб.: Пітер, 2003. – 847 с.
7. Уоррен, Г. Алгоритмічні трюки для програмістів [Текст] / Г. Уоррен – М.: Видавничий дім "Вільямс", 2004. – 288 с.

В.В. Крохин, Н.Ю. Митиков

ОБ ИССЛЕДОВАНИИ КАЧЕСТВА ПРОЕКТИРОВАНИЯ ЦИФРОВЫХ ФИЛЬТРОВ

Аннотация. Предложено использовать для оценки качества проектирования цифровых фильтров не только уровень пульсаций в полосе задержания и пропускания, но и ширину переходной зоны между этими полосами. Разработано программное обеспечение, позволяющее определять ширину переходной полосы. С использованием предложенной методики проведен анализ качества фильтров, разработанных с помощью различных весовых функций.

Ключевые слова: цифровой фильтр, весовые функции, уровень пульсаций, ширина переходной полосы.

Введение. При наличии большого числа методов проектирования цифровых фильтров, значительный интерес представляет вопрос о сравнительных характеристиках их качества [1,2]. В данной статье предлагается методика, позволяющая автоматизировать анализ качества разрабатываемых цифровых фильтров. На основе предложенной методики, проводится сравнительный анализ характеристик нерекурсивных цифровых фильтров, построенных с использованием различных весовых функций. [2,3]. Для этой цели разработано специализированное программное обеспечение.

При построении дискретных фильтров с применением различных весовых функций использовался МАТЛАБ - пакет, в котором имеется множество встроенных функций для проектирования фильтров, а также средства измерения и визуализации характеристик получаемых устройств. [1,4].

Постановка задачи. Задачей настоящей работы является сравнительный анализ качества разрабатываемых цифровых фильтров на основе количественного определения не только уровня пульсаций, но и ширины переходной зоны между полосами пропускания и задержания.

Основная часть. Анализ качества разработанного фильтра проводится на основе измерения следующих параметров:

- ширина переходной зоны между полосой пропускания и полосой задержания ($\Delta W_{пер.}$);
- уровень пульсаций в полосе пропускания ($P_{проп.}$);
- уровень пульсаций в полосе задержания ($P_{зад.}$);

Для целей автоматизации сравнительного анализа качества разрабатываемых фильтров большое значение имеет алгоритмически корректное определение характеристик, которые должны быть измерены. В этом отношении, наибольшая проблема связана с определением ширины переходной зоны между полосами пропускания и задержания [3]. В связи с наличием пульсаций, определение ширины этой зоны, как расстояния по оси частоты между максимальным и минимальным значениями амплитудно-частотной характеристики (АЧХ), не представляется корректным. Поэтому предлагается для определения ширины переходной зоны в случае проектирования фильтра нижних частот (ФНЧ) использовать величину:

$$\Delta W_{пер.} = W_{проп.} - W_{зад.},$$

где $W_{проп.}$ - точка АЧХ, в которой амплитуда последний раз пересекает уровень $k_1 A_{max}$, а $W_{зад.}$ - точка АЧХ, в которой амплитуда в первый раз пересекает уровень $k_2 A_{max}$. (Здесь A_{max} - максимальное значение амплитуды в области пропускания). Величина коэффициентов k_1 и k_2 может выбираться произвольно, в зависимости от требований к проектируемому устройству. Вычисление характеристики $\Delta W_{пер.}$ а также параметров $P_{проп.}$ и $P_{зад.}$ иллюстрирует рис. 1.

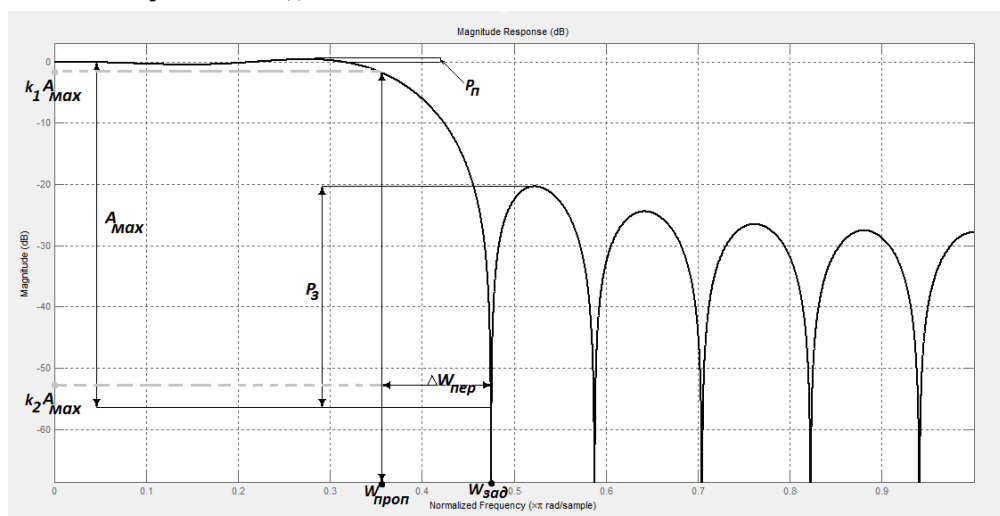


Рисунок 1 - Определение характеристик качества ФНЧ по АЧХ

Для фильтров других типов (фильтра высоких частот, полосового или заградительного) изменения в определении $\Delta W_{пер}$ являются очевидными.

Для вычисления $\Delta W_{пер}$ была разработана специальная программа на языке C Sharp с использованием Windows.Forms, для которой k_1 и k_2 являются входными параметрами. Данная технология позволяет быстро разработать программное обеспечение с пользовательским интерфейсом. Использование данной программы позволяет быстро и наглядно сравнивать качество разрабатываемых цифровых фильтров.

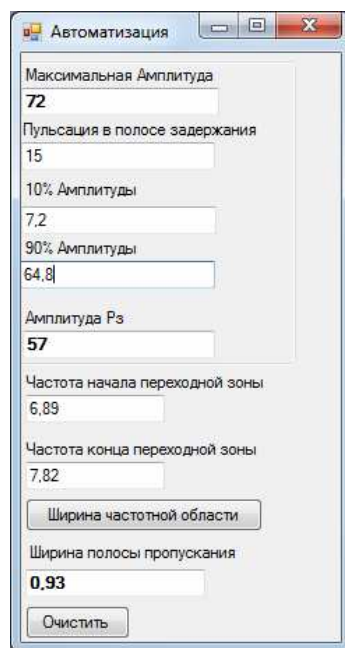


Рисунок 2 - Программа для автоматизации вычислений

В качестве примера нами было проанализировано качество проектирования нерекурсивных фильтров, получаемых квазиоптимальным методом с использованием различных весовых функций (окон). [1]. Применялись следующие весовые функции:

- Прямоугольное окно
- Окно Хэмминга
- Окно Чебышева
- Окно Кайзера
- Окно Гаусса
- Окно Тукея

Параметры измерялись в соответствии с изложенной методикой. Использовались значения $k_1 = 0,9$ и $k_2 = 0,1$. В приведенной ниже

таблице 1 показаны значения параметров спроектированных фильтров в зависимости от порядка фильтра.

Таблица 1

Вид весовой функции	Порядок	$\Delta W_{\text{пер}}$, кГц	$ A_{\text{max}} $, дБ	P_z , дБ
Прямоугольное	8	3,65	52	31,3
	16	2,2	46	26,35
	32	0,95	40	19,4
	64	0,5	37,45	15,75
Хемминга	8	11,65	64	22,02
	16	5,79	68	11,72
	32	2,89	65	11,16
	64	1,3	65	11
Гаусса $\sigma=1,25$	8	9,68	61	21,82
	16	3,48	47	17,7
	32	2,38	44,25	14
	64	0,73	39,31	8,07
Чебышева $\beta=100$	8	21,37	134	20,2
	16	12,44	128,64	19,4
	32	7,17	122	18,28
	64	3,69	121,3	7,61
Кайзера $\alpha=4$	8	7,53	61,6	21,2
	16	4,4	62	12,3
	32	2	56,53	8,9
	64	0,83	54,12	8,2
Тукея $\alpha=1$	8	10,43	65	19,5
	16	5,27	63,5	17,2
	32	2,11	52	8
	64	0,99	50	6,2

Для сравнительного анализа полученных результатов удобно представить данные из таблицы 1 в виде графиков. На рисунке 3 приведена зависимость длины переходной полосы от порядка фильтра. Чем меньше переходная полоса, тем быстрее происходит переключение фильтра из состояния пропускания в состояние задержания сигналов.

Анализ графиков, приведенных на рис. 3 показывает, что для всех проанализированных типов фильтров ширина переходной зоны $\Delta W_{пер}$ убывает по мере увеличения порядка фильтра, что вполне естественно. Однако это уменьшение является нелинейным и происходит значительно быстрее для невысоких значений порядка фильтра. Среди различных типов весовых функций, наименьшая ширина переходной зоны наблюдается для прямоугольного окна.

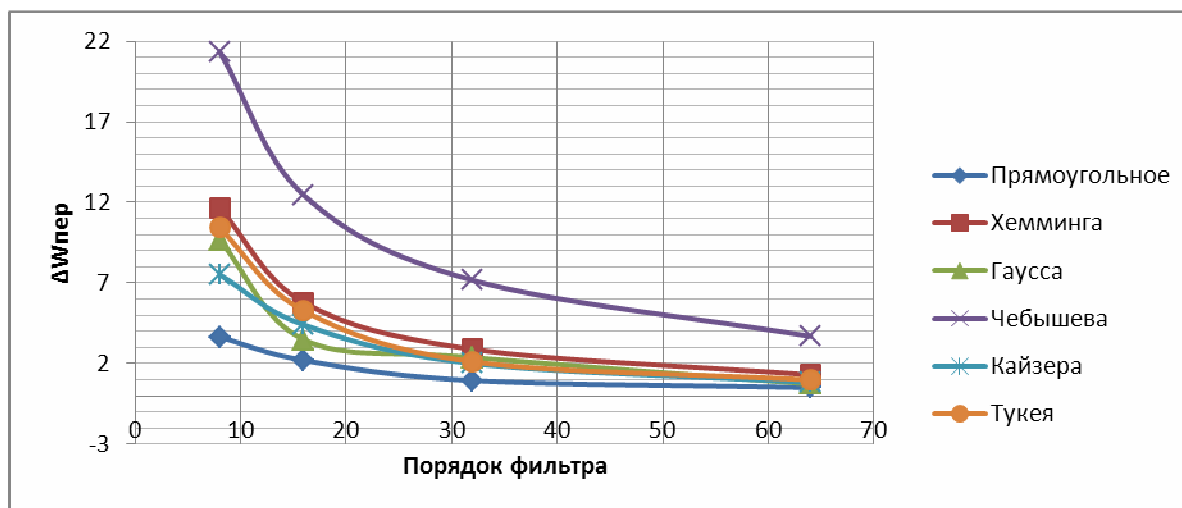


Рисунок 3 - Зависимость длины переходной полосы от порядка фильтра

На рисунке 4 приведена зависимость уровней пульсаций в полосе задержания от порядка фильтра.

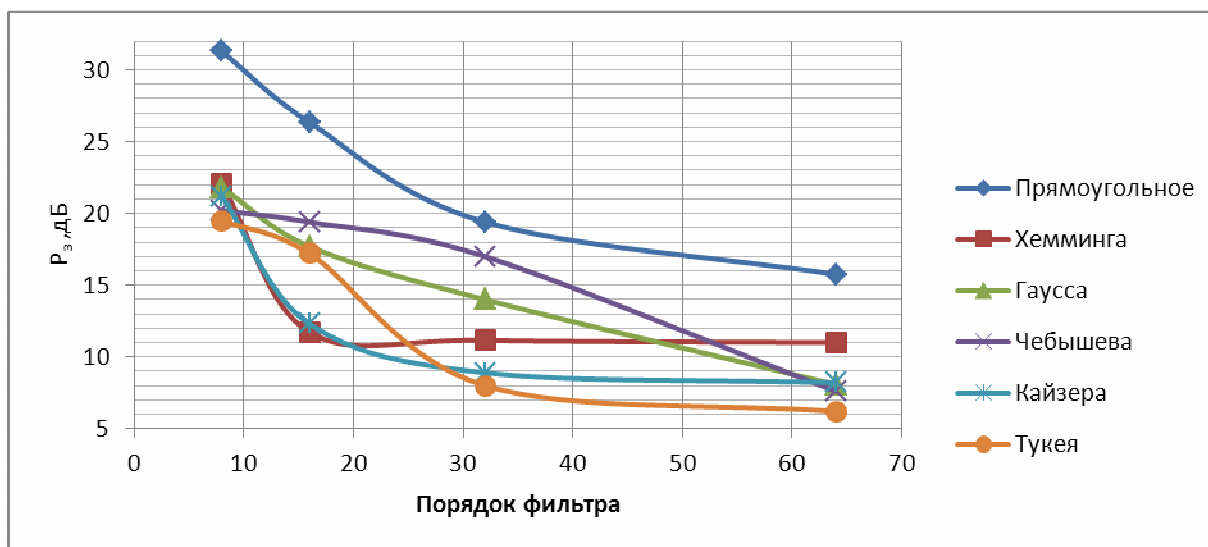


Рисунок 4 - Зависимость уровней пульсаций в полосе задержания от порядка фильтра

Эти графики показывают, что при использовании прямоугольного окна пульсации в полосе задержания являются наибольшими. С точки зрения уровня пульсаций $P_{\text{зад}}$ наилучшие результаты получаются при использовании окна Чебышева.

Выводы. Таким образом, выбор оптимальной весовой функции с точки зрения только одного из параметров качества не представляет затруднений. Однако при оптимизации по нескольким параметрам выбор не является столь очевидным. Сравнение графиков, приведенных на рисунках 3 и 4, дает основание рекомендовать использовать окно Кайзера. Данный тип окна позволяет получить достаточно узкую полосу переключения и невысокий уровень пульсаций в полосе задержания.

ЛИТЕРАТУРА

1. А.Б. Сергиенко. "Цифровая обработка сигналов ", СПб.: Питер, 2002.
2. A.R. Collins "FIR Filter Design", July 2011, University of Oslo.
3. F.P.G. Márquez "Digital Filters" Publisher: InTech., Published: April 11, 2011 under CC BY-NC-SA 3.0 license.
4. Z. Milivojević "Digital Filter Design", mikroElektronika, 2009.

.

Н.И. Твердоступ

ОБ ЭФФЕКТИВНОСТИ АЛГОРИТМА АДАПТИВНОГО СГЛАЖИВАНИЯ В УСЛОВИЯХ ВЛИЯНИЯ МЕШАЮЩИХ ФАКТОРОВ

Аннотация. Исследовано влияние экспоненциального сглаживания на погрешность результата измерения индуктивности параметрического датчика, расположенного на поверхности железорудного штуфа. Установлено, что сглаживание почти на порядок уменьшает погрешность результата измерения при мешающем влиянии неоднородности магнитного поля датчика.

Ключевые слова: экспоненциальное сглаживание, результат измерения, погрешность, индуктивный датчик, квазиосевая симметрия, железорудный штуф, неоднородность поля.

Вступление. Целью определения количественного значения любой физической величины является получение надежного результата с минимальной погрешностью. В экспериментальной практике эта цель достигается предварительной обработкой значений отдельных измерений. Определяющим при этом является выбор оптимального алгоритма обработки, так как от него, в большинстве случаев, зависит качество полученного результата.

Анализ исследований. Из множества алгоритмов обработки измерительной информации выделяются методы линейного и нелинейного сглаживания, основанные на замене эмпирических данных расчетными, имеющими значительно меньшую колеблемость, чем исходные данные, что позволяет уменьшить погрешность результата измерения [1]. В настоящее время для обработки временных рядов используют адаптивный алгоритм экспоненциального сглаживания [2], суть которого сводится к преобразованию текущих значений переменной X_i согласно рекуррентной формуле Брауна [3]

$$\tilde{X}_i = \alpha X_i + (1 - \alpha) \tilde{X}_{i-1}, \quad (1)$$

где $\tilde{X}_i$, $\tilde{X}_{i-1}$ – экспоненциальные средние (сглаженные значения уровней временного ряда) на моменты времени i и $i-1$ соответственно; α – параметр сглаживания ($0 < \alpha \leq 1$). Экспоненциальное сглаживание широко используют для решения задач социально-экономического прогнозирования [4, 5]. Представляет интерес определение эффективности этого алгоритма при обработке числовых рядов в виде последовательностей значений отдельных измерений некоторой физической величины, полученных в условиях влияния мешающих факторов, которые способствуют уменьшению надежности результата измерения и его точности.

Постановка задачи. Целью настоящей работы является определение степени влияния экспоненциального сглаживания на погрешность результата измерения в зависимости от количества эмпирических данных, величины параметра сглаживания и влияния отдельных мешающих факторов.

Основная часть. Для исследования влияния экспоненциального сглаживания на погрешность ΔX результата измерения в качестве числовых рядов были использованы значения X_i индуктивности параметрического датчика, располагаемого на поверхности железорудного штуфа, представляющего собой слабомагнитное вещество с сильной анизотропией, с целью определения действительной величины индуктивности в виде среднего арифметического X_{cp} . Влияние на результат измерения мешающих факторов было обеспечено использованием в качестве параметрических датчиков катушек с полной осевой или квазиосевой симметрией, существенно отличающихся величиной градиента магнитного поля в зоне взаимодействия с поверхностью штуфа, при этом датчики располагали или на гладкой поверхности штуфа или на шероховатой со случайным характером распределения кривизны. Эти мешающие факторы влияют на величину неоднородности магнитного поля датчика, которая является основным источником погрешности при контроле магнитной проницаемости железорудных штуфов.

Для оценки динамики поведения X_{cp} и ΔX в зависимости от количества измерений N среднее значение определяли как кумулятивное скользящее

$$X_{cp}(N) = \frac{1}{N} \sum_{i=1}^N X_i$$

при $1 \leq N \leq 100$, а погрешность среднего как доверительный интервал ΔX с вероятностью 0,95

$$\Delta X(N) = t_N \frac{\sqrt{\sum_{i=1}^N (X_i - X_{cp})^2}}{\sqrt{N(N-1)}},$$

где t_N – коэффициент Стьюдента [1]. Величины $X_{cp}(N)$ и $\Delta X(N)$ вычисляли для исходного и сглаженного согласно (1) числовых рядов. В качестве действительной величины индуктивности датчика были приняты значения X_{cp} при $N = 100$ для исходного числового ряда.

Оптимизация параметра сглаживания α проведена при наиболее сильном влиянии неоднородности магнитного поля датчика. Этому условию соответствует числовой ряд значений индуктивности датчика с квазиосевой симметрией, расположенного на шероховатой поверхности при выбранных значениях α : 0,01, 0,1 и 0,5. На рис. 1, 2, 3 представлены наиболее характерные результаты сглаживания при $\alpha = 0,1$. Видно, что сглаживание существенно уменьшает диапазон колебаний значений исходного ряда X_i и кумулятивного скользящего среднего X_{cp} , уменьшает погрешность среднего ΔX . Количественные результаты оптимизации параметра сглаживания показаны в таблице 1.

Таблица 1

Влияние параметра сглаживания α на результат измерения и его погрешность

Действительная величина	Исходный ряд	Сглаженный ряд		
		$\alpha = 0,01$	$\alpha = 0,1$	$\alpha = 0,5$
Результат измерения, X_{cp}	5,789	5,775	5,788	5,790
Погрешность, ΔX	$\pm 0,016$	$\pm 0,001$	$\pm 0,002$	$\pm 0,009$

Из таблицы 1 следует, что значение параметра $\alpha = 0,1$ является наиболее близким к оптимальному, так как разность результатов измерений X_{cp} до и после сглаживания составляет только 0,001 по сравнению с 0,014 при $\alpha = 0,01$, также погрешность ΔX равна $\pm 0,002$, что меньше погрешности $\pm 0,009$ при $\alpha = 0,5$. При значении $\alpha = 0,1$ получена погрешность результата измерения, которая в восемь раз меньше погрешности для исходного (несглаженного) ряда.

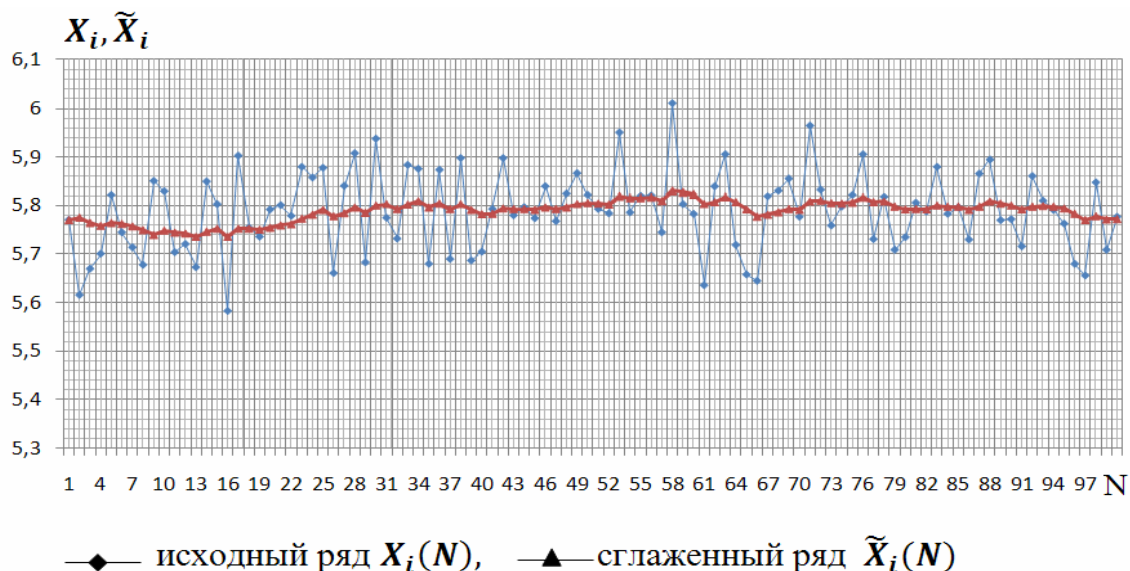


Рисунок 1 – Исходные X_i и сглаженные $\tilde{X}_i$ значения в зависимости от количества измерений N (параметр $\alpha = 0,1$)

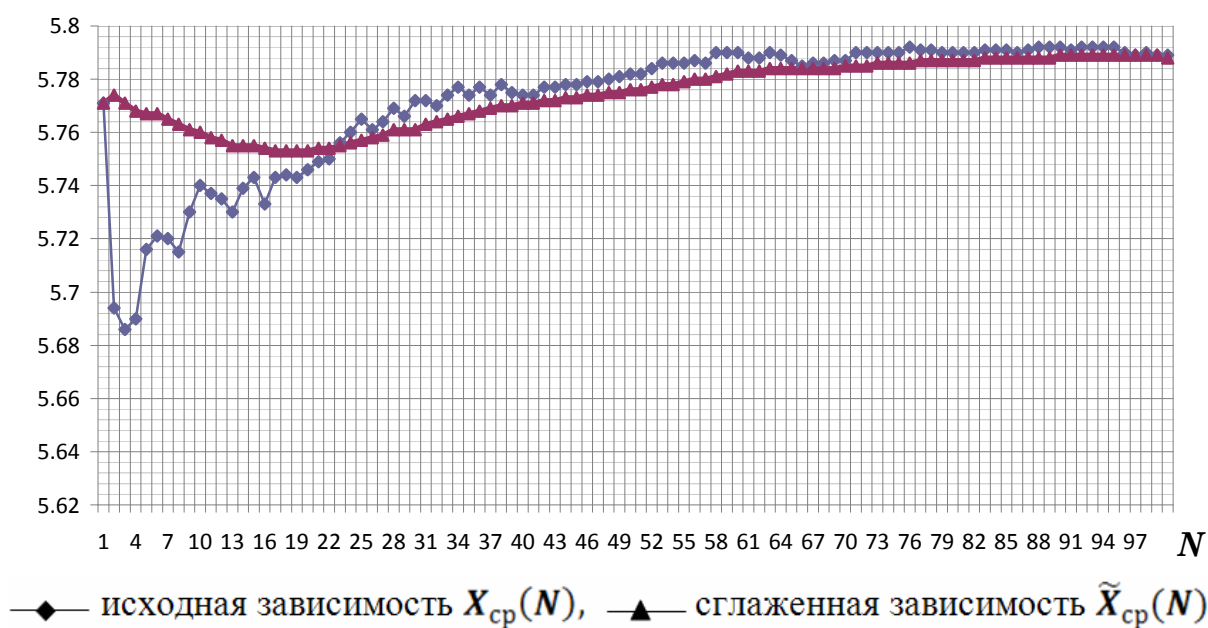


Рисунок 2 – Кумулятивное скользящее среднее исходных X_{cp} и сглаженных $\tilde{X}_{cp}$ значений ($\alpha = 0,1$)

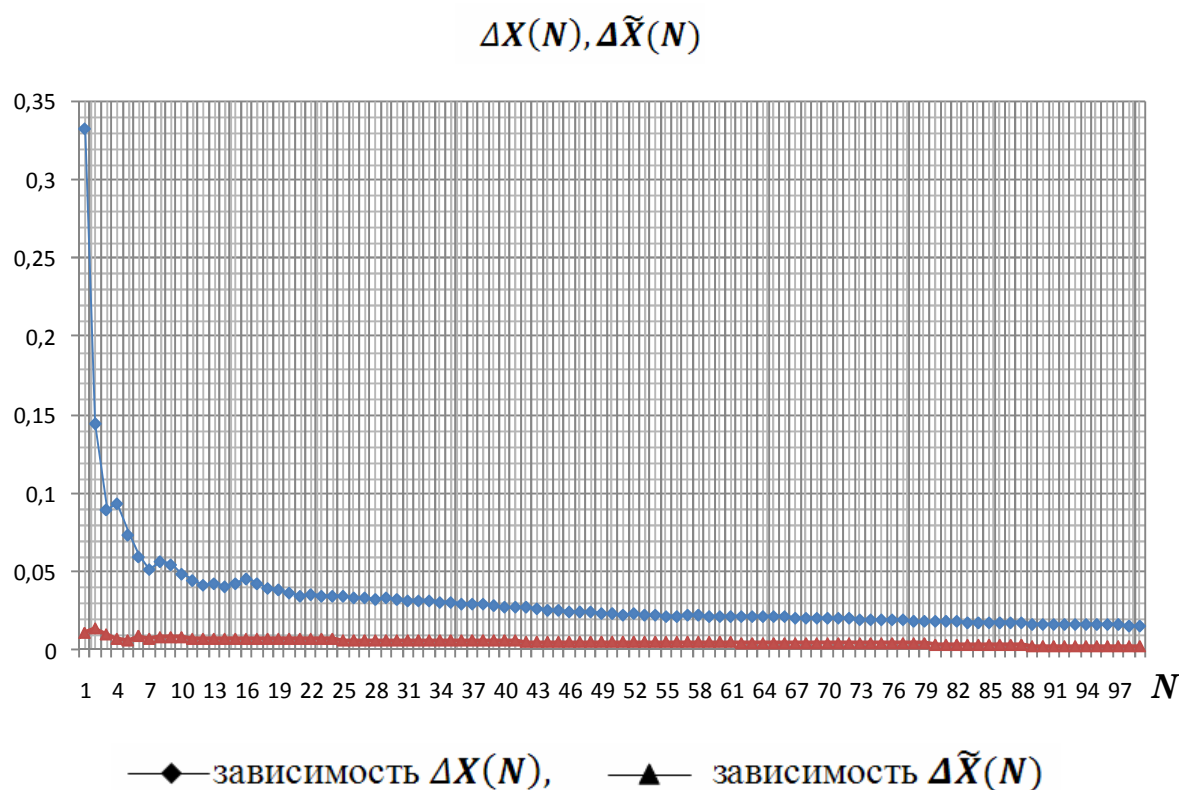


Рисунок 3 – Погрешности скользящего среднего исходных ΔX и
сглаженных $\Delta \tilde{X}$ значений как функции количества
измерений N ($\alpha = 0,1$)

Полученный результат оптимизации α совпадает с утверждением о том, что оптимальный диапазон параметра α находится в пределах $0,1 - 0,3$ [2,3] (утверждение относится к сглаживанию временных рядов).

Эффективность экспоненциального сглаживания в зависимости от степени неоднородности магнитного поля датчика была определена путем обработки числовых рядов, полученных от датчиков с заведомо малой неоднородностью поля (полная осевая симметрия) и с большой (квазиосевая симметрия), расположенных на шероховатой поверхности штуфа. Сглаживание проводилось при $\alpha = 0,1$ с последующим сравнением относительных погрешностей $\delta = \Delta X / X_{cp}$ из-за отличающихся собственных значений индуктивностей датчиков (см. таблицу 2).

Из сравнения данных таблицы 2 следует:

– для исходного ряда относительная погрешность от неоднородности поля у датчика с квазиосевой симметрией на порядок больше погрешности датчика

с полной осевой симметрией;

– отношение относительных погрешностей δ для датчиков с квазиосевой и осевой симметрией до сглаживания равно 12, после сглаживания – 4,4, следовательно, сглаживание уменьшает различие погрешностей по величине;

– для датчика с квазиосевой симметрией после сглаживания погрешность δ уменьшилась в 7,89 раза, а для датчика с осевой симметрией – в 2,84 раза, следовательно, сглаживание оказывает большее влияние на большую погрешность чем на меньшую.

Таблица 2

Погрешности датчиков с осевой и квазиосевой симметрией

Параметр результата измерения	Исходный ряд		Сглаженный ряд	
	Квазиосевая симметрия	Осевая симметрия	Квазиосевая симметрия	Осевая Симметрия
Действительное значение, X_{cp}	5,789	26,380	5,788	26,388
Абсолютная погрешность, ΔX	$\pm 0,016$	$\pm 0,006$	$\pm 0,002$	$\pm 0,002$
Относительная погрешность, δ	0,276%	0,0227%	0,035%	0,008%

Количественные результаты влияния мешающего воздействия шероховатости поверхности до и после сглаживания для датчика с осевой симметрией показаны в таблице 3.

Погрешности для датчика с осевой симметрией на шероховатой и
гладкой поверхностях

Параметр результата измерения	Исходный ряд		Сглаженный ряд	
	Шероховатая	Гладкая	Шероховатая	Гладкая
Действительное значение, X_{cp}	26,380	26,370	26,388	26,372
Абсолютная погрешность, ΔX	$\pm 0,006$	$\pm 0,006$	$\pm 0,002$	$\pm 0,002$
Относительная погрешность, δ	0,0227%	0,0228%	0,0076%	0,0076%

Из сравнения соответствующих параметров из таблицы 3 следует:

- на действительное значение результата измерения X_{cp} шероховатость поверхности оказывает влияние, превосходящее абсолютную погрешность ΔX , после сглаживания влияние шероховатости на X_{cp} практически не изменяется; следовательно, сглаживание не уменьшает разницу в значениях X_{cp} , вызванную влиянием шероховатости;

- сглаживание исходного ряда в три раза уменьшает погрешность результата без ощутимого влияния на ее величину шероховатости поверхности.

Выводы. Проведенные исследования эффективности экспоненциального сглаживания показали следующее:

- экспоненциальное сглаживание числовых рядов измерительных данных позволяет почти на порядок уменьшить погрешность результата измерений;

- в условиях влияния мешающих факторов сглаживание оказывает большее влияние на большую погрешность, чем на меньшую;

- сглаживание эффективно подавляет мешающее влияние неоднородности магнитного поля датчика в зоне контроля; влияние шероховатости контролируемой поверхности анизотропного вещества сглаживанием не уменьшается;

– алгоритм экспоненциального сглаживания может быть использован для предварительной обработки измерительной информации с целью повышения точности результата измерения;

– дальнейшее повышение эффективности сглаживания требует решения задачи нахождения критерия оптимизации параметра сглаживания.

ЛИТЕРАТУРА

1. Румшицкий Л.З. Математическая обработка результатов эксперимента. – М.: Наука, 1971. – 192 с.
2. Лукашин Ю.П. Адаптивные методы краткосрочного прогнозирования временных рядов. – М.: Финансы и статистика, 2003. – 416 с.
3. Четыркин Е.М. Статистические методы прогнозирования. – М.: Статистика, 1977. – 200 с.
4. Льюис К.Д. Методы прогнозирования экономических показателей. – М.: Финансы и статистика, 1986. – 130 с.
5. Грешилов А.А., Стакун В.А., Стакун А.А. Математические методы построения прогнозов. – М.: Радио и связь, 1997. – 112 с.

О.С. Волковский, Я.Ю. Петренко

**КОНВЕРТАЦИЯ МЕДИА ФАЙЛОВ НА ПРИМЕРЕ
АУДИО, В ОПЕРАЦИОННОЙ СИСТЕМЕ WINDOWS 8 ОС
ИСПОЛЬЗОВАНИЕМ WINDOWS STORE БИБЛИОТЕКИ.
СРАВНЕНИЕ С FFMPG**

Аннотация: В работе исследована непосредственная конвертация аудио файлов с использованием Windows Store библиотек, проведено сравнение с ffmpeg библиотекой. Написана тестовая программа, которая может быть использована для создания приложений по обработке медиа файлов. Проведено сравнение Windows Store библиотеки и ffmpeg библиотеки для определения рациональности использования ресурсов ПК при конвертации медиа.

Ключевые слова – конвертация, медиа, аудио, Windows Store, ffmpeg

Постановка проблемы: с выходом новой операционной системы Windows 8 пришло такое понятие как Metro приложения, которые могут быть запущены как на стационарном ПК так и на различных мобильных устройствах(планшетах, мобильных телефонах). Новые технологии требуют написания широко используемых программ под свою платформу, что приводит в свою очередь к необходимости изучения и использования новых библиотек и фреймворков. В начале разработки нового продукта нужно исследовать насколько технология будет рациональна в использовании для данной цели, какая из платформ будет использована и насколько это будет эффективно.

Анализ последних исследований: Windows 8 представила новый тип приложений: Windows Store приложения – это приложения в стиле Metro. Windows Store приложения имеют совершенно новый внешний вид, они работают на различных устройствах.

Можно разрабатывать Windows Store приложения на множестве языков:

Если есть знания технологий веб-разработки, можно разрабатывать Windows Store приложения используя HTML5, Cascading Style Sheets, Level 3 (CSS3), и JavaScript.

Если есть опыт в .NET, Windows Presentation Foundation, или Microsoft Silverlight приложениях, можно разрабатывать Windows Store приложения используя XAML, с кодом на C++, C#, или Visual Basic.

Если же есть знания DirectX, то можно разрабатывать DirectX Windows Store приложения используя C++ и HLSL, что бы в полной мере воспользоваться графический оборудованием.

Формулировка целей работы: Целью данной работы является разработка программы в ОС Windows 8, на платформе .NET с использованием Windows Store библиотек, которую в последующем можно взять за основу для написания программного продукта конвертации медиа файлов. Сравнение производительности предложенного метода конвертации медиа с уже существующим.

Основная часть: Рассмотрим конвертацию цифровых медиа файлов, таких как аудиофайлы, из одного формата в другой. Для начала рассмотрим аудио форматы, с которыми позволяет работать Windows Store библиотека.

Таблица 1

Аудио форматы [1]

Медиа контейнер или формат файла	Расширение файла	Форматы аудио потока (Кодеки)
MPEG-4	.3g2	AAC (LC, HE)
	.3gp2	
	.3gp	
	.3gpp	
	.m4a	AAC (LC, HE) MP3 AC3 (DD, DD+)
	.m4v	
	.mp4v	
	.mp4	
	.mov	
MPEG-2	.m2ts	MPEG-2 (L1, L2, stereo-only), MPEG-1 (L1, L2), AAC (LC, HE), AC3 (DD, DD+)
ASF	.asf	WMA standard
	.wm	WMA voice
	.wmv	WMA lossless
	.wma	WMA Pro, AC3 (DD, DD+)
ADTS	.aac	AAC (LC, HE)
	.adt	
	.adts	
MP3	.mp3	MP3
WAV	.wav	PCM, MP3, MS ADPCM, IMA ADPCM, MS DDIT G.711, MS GSM 6.10, AC3 (DD, DD+)

AVI	.avi	PCM, MP3, MS ADPCM, IMA ADPCM, MS CCITT G.711, MS GSM 6.10 AC3 (DD, DD+)
AC-3	.ac3 .ec3	AC3 (DD, DD+)

Немаловажной особенностью Windows Store библиотеки является асинхронное программирование, что принципиально отличает реализацию с ее помощью и дает этому способу преимущество. Типичный сегмент кода, написанный на C# или Microsoft Visual Basic, выполняется синхронно, это значит, что выполнение строки заканчивается до начала выполнения следующей. Различные программные модели, такие как Модель асинхронного программирования и Асинхронный паттерн, используются долгое время, и позволяют разработчикам писать асинхронный код. Пока эти модели успешно реализовывали асинхронность, результирующий код сосредотачивался именно на ней, в ущерб выполняемой задаче. Это сделало асинхронный код тяжелым для написания, понимания и поддержки. В Windows Runtime, .NET фреймворк, C# и Visual Basic в компиляторы были добавлены функции, которые позволяли абстрагировать асинхронный механизм от кода, чтобы можно было писать асинхронный код, который сосредотачивается на том что делает код, а не как он это делает. Таким образом можно писать асинхронный код который очень похож на синхронный.

Использование асинхронного программирования помогает приложениям оставаться активными, когда он выполняет работу, занимающую длительный период времени. Например, приложение, которое получает информацию из Интернета, может провести несколько секунд в ее ожидании. Если использовать синхронный метод в потоке пользовательского интерфейса для получения данных, приложение будет заблокировано, пока метод не вернет что-либо. Приложение не будет в таком случае отвечать на действия пользователя. Намного лучше использовать асинхронное программирование, когда приложение продолжает быть активным и отвечать на пользовательский интерфейс, пока оно ждет окончания операции.

Имена асинхронных функций заканчиваются с "Async" чтобы указать, что часть их выполнения может продолжаться после возвращения функции. При использовании асинхронных функций в Windows Store приложениях, код осуществляет неблокирующие вы-

зовы постоянно. При реализации этих асинхронных моделей в функциях код будет понятен и использован предсказуемым образом [2].

Типичные задачи, которые требуют вызов Windows Runtime функций:

- отображение диалогового окна сообщения, - работа с файловой системой, - отправка и получение данных из Интернета.

Используя асинхронные API среды выполнения Windows, не нужно управлять потоками явным образом и напрямую обращаться к базовым функциям. [1]. Рассмотрим простой метод перекодирования аудио файлов. В начале примера используем MP3 и ASF форматы.

Для начала реализуем функцию открытия файла и опишем все его особенности. Что бы открыть аудио файл из системы нам нужно использовать класс `FileOpenPicker` [3] . Алгоритм открытия аудио файла (рис. 1):

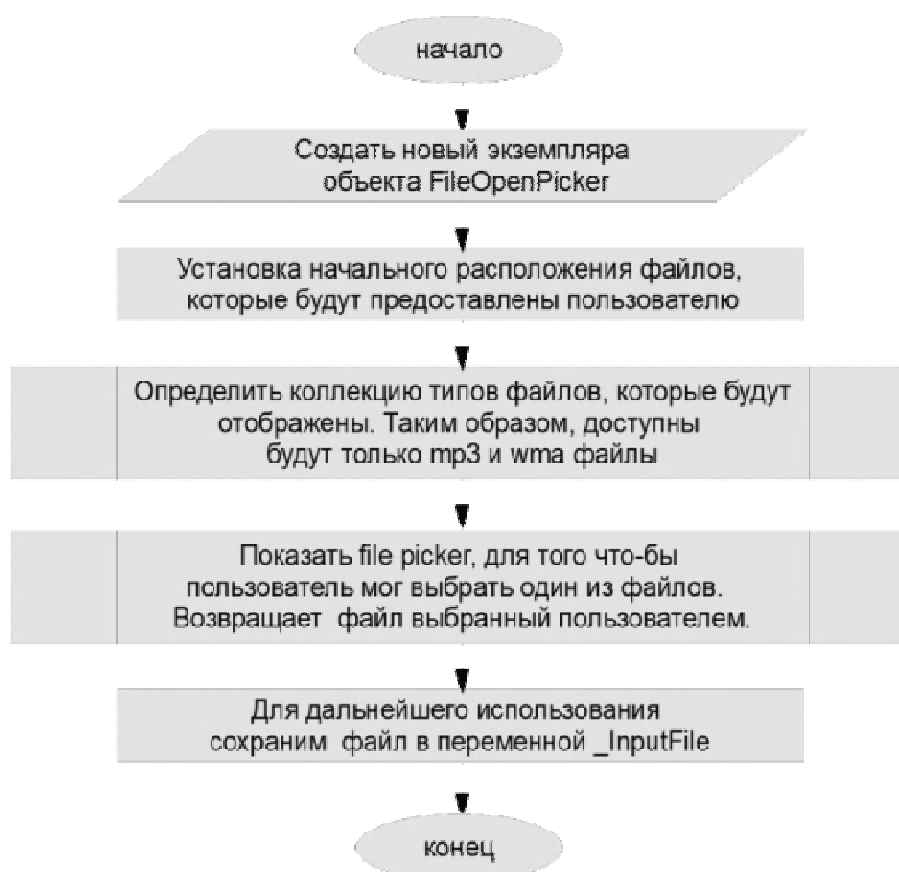


Рисунок 1 – Блок – схема алгоритма открытия аудио файла

Теперь перейдем к функции перекодирования аудио. Алгоритм функции перекодирования аудио (рис. 2):



Рисунок 2 – Блок – схема алгоритма перекодирования аудио

Проанализируем полученные результаты, тестирования медиа конвертации с учетом использования следующей аппаратной конфигурации:

Таблица 2

Аппаратная конфигурация

ОС	Windows 8 Pro
Процессор	Intel(R) Core(TM) i3 CPU M380 @ 2.53GHz
Установленная память (RAM)	3,00GB
Тип системы	64-bit Operating System, x64-based processor

Конвертация с использованием Windows Store библиотеки из mp3 в wma привела к результатам, представленным на рис.3

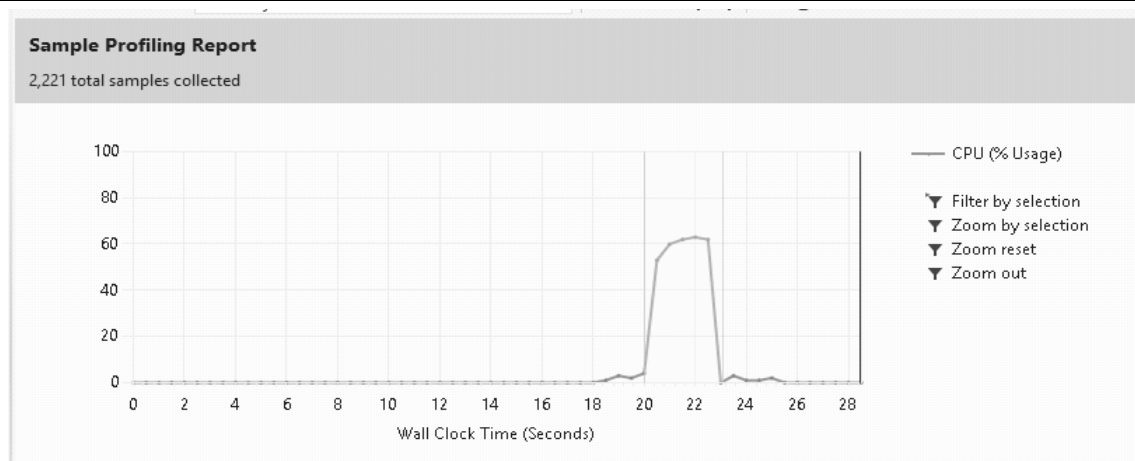


Рисунок 3 - Конвертация из mp3 в wma. Загрузка процессора

Из графика видно, что загрузка процессора находилась на уровне 60%, а время конвертации заняло 3 секунды.

Использование Windows Store библиотеки для конвертации из wma в mp3 привело к следующим результатам (Рис. 4).



Ри

сунк 4 - Конвертация wma в mp3 . Загрузка процессора

В этом случае загрузка процессора составила 55%, а время конвертации около 3 секунд.

Для сравнения полученных результатов была также разработана тестовая программа для конвертации с использованием ffmpeg, которая, в свою очередь, показала следующие результаты (рис.5).

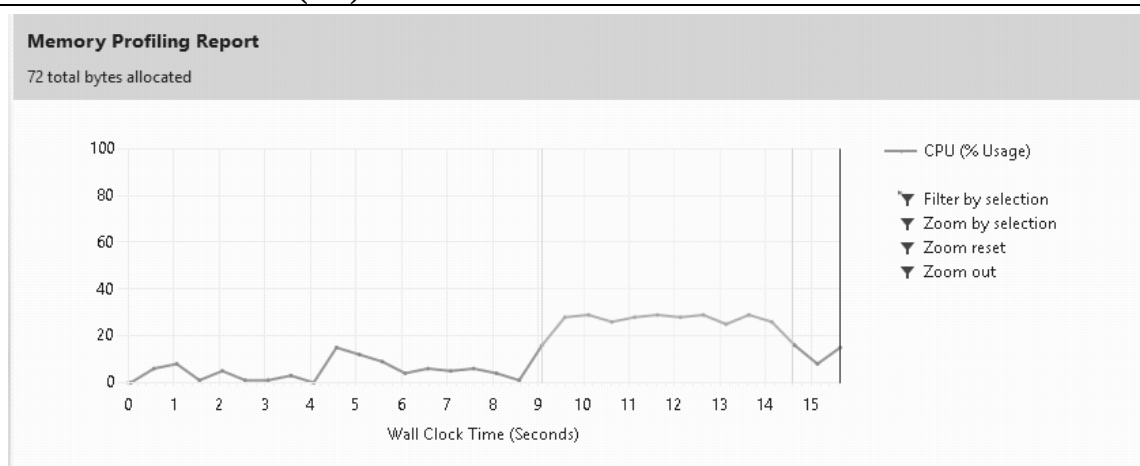


Рисунок 5 - Конвертация wma в mp3. Загрузка процессора

Анализируя график из рис.5 можно увидеть, что загрузка процессора снизилась до 30%, при этом время конвертации возросло до 6 секунд.

Конвертация mp3 в wma с использованием ffmpeg привела к ошибке, таким образом файл не удалось конвертировать с помощью данной библиотеки (рис.6).

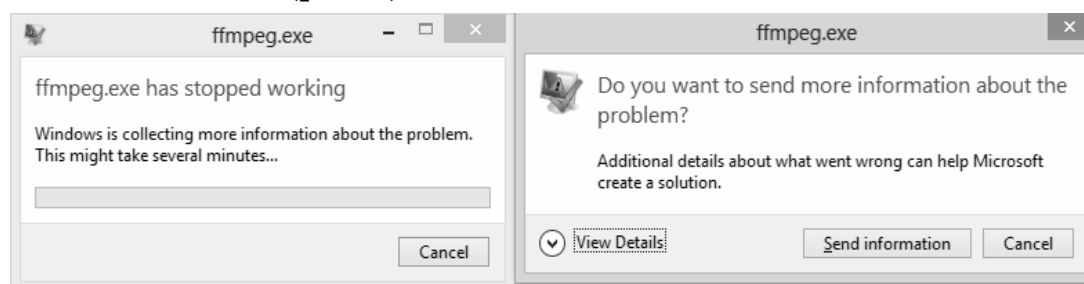


Рисунок 6 - Конвертация из mp3 в wma.

Ошибка во время выполнения

Сравним полученные данные:

Таблица 3

Сравнение данных

Конвертация	Windows Store биб- лиотека	ffmpeg библиотека
	загрузка процессора	время конвертации
из mp3 в wma	60%	3 с.
из wma в mp3	55%	3 с.

Из таблицы видно, что загрузка процессора для библиотеки `ffmpeg` значительно меньше в случае конвертации из `wma` в `mp3`, но время при этом в два раза больше. Учитывая современные технологии загрузка процессора не так критична как время выполнения, таким образом в этом случае можно отдать предпочтение `Windows Store` библиотеке.

Для варианта конвертация из `mp3` в `wma` безусловно лучше использовать библиотеку `Windows Store`, так как `ffmpeg` библиотека не показала положительных результатов для этого случая.

Если же сравнивать эти две библиотеки в перекодировке `lossless` форматов, то сделать это можно лишь с использованием `ffmpeg`, `Windows Store` библиотека в этом случае выдает неизвестную ошибку (рис. 7).

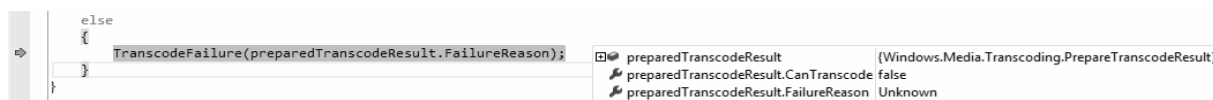


Рисунок 7 - Ошибка при перекодировке `lossless` форматов.
Windows Store

Выводы: Как видно из проведенных исследований `Windows Store` библиотеки лучше для работы с `Windows media formats`, таких как `wma` или `wav`. В случае использования `ffmpeg` для этих целей, могут возникнуть некоторые трудности при перекодировке этих форматов.

Если нужно перекодировать `lossless` форматы (напр. `flac` или `ogg`) – есть возможность сделать это только с использованием `ffmpeg`. `Windows Store` не поддерживают `lossless` форматы.

ЛИТЕРАТУРА

1. <http://msdn.microsoft.com/ru-ru/windows/apps/>.
2. Лутай С., Байдачный С. «Windows 8 для C# разработчиков», Издательство: Самиздат, 2011г., 278с
3. С. В. Пугачев, А. М. Шериев, К. А. Кичинский «Разработка приложений для Windows 8 на языке C#» — СПб.: БХВ-Петербург, 2013. — 416 с.: ил. —(Профессиональное программирование)

А.А. Литвинов, М.В. Павленко, А.А. Матвеев

ТЕХНОЛОГИЯ АНАЛИЗА РАБОТЫ КЛИНИКИ ХИРУРГИЧЕСКОГО ПРОФИЛЯ

Аннотация. В работе рассматривается вариант технологии (методов, средств) информационного анализа работы клиники хирургического профиля, которая базируется на специальном программном обеспечении поддержки работы менеджера клиники по вопросам планирования активностей и ресурсов, врача-эксперта по вопросам возможной корректировки существующих стандартов. Рассмотренная технология может быть использована для создания системы поддержки принятия решений по управлению клиникой хирургического профиля.

Ключевые слова: стандарты, планирование, расписание, информационная система, хирургическое отделение, алгоритм.

Введение. Усовершенствование организации и повышение качества услуг медицинской помощи, интенсификация труда врачей поликлиник и стационаров являются актуальной задачей учреждений здравоохранения. Известно, что на качество лечения влияет много факторов, в частности, своевременное и полное обследование больных, взаимосвязь между этапами оказания медицинской помощи, контроль деятельности медперсонала, внедрение новых методов диагностики и лечения. Не менее важным в улучшении диагностико-лечебного процесса является его стандартизация. Большое значение для стандартизации лечения имеют протоколы предоставления медицинской помощи, которые начали более активно внедряться в последние годы [1].

Одной из проблем разработки стандартов является создание описания врачом-экспертом протоколов, которые отражают специфику услуг для клинического состояния [2, 3], учитывая при этом множество различных параметров, связанных с пациентом (напр.: вес, реакция на препараты и др.) и влияющих на обязательность выполнения тех или иных лечебных/диагностических активностей. В результате протокол представляет последовательность активностей, раз-

битых на фазы-этапы лечения (напр., для хирургических больных – предоперационный этап, операционный, интенсивная терапия, послеоперационное лечение), связанных с 5-ю основными классами ресурсов: медикаменты, материалы, койки, операционное время, исполнители.

Роль таких протоколов чрезвычайно важна для формирования плана лечения больного, планирования активностей и ресурсов, планирования работы клиники, оценки стоимости и качества лечения, построения страховой медицины [4]. Следует отметить, что решение по применению тех или иных активностей всегда остается за лечащим врачом и с этой точки зрения протокол может расцениваться как база для оценки эффективности его работы. Важным аспектом стандарт-ориентированного подхода является обязательные фазы мониторинга и оптимизации стандартов, которые тесно связаны с информационными технологиями [5]. Таким образом, построение информационной технологии, направленной на мониторинг работы клиники по заданным стандартам, выделение протоколов и их оптимизация является важным вопросом медицинской информатики.

Постановка задачи. В работе рассматривается вариант технологии (методов, средств) анализа работы клиники хирургического профиля, которая базируется на специальном программном обеспечении поддержки работы менеджера клиники по вопросам планирования активностей и ресурсов, врача-эксперта по вопросам возможной корректировки существующих стандартов.

Основная часть. Понятие протокола, которое рассматривается в данной работе, связано в первую очередь с тремя основными концептами: клиническим состоянием (клиническим диагнозом), операцией и ее характеристиками (тип, продолжительность, сложность), пребыванием на койке определённого типа. Следует отметить, что данное понятие соответствует понятию «потока работ» (абстракция, описывающая структуру и порядок активностей, временные и пространственные ограничения, использование людских и материальных ресурсов для достижения цели, которая имеет ценность для клиента) [5]. Основными фазами потока работ оперативного лечения являются: подготовка к операции, проведение операции, реанимационная и интенсивная терапия, послеоперационное лечение и выписка. Каждая фаза потока имеет четкие временные границы и характеризуется

используемыми ресурсами (койко-дни, время занятия операционной или процедурного кабинета, медикаменты, материалы), ответственными исполнителями и целью (табл.1). Иными словами, протокол представляет информацию о типе операции, времени необходимом на выполнение операции, количестве дней до- и после- операционного лечения, процедурах, материалах и медикаментах. Например, время на выполнение заданной операции не может превышать заданного временного интервала, при этом больной должен быть госпитализирован за указанное в протоколе количество дней до операции, подготовлен к операции с использованием процедур и медикаментов, и быть выписанным из клиники через указанное количество дней после операции.

Таблица 1

Связь основных типов ресурсов с фазами потока работ

Фаза	Ресурсы	Ответственный	Цель
Предоперационная	Коечный фонд, медикаменты, оборудование, др.	Лечащий врач, Обслуживающий персонал	Подготовка к операции
Операция	Операционная, медикаменты, материалы, оборудование и др.	Операционная бригада	Проведение операции
Интенсивная терапия	Реанимационный коечный фонд, медикаменты, материалы и др.	Обслуживающий персонал	Переход к послеоперационному лечению
Послеоперационная	Коечный фонд, медикаменты, оборудование и др.	Лечащий врач, обслуживающий персонал	Выздоровление и выписка пациента

Специфика системы, ориентированной на стандарты, состоит в определении потока работ, отвечающего заданному протоколу, и фиксации необходимой информации, связанной с выполнением потока работ для контроля выполнения и оптимизации стандартов. От эффективной методики анализа потоков работ зависит эффективность работы клиники. Для этого определяется цель и стратегия анализа.

Цель:

- Предсказание возможных рисков, простоев коек и, как следствие, неэффективной работы клиники, осложнений операции и лечения пациента, задержки его лечения по стандарту. Результатом может быть коррекция стандарта или принятие решений относительно организации клиники.

- Определение пика поступлений больных по годам, месяцам, дням и, как

следствие, принятие решения о перераспределении ресурсов, включая привлечение дополнительных ресурсов (коек, операционных) на пиках нагрузки.

Стратегия:

Поскольку возраст человека сильно влияет на его здоровье и, соответственно, сказывается на появлении различных заболеваний, разбиение всего множества пациентов на возрастные группы и оценка их является первым шагом анализа работы клиники. Исследования данных на определенных временных интервалах способствует планированию распределения необходимых ресурсов. В самих группах выполняется разбиение по диагнозу и видам операций, в которых возможны отклонения от стандарта. Для более эффективного анализа проводится фильтрацию групп по дополнительным параметрам. Каждый этап имею свою задачу (табл.2).

Таблица 2

Основные этапы анализа информации

Этап	Цель	Временной срез		
Анализ возрастных групп	Оценка контингента поступающих пациентов Оценка работы клиники	Год	Месяц	Неделя
Анализ клинических состояний				
Анализ больных по видам, классам операций				
Анализ отклонений от стандарта	Оценка стандартов			

Так, анализ распределения 16 возрастных групп (рис.1) дает возможность спрогнозировать работу клиники на базе поступивших пациентов за прошлые годы. Как видно из графика с каждым годом

происходит увеличение количества услуг, при этом основным контингентом являются люди из группы «25-35».

Далее, каждая группа пациентов анализируются в разрезе месяцев (рис. 2). Такая оценка может влиять на прогнозирование отклонений, оценку качества лечения и ресурсные затраты на лечение (количество мест и др.).

Из графика видно, что минимум пациентов поступает в мае, августе. Пик поступления пациентов - ноябрь. Важно отметить также, что в 2011 году были убраны «крутые» спады, присутствующие в предыдущие годы. Это произошло из-за проведения дополнительных мероприятий по привлечению пациентов в клинику. В случае устойчивой тенденции возможны решения по временному сокращению ресурсов клиники - отказ от аренды дополнительной операционной, ведение семинаров для врачей. И, наоборот, в «пиковые» периоды возможно планирование дополнительных ресурсов (аренда операционной и коек и пр.)

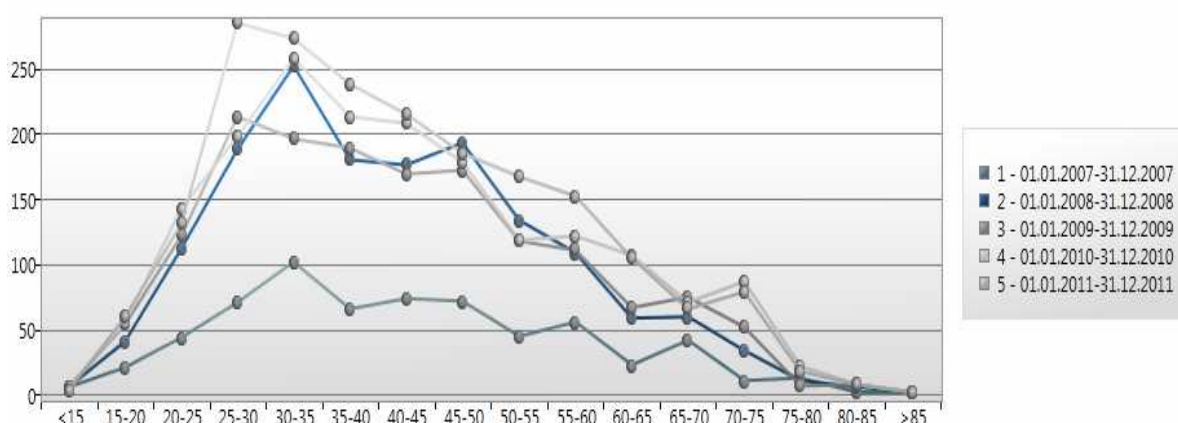


Рисунок 1 - Распределение больных по годам и возрастным группам

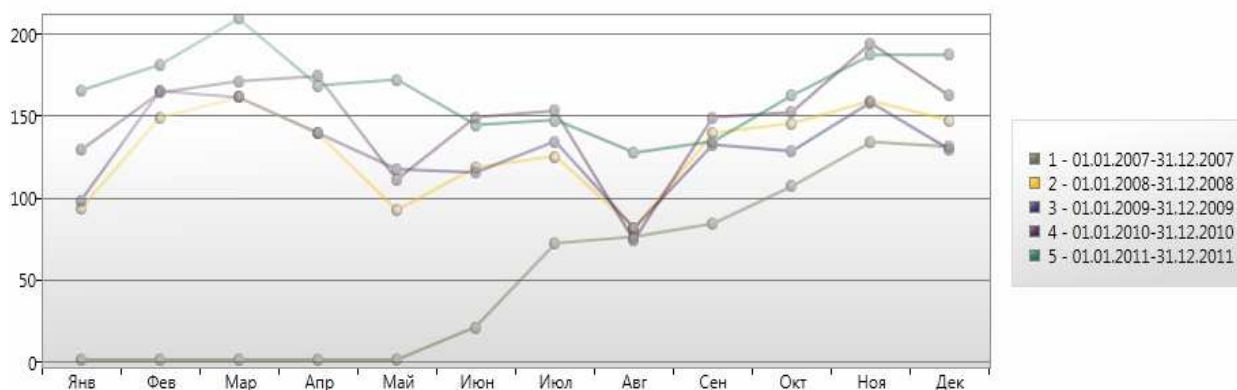


Рисунок 2 - Распределение больных по месяцам на протяжении исследуемых лет

Распределение фактов поступления и выписки в рамках недели (рис. 3,4) дает информацию для оценки правильности распределения нагрузки в рамках недели, что позволяет исключить простои койки, сгладить нагрузку на операционную. Согласно полученным данным, пиковыми днями поступления являются вторник и среда, а днями выписки - пятница и суббота, что приводит к неэффективному использованию коечного фонда.

Описанные этапы исследования групп детализируются с помощью дополнительных параметров. Это - пол, статус пациента (поступивший/выписавшийся), код диагноза (включая классы заболеваний и клинические состояния), операции: вид операции, тип операции, наличие симультанных операций, отклонения операции, отклонения в количестве койко-дней.

Особую роль при этом играет оценка работы клиники по классам заболеваний, группам и отдельным клиническим состояниям, видам и типам операций с поиском отклонений от стандартов.

Главная форма программы анализа медицинских данных представлена на рис. 5.

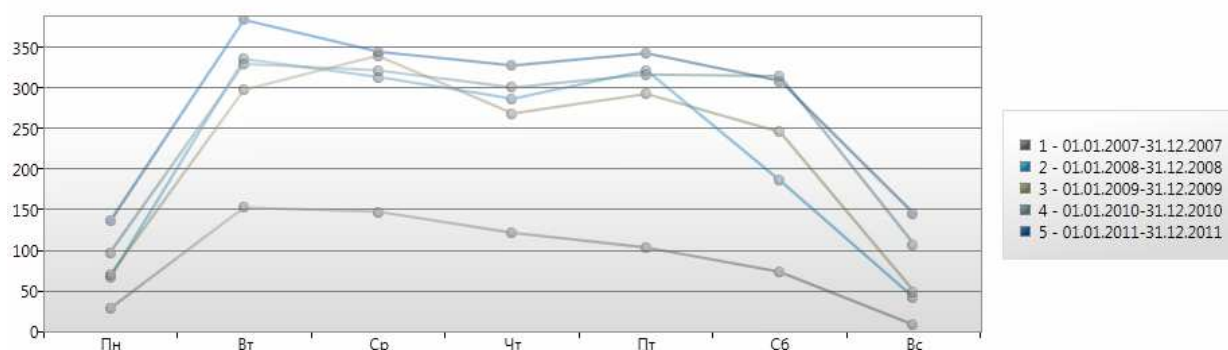


Рисунок 3 - Распределение поступивших пациентов по дням недели

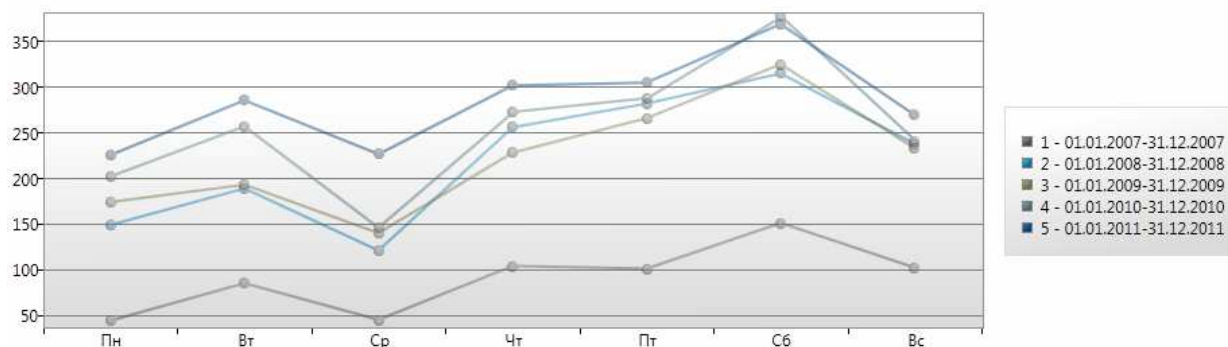


Рисунок 4 - Распределение выписавшихся пациенты по дням недели

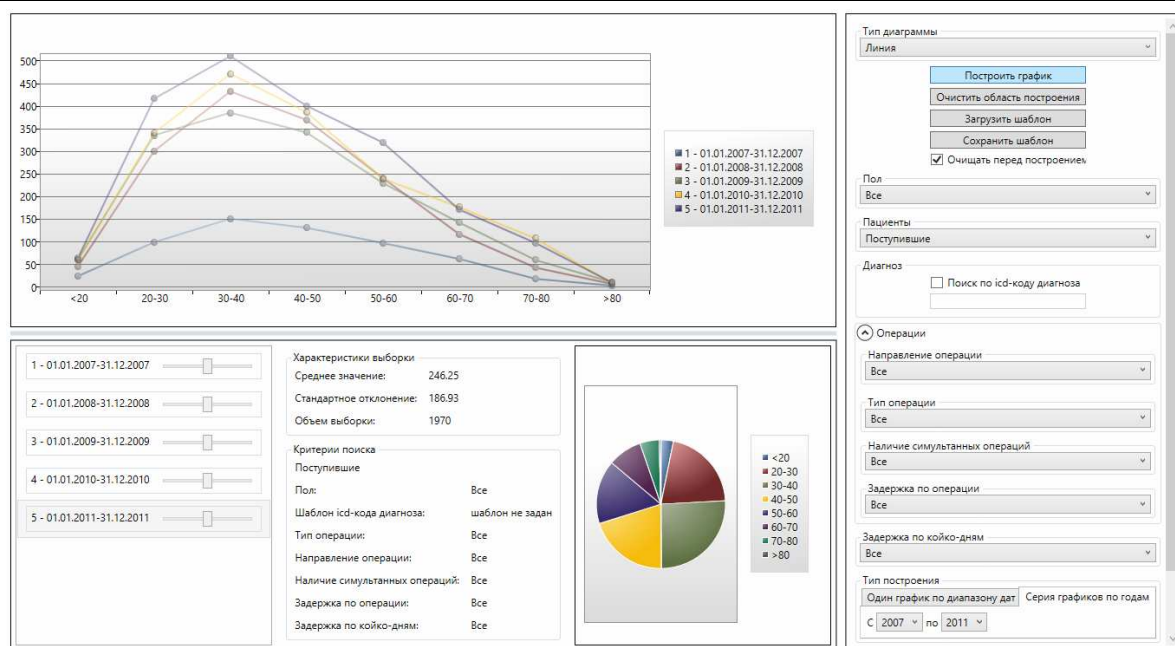


Рисунок 5 - Главное окно программы

Для облегчения повторного использования запросов со сложной комбинацией параметров был введен механизм шаблонов. Каждый шаблон выборки сохраняется в отдельный файл и далее может быть использован для ручного или автоматизированного поиска отклонений, анализа результатов.

Важной функциональной особенностью представленного решения является возможность визуализации, и, соответственно, оценки нескольких результатов, сформированных на базе разных шаблонов запросов. Например, оценка доли урологических операций от общего числа операций.

Важным техническим аспектом решения является применение динамически формируемых лямбда-выражений для обработки в памяти сравнительно небольшого количества объектов (до 100 тысяч объектов), что дает выигрыш в сравнении с СУБД на большом количестве запросов.

Выводы. Рассмотрен порядок оценки работы клиники по выбранным показателям и метод оценки, при использовании которого возможно формирование маркетинговых, управленческих решений, оценки существующих стандартов. Рассмотрены особенности работы системы анализа медицинских данных для поддержки планирования ресурсов медицинского учреждения на примере клиники хирургического профиля. Рассмотренный вариант может быть использован в

качестве платформы для создания системы поддержки принятия решений по управлению клиникой хирургического профиля.

ЛИТЕРАТУРА

1. Березницкий Я.С., Бойко В.С., Брусницина М.П. и др. Клинические рекомендации для врачей по вопросам организации и оказания медицинской помощи больным с острыми хирургическими заболеваниями органов живота (ведомственная инструкция). – Киев – 2004, – 353 с.
2. Дзяк Г.В., Березницкий Я.С., Филиппов Ю.А. и др. Библиотека практического врача. Унифицированные клиничко-статистические классификации болезней органов пищеварения (ведомственная инструкция). – Киев, 2004. – 93 с.
3. Литвинов О. А. Формалізація клінічного діагнозу і модель електронної класифікації діагнозів для Медичних Інформаційних Систем України. // Системные технологии. Рег. межвуз. сб. научн. работ. – Выпуск 2(49). – Днепропетровск, 2007. - С.46-56.
4. В.Н. Лехан, М.В.Павленко, А.А. Литвинов. Подход к автоматизации планирования лечения больных хирургического профиля. // Системные технологии. Рег. межвуз. сб. научн. работ. – Выпуск 5(64). – Днепропетровск, 2007. - 89-102 с.
5. Wil van der Aalst, K.M. van HeeM. Workflow Management: Models, Methods, and Systems. IT press, Cambridge, MA, 2002. – 384 p.

А.А. Литвинов, А.С. Вякилов, А.И. Брежнев

ОСОБЕННОСТИ ПОСТРОЕНИЯ МЕДИЦИНСКОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ОРИЕНТИРОВАННОЙ НА СТАНДАРТЫ

Аннотация. В данной работе рассмотрен вариант разработки гибкой информационной системы поддержки стандартов. Результатом работы является подход генерации стандартов на основе существующей практики лечения, что позволяет обеспечить гибкость медицинской информационной системы и является неотъемлемой частью систем контроля качества и стоимости лечения.

Ключевые слова: стандарт, протокол, информационная система, сервис.

Введение и актуальность темы. В настоящее время в области здравоохранения организуются работы по стандартизации, с целью повышения качества медицинской помощи. Высокое качество такой помощи невозможно без современных технологий, которые базируются на стандартах диагностики и лечения, преследующих цель использования наиболее эффективным и экономически выгодным способом [1]. Таким образом, актуальными вопросами медицинской информатики являются вопросы построения информационных технологий поддержки (построения, выполнения, контроля) стандартов медицинской помощи, без которых практически невозможен качественный контроль выполнения медико-экономического стандарта [2]. Внедрение стандартов, основанных на эвристических и статистических оценках опыта успешного лечения клинических состояний, на базе информационных технологий позволяет качественно повысить уровень мониторинга, планирования активностей и ресурсов, оптимизировать бизнес-процессы.

Постановка задачи и краткий обзор существующих методов. Построение систем направленных на поддержку стандартов связано с решением ряда проблем: способ получения протоколов, структура и подробность описания протоколов, учет отклонений и альтернатив, учет аналогов медикаментов, адаптация системы к динамике измене-

ний бизнес-процессов, учет специфики и организации клиники, интеграция компонентов информационной системы.

Путь построения системы, предложенный в работе [2] и основанный на описании протокола экспертом-медиком с его последующей трансформацией в поток работ WF с динамической связью с подсистемами, не является эффективным по ряду причин, основными из которых являются: жесткость задания порядка активностей (в реальной жизни выбор активностей является прерогативой лечащего врача, стандарт играет вспомогательную роль); сложность описания протокола, учета всех альтернатив, аналогов материалов и медикаментов; сложность выделения клинических случаев, с которыми связываются комплексы активностей, учитывающих кроме клинического диагноза также ряд дополнительных характеристик, влияющих на обязательность активностей.

Учитывая множество проблем, связанных с постарением и внедрением такой системы, в данной работе рассматривается иной подход, основанный на динамическом построении стандартов без жесткой структуры потока работ, связанного с осуществлением протокола.

Основная часть. Протокол лечения пациента легко представить как множество активностей (не всегда последовательных), которое разбито на определенное количество фаз. Например, предоперационная, операционная, интенсивная терапия, постоперационная. Фазы в свою очередь состоят из потенциально параллельных потоков активностей (с точки зрения информационной модели, а не реального хода событий), которые связаны с 5-ю основными типами ресурсов: медикаменты, материалы, койки, операционное время и исполнители. На рис.1 представлена общая модель протокола и его выполнение. Протокол состоит из распределенных во времени активностей. Прямоугольники с буквами означают тип и временные рамки выполнения активностей заданного типа (М – медикаментозное лечение, отраженное листом назначений; ТР – лечебная процедура; О – операция; А – лабораторное исследование (напр., общий анализ крови); ДР – диагностическая процедура). Вертикальные прямоугольники, объединяющие группы активностей – фазы. Горизонтальные – 5 основных типов: медикаментозное лечение, лечебные процедуры, операция, лабораторное исследование, диагностические процедуры.

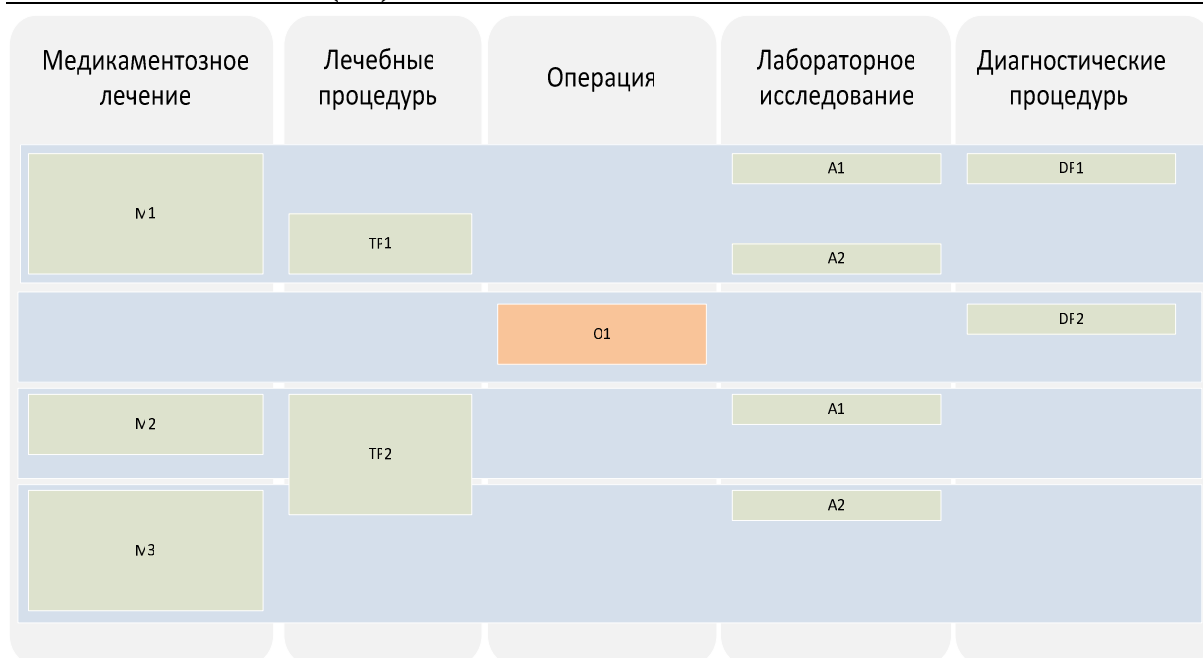


Рисунок 1 - Общая модель выполнения протокола

В целом, данная модель описывает следующее. После определения клинического диагноза [3] больного, лечащим врачом планируется набор активностей. В ходе планирования лечащий врач опирается на протокол лечения, связанный с клиническим состоянием (пациент возрастной группы «50-60 лет» с клиническим диагнозом «язва желудка осложненная кровотечением ... степени тяжести» с весом ...). Лечащий врач может изменять стандартный набор активностей, представленных протоколом, при этом возможно согласование введения таких изменений с контролирующей инстанцией (напр., зав. отделением). В идеале, при правильном построении протокола и отсутствии осложнений лечения, работа лечащего врача сводится к подтверждению планирования и выполнения активностей, предусмотренных протоколом. Планирование активностей может осуществляться и в рамках фазы и по фазам, для этого необходимо согласование их со структурами, отвечающими за их осуществление, которые связаны с соответствующими информационными подсистемами-компонентами. При этом необходимо предусмотреть отмену запланированных активностей и, естественно, вести мониторинг процесса изменений.

Концептуальная модель информационной системы обеспечивающей поддержку подобного процесса представлена на рис.2.

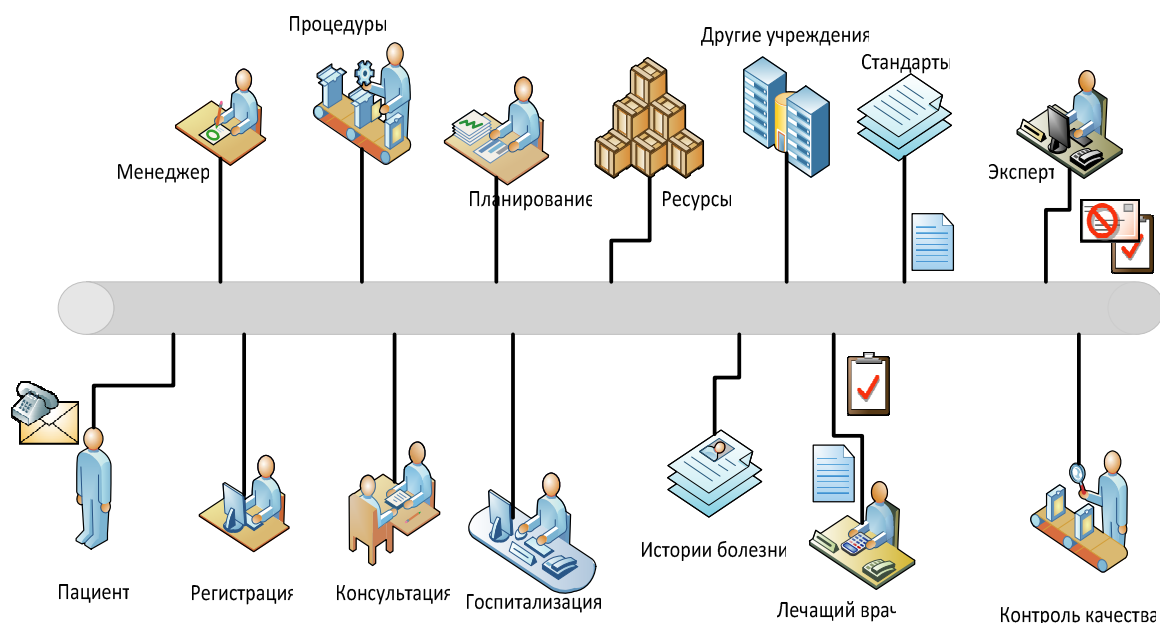


Рисунок - Концептуальная модель системы

Все компоненты бизнес-процесса представлены своими информационными отражениями – программными службами-компонентами, которые взаимодействуют через слой интеграции (middleware), при этом политика связи этих компонентов может быть изменена в любой момент без особых усилий. Взаимодействие подсистем осуществляется путем обмена сообщениями. Передача сообщений контролируется слоем интеграции, который кроме маршрутизации и политики отвечает за ведение истории передачи сообщений, создавая информационную базу для фаз мониторинга и оптимизации бизнес-процессов. Таким образом, изменение системы, подключение новых подсистем не требует дополнительных временных или финансовых затрат: определяется тип сообщений, за обработку которых отвечает данный компонент; маршрут и способ доступа. Важной особенностью при этом является возможное сохранение на стороне клиентов уже существующее программное обеспечение за счет введения механизмов преобразования стандартных сообщений в структуры и функции принятые внутри системы.

Данные особенности позволяют говорить об использовании сервисно-ориентированного подхода [4] и архитектуре «шина событий» (event-bus) для разработки системы [5]. В основе лежит идея использования распределенных, слабо связанных заменяемых компонентов, оснащенных стандартизированными интерфейсами для взаимодейст-

вия по стандартизированным протоколам. В данном случае под стандартом обмена сообщениями понимается внутренний стандарт системы, согласно которому все активности и типы обработчиков классифицируются.

Применение архитектуры шины позволяет исключить зависимость между компонентами системы. Логическая модель системы показана на рис. 3.

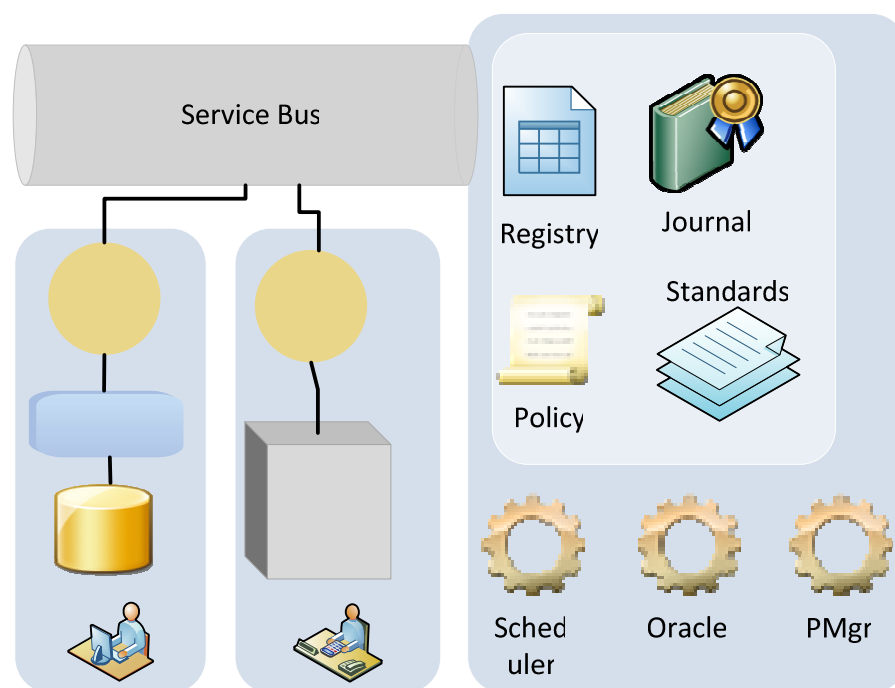


Рисунок 3 - Логическая модель системы

Работа интеграционного слоя основана на трех компонентах: диспетчер (Scheduler), планировщик (Oracle), менеджер протоколов (PMgr). “Scheduler”, главным образом занимается перенаправлением сообщений, регистрацией компонентов, отвечает за выполнение политики пересылки сообщений, ведет лог. “Oracle” отвечает за анализ возможности назначения и выполнения списка активностей протокола, для этого он опрашивает все возможные компоненты-исполнители, отвечающие за выполнение активностей заданного типа, и анализирует полученные варианты, с возможным выбором оптимального. Результат его работы – план выполнения активностей, который подтверждается лечащим-врачом. За анализ клинического состояния (клинического диагноза с дополнительными характеристиками, описывающими состояние пациента при поступлении) и его связь с протоколом отвечает компонент менеджера протокола.

Важным компонентом системы является «АРМ лечащего врача», работа которого состоит из следующих шагов: определение клинического диагноза; запрос протокола у менеджера протоколов. Если такой протокол существует, то согласно ему и специфике клинического случая, формируется план лечения, при этом врач может отметить все активности протокола, назначив свои, что далее анализируется подсистемами пересмотра стандартов и контроля качества. Созданный план проверяется на возможность выполнения компонентом планирования и при успешном планировании происходит назначение активностей на компоненты-исполнители. Стоит отметить, что врач может изменять план в процессе исполнения: удалять запланированные активности, добавлять новые.

По причине того, что невозможно предопределить изначально все возможные протоколы, система направлена на постепенное их введение и эволюцию. Изначально введение в эксплуатацию осуществляется с ограниченным набором протоколов, которые заданы экспертом, либо с построены с использованием уже существующих статистических данных, с последующим переходом на подход ориентированный на стандарты. Протокол имеет ряд обязательных и необязательных активностей разбитых на фазы. В случае отсутствия протокола (или даже стандарта) лечащий врач строит план лечения самостоятельно, сохраняя формат описания лечения согласно протоколу (формат задания и типы активностей, привязанность к фазам). По мере накопления результатов лечения предполагается генерация протоколов, которые в дальнейшем проходят стадии: анализа, апробации, утверждения и внедрения.

Вывод. В данной работе рассмотрена модель медицинской информационно-системы с поддержкой медицинских стандартов. Предлагаемый вариант построения системы позволяет масштабировать и расширять систему, за счет сервис-ориентированного и централизованного подхода, дополнять новыми сервисами-исполнителями не внося изменений в работу системы, генерировать стандарты при лечении и обеспечивает возможность контроля выполнения стандартов.

ЛИТЕРАТУРА

1. Брежнев А.И., Литвинов А.А., Павленко М.В. "Основные требования к системе управления потоками работ для оценки качества и стоимости лечения". - Донецк ДНТУ 2010, Інформаційні управляючі системи, та комп'ютерний моніторинг, Збірка матеріалів І всеукраїнської науково-технічної конференції студентів, аспірантів та молодих вчених - том 1 с. 38-39.
2. Литвинов А.А., Брежнев А.И., Гаврилюк Ю.В. "Особенности построения подсистемы управления потоками работ стационарного лечебного учреждения". //Системные технологии. Региональный межвузовский сборник научных работ.— Выпуск 1(72) — Днепропетровск, 2011.— С.
3. Литвинов О. А. Формалізація клінічного діагнозу і модель електронної класифікації діагнозів для Медичних Інформаційних Систем України. // Системные технологии. Рег. межвуз. сб. научн. работ. — Выпуск 2(49). — Днепропетровск, 2007. - С.46-56.
4. Thomas Erl. SOA Principles of Service Design. Prentice Hall/PearsonPTR. 2007. - 573 p.
5. Andrew Tanenbaum. Distributed systems: principles and paradigms. Prentice Hall. 2 edition. 2006. — 706 p.

Л.Я. Мартинович, О.О. Стеценко

ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ СКЛАДАННЯ РОЗКЛАДУ ЗАНЯТЬ У КОЛЕДЖІ

Анотація. Розроблено автономне програмне забезпечення для складання розкладу занять у коледжі, з урахуванням обмежень на кількість щоденного та тижневого навантаження для студентів, неприпустимості пропусків між заняттями для студентів та обмежень тижневого навантаження для викладачів на основі навчального та робочого планів.

Ключові слова: алгоритм, оптимальне управління ресурсами, тижневе навантаження, навчальний план, робочий план, об'єктно-орієнтований підхід, Borland Delphi, база даних.

Вступ. Якість підготовки студентів у коледжах, а також ефективність використання педагогічного потенціалу залежать певною мірою від рівня організації навчального процесу. Розклад занять є однією з основних складових цього процесу. Оскільки всі фактори, що впливають на розклад, практично неможливо врахувати, а інтереси учасників навчального процесу різноманітні, завдання складання розкладу є багатокритеріальним з нечіткою безліччю факторів. Вирішення таких завдань, як правило, здійснюється в два етапи: отримання оптимального (з точки зору використовуваних критеріїв) варіанта і його подальша доробка людиною з метою максимального врахування неформалізованих факторів. Найбільший внесок у розвиток теорії розкладів[1] внесли: Левин В.І. [5], А. Кофман[6], Танаєв В.С. [7] та ін. В роботі запропоновано вирішення першого етапу проблеми - розробка автоматичного створення розкладу за допомогою ЕОМ. Для розробки алгоритму в першу чергу були виділені вимоги до розкладу занять.

Постановка задачі. Метою даної роботи є розробка програмного продукту для автоматизації процесу складання розкладу занять навчального закладу I-II рівня акредитації на основі навчального та робочого планів з урахуванням обмежень на кількість годин тижневого та щоденного навантаження для студентів, неприпустимості

пропусків між заняттями для студентів та рівномірного тижневого навантаження для викладачів.

Рішення задачі. Залежно від поставленого завдання і послідовності виконуваних кроків розрізняють різні види алгоритмів:

1. Лінійний;
2. Алгоритм з розгалуженням;
3. Циклічний;
4. Комбінований алгоритм - містить декілька з попередніх алгоритмів.

Генетичний алгоритм (англ. genetic algorithm) — це еволюційний алгоритм пошуку, що використовується для вирішення задач оптимізації і моделювання шляхом послідовного підбору, комбінування і варіації шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію. Особливістю генетичного алгоритму є акцент на використання оператора "схрещення", який виконує операцію рекомбінацію рішень-кандидатів, роль якої аналогічна ролі схрещення в живій природі. "Батьком-засновником" генетичних алгоритмів вважається Джон Голланд (англ. John Holland), книга якого "Адаптація в природних і штучних системах" (англ. Adaptation in Natural and Artificial Systems) є фундаментальною в цій сфері досліджень. Генетичні алгоритми є новим напрямком у алгоритмі. Вони здатні не тільки вирішувати і скорочувати перебір у складних завданнях, але й легко адаптуватися до зміни проблеми.

В програмі автоматичного складання розкладу занять був використаний генетичний алгоритм, що включає в себе всі описані вище види алгоритмів.

Схема генетичного алгоритму:

Генерація випадкового початкового стану

2. Перше покоління створюється з довільно вибраних рішень (хромосом), на відміну від стандартних методів, коли початковий стан завжди один і той самий.

3. Обчислення коефіцієнта пристосованості (fitness). Кожному рішенню (хромосомі) зіставляється якесь чисельне значення, яке залежить від його близькості до відповіді.

4. Відтворення. Хромосоми, що мають високий рівень пристосованості (fitness), потрапляють до нащадків (які потім можуть мутувати) з більшою ймовірністю. Нащадок, результат злиття 'батька' і

'матері', є комбінацією їх ген. Цей процес називається 'кроссінг-вер' (crossing over).

5. Наступне покоління. Якщо нове покоління містить рішення, досить близьке до відповіді, то задача вирішена. У протилежному випадку воно проходить через той же процес. Це продовжується до досягнення рішення.

Опис всіх критеріїв складання розкладу, реалізація завдання. Вимоги до розкладу навчальних занять

При складанні розкладу виникає проблема оптимального управління ресурсами: викладацьким складом і аудиторним фондом, а також навантаженням студентів в кожен день занять. В процесі рішення задачі необхідно враховувати обов'язкові обмеження:

– розклад має бути складено так щоб дисципліни, які читаються більше 2 годин на тиждень - не проводились в один день. Тому, якщо дисципліна читається 4 години на тиждень - між заняттями повинна бути перерва мінімум в один день, якщо 6 – дві пари можуть знаходитися поряд, але третя повинна бути як мінімум через день.

– тижневе навантаження практично у всіх груп становить 32 години, з яких 2 або 4 години - це дисципліна «Фізична культура». У студентів повинен бути розклад з рівномірним навантаженням, кожен день може бути по 3 пари і тільки в один день - 4. Чотири пари у студентів може бути тільки в той день, коли у них є дисципліна «Фізична культура», а також в день у студентів важкі пари повинні перемежатися з гуманітарними дисциплінами.

– перша пара в понеділок і остання пара в п'ятницю не повинні бути складними дисциплінами, а в інші дні повинно бути відносно середнє навантаження (тобто складні дисципліни «розбавляються» більш відносно простими).

– так само є обмеження і для викладачів, у яких розклад занять має бути рівномірним. Це означає, що у викладача тижневе навантаження дорівнює або менше 30 годин на тиждень, в день у нього повинно бути не більше трьох пар.

– в заняттях студентів не повинно з'являтися вікон, в той же час наявність вікна в розкладі викладача допустима.

У підсумку виконання програми повинен бути отриманий розклад занять, в який після автоматичного створення можна буде внес-

ти необхідні зміни людиною. Розклад буде представлено в трьох формах взаємопов'язаних між собою (при зміні даних в одній з форм - інші 2 змінюються автоматично) для подальшої роботи, це:

- Студентський розклад занять, який погоджують з заступником директора з НР, з НМР та затверджує директор коледжу.
- Розклад занять по викладачам, у якому можна побачити розклад викладачів, відсортованих по предметно-цикловим комісіям.
- Картку розподілу аудиторій по днях тижня, для того щоб можна було переглянути зайнятість аудиторій для подальшої роботи з розкладом занять.

Таблиці, вихідні коди, ієрархія таблиць

Вхідні таблиці, які необхідні для складання розкладу:

- список викладачів;
- список навчальних груп;
- навчальний план;
- робочий план, який включає в себе прізвища викладачів;

Вихідні таблиці:

- студентський розклад занять:

STUDENT.xlsx - Microsoft Excel

Главная Вставка Разметка страницы Формулы Данные Рецензирование Вид Разработчик Нагрузочный тест

A1 fx Дніпропетровський коледж ракетно-космічного машинобудування Дніпропетровського національного університету імені Олеся Гончара

Дніпропетровський коледж ракетно-космічного машинобудування
Дніпропетровського національного університету імені Олеся Гончара

УЗГОДЖЕНО

Заст.директора з НМР

В.Г. Мирошник

ЗАТВЕРДЖУЮ

Директор коледжу

О.М. Романовський

" " 2012 р.

РОЗКЛАД ЗАНЯТЬ

пн	час заняття	БО-11-1	КА	КС-11-1	КА	КС-11-2	КА	ПЗ-11-1	КА	ПЗ-11-2	КА	ПУ-11-1	КА	ТО-11-1	КА
понеділок	1. 8.00-9.20	Укр.літ. Андрушак Ю.О.	12	Математика Малик О.М.	39	Укр.літ. Андрушко О.М.	24	Ін.мова Шершук О.В.	40 9	Ін.мова Антонюк О.В.	30	Математика Віталь Т.П.	36	Біологія Гуцанко Л.М.	32
	2. 9.30-10.50	Математика Чопенко О.В.	36	Укр.літ. Андрушак Ю.О.	12	Фізика Мирошник О.В.	47	Осн.прогр.та АМ Вавченко І.В.	30 3	Укр.мова Андрушко О.М.	30 3	Укр.літ. Мороз Ю.О.	36	Іноземна мова Клименко І.А.	48
	3. 11.00-12.20	Біологія Гуцанко Л.М.	32	Фізика Мирошник О.В.	47	Математика Чопенко О.В.	36	Укр.мова Андрушак Ю.О.	12	Осн.прогр.та АМ Вавченко І.В.	30 3	Ін.мова Антонюк О.В.	30	Укр.літ. Мороз Ю.О.	32
	4. 12.50-14.10	Техн.галузі Мирошник В.Г.	51				Укр.літ. Мороз Ю.О.	12	Іст. України Григор'єв О.В.	20	Укр.мов. Андрушко О.М.	20	Біологія Гуцанко Л.М.	32	
	5. 8.00-9.20	Укр.літ. Андрушак Ю.О.	12	Математика Малик О.М.	39	Укр.літ. Андрушко О.М.	24	Ін.мова Шершук О.В.	40	Ін.мова Антонюк О.В.	30	Математика Віталь Т.П.	36	Біологія Гуцанко Л.М.	32

Рисунок 1 – Приклад розробленого розкладу занять в вигляді студентського розкладу в програмі для перегляду і подальшого друку Microsoft Office Excel 2007. Розклад занять по викладачам

The screenshot shows an Excel spreadsheet titled 'PREPOD.xlsx - Microsoft Excel'. The worksheet is named 'X14' and contains a lesson schedule for 'TO-11-1'. The columns are labeled A through T, and the rows are numbered 1 through 31. The schedule is organized by day of the week, with days labeled in the first column: 'понеділок' (Monday), 'вівторок' (Tuesday), 'середа' (Wednesday), 'четвер' (Thursday), and 'п'ятниця' (Friday). Each day has a list of lessons with their numbers and the names of the teachers. For example, on Monday, lesson 1 is taught by Акімов Ю.О., lesson 2 by Андрющак Ю.О., and lesson 3 by Клименко І.А. The schedule continues through Friday, with lesson 4 being the last entry.

Рисунок 2 – приклад розробленого розкладу занять в вигляді розкладу по викладачам в програмі для перегляду і подальшого друку Microsoft Office Excel 2007. Загрузка аудиторій

The screenshot shows a more detailed Excel spreadsheet for lesson scheduling. It includes columns for the day of the week (Понеділок, Вівторок, Середа) and the lesson number (1, 2, 3, 4, 5). The rows are numbered 1 through 65. The spreadsheet lists the names of the teachers and the rooms assigned to each lesson. For example, lesson 1 on Monday is taught by Акімов Ю.О. in room 11. The schedule continues through Friday, with lesson 65 being the last entry. The spreadsheet is divided into three sections, labeled 'Страница 1', 'Страница 2', and 'Страница 3'.

Рисунок 3 – приклад розробленого розкладу занять в формі завантаження аудиторій по дням тижня в програмі для перегляду і подальшого друку Microsoft Office Excel 2007

Всі вихідні таблиці взаємодіють між собою, таким чином - якщо треба внести зміни у розклад, то можна змінити лише в студентському розкладі занять, а в двох інших вихідних таблицях розклад зміниться автоматично.

Алгоритм програми

Алгоритм рішення задачі представлений у вигляді піктограми сценарію рішення задачі, яка наведена на рисунку:

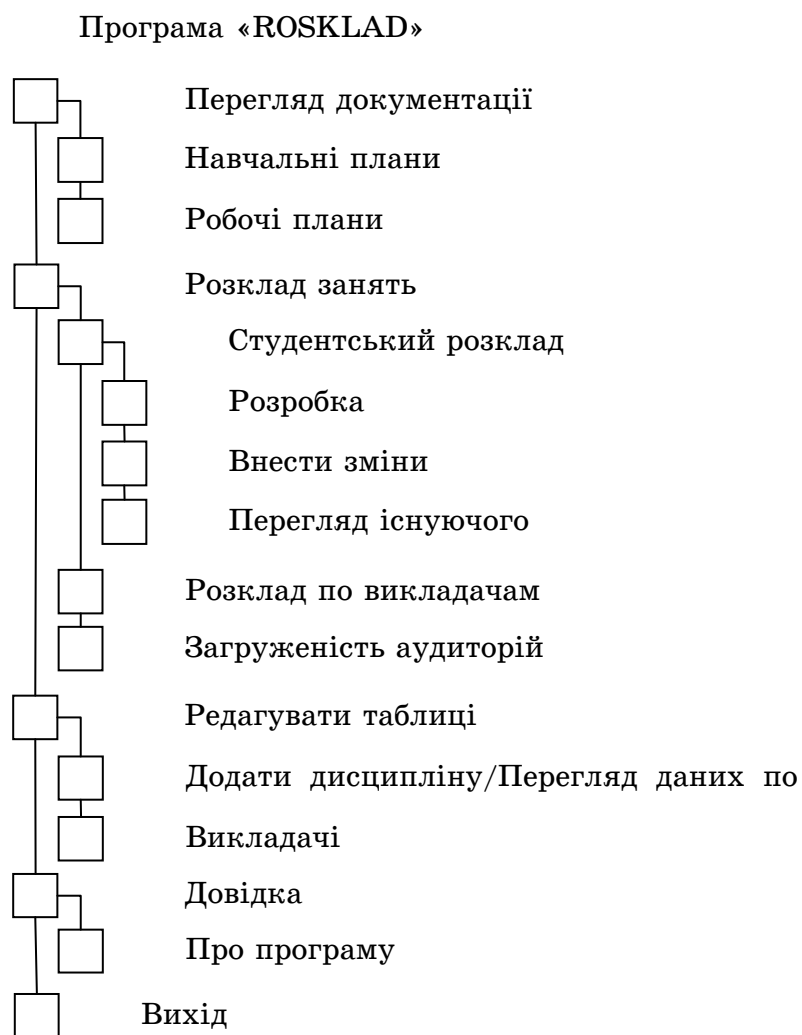


Рисунок 4 - Логічна структура задачі, яка проектується за методом "зверху-вниз"

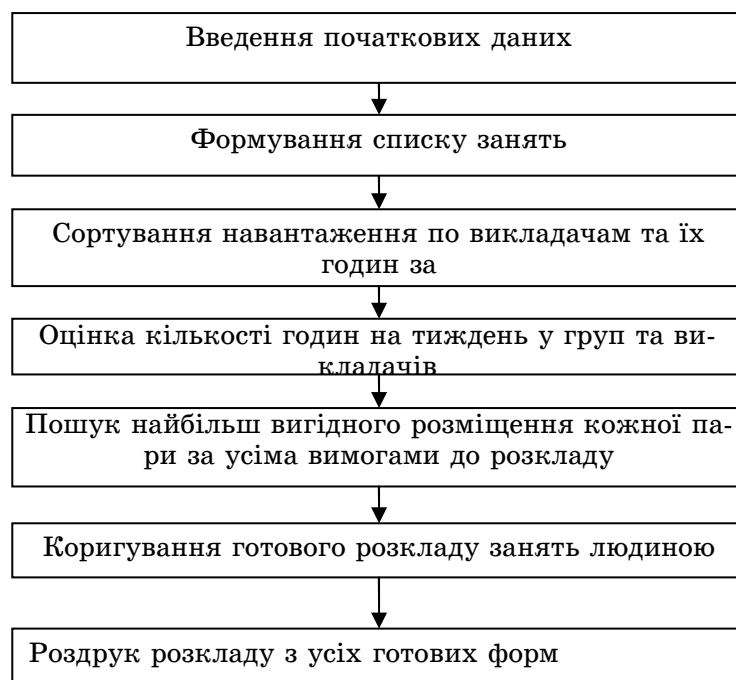


Рисунок 5 - Структурна схема алгоритму створення розкладу занять

В майбутньому є можливість розвинути дану програму, додавши в неї нові можливості, такі як доповнення у вигляді програми замін, побажання або вимоги до викладацького розкладом, складання екзаменаційної сесії та графіка відвідувань ДПА та іспитів адміністрацією, і т.п.

Система розроблена на платформі Win32 в середовищі Borland Delphi. У якості бази даних використовується формат Microsoft SQL Server. Система працює в операційних системах Windows XP і т.д. Рекомендована конфігурація: Pentium III і вище, 128-256 Mb, 600 Mb на жорсткому диску без розміру баз даних користувача. З метою економії часу для роботи з великими масивами та статистичного аналізу даних рекомендується Pentium IV (Celeron) і вище. Система працює в багатокористувальницькому режимі (клієнт-сервер). Вихідні документи формуються у форматі Microsoft Office Excel 2007.

Висновки. В результаті проектування автоматизованої системи була розроблена та впроваджена у навчальний відділ денного відділення Дніпропетровського коледжу ракетно-космічного машинобудування програма, яка містить в собі наступні можливості для роботи з розкладом занять:

- автоматизація складання розкладу занять;
- внесення змін у вже складений розклад людиною;

- перегляд існуючого студентського розкладу, розкладу по викладачам та перегляд зайнятості аудиторій у MS Excel;
- внесення нових, перегляд та внесення коректив у БД з викладачами;
- перегляд та внесення змін у робочі плани.

Програма була написана з використанням середовища розробки програмних продуктів Borland Delphi 7. Завдяки об'єктно-орієнтованому підходу була забезпечена цілісність даних, також програма може бути реалізована без ускладнень та в майбутньому модернізована. Всі дані, які використовуються в програмі, зберігаються віддалено у базі даних. Створення бази даних та робота з нею здійснюється за допомогою реляційної СУБД Microsoft SQL Server 2008, що має великі можливості для роботи з базами даних. Взаємодія програми з базою даних виконується за допомогою технології ADO. Обробка даних проводиться за допомогою мови запитів SQL. Програма може бути використана в інших навчальних закладах I - II рівня акредитації.

ЛІТЕРАТУРА

1. Конвей Р. В., Максвелл В. Л., Миллер Л. В. Теория расписаний. Москва: Главная редакция физико-математической литературы изд-ва "Наука", 1975.
2. Кормен Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест // МЦНМО. – Москва, 2000. – 960 с.
3. Колмогоров А. Н. Теория информации и теория алгоритмов. — М.: Наука, 1987. — 304 с.
4. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. — Горячая Линия – Телеком, 2007. — 452 с.
5. Левин В.И. Структурно-логические методы в теории расписаний. Пенза: Изд-во Пенз. гос. технол. акад., 2006.

М.А. Алексеев, К.А. Ахметшин

СИНТЕЗ ОБОБЩЕННЫХ ФАЗОВЫХ ПОРТРЕТОВ МНОГОКАНАЛЬНЫХ НЕЛИНЕЙНЫХ ДИНАМИЧЕСКИХ СИСТЕМ В СИНГУЛЯРНОМ БАЗИСЕ ВЛОЖЕННЫХ ВЕКТОРОВ

Аннотация. Исследован новый метод отображения динамики многоканальных нелинейных нестационарных систем. суть метода основана на двукратном использовании алгоритма сингулярного разложения: на первом этапе происходит сжатие многоканальных данных на основе использования сингулярного разложения первого порядка матрицы сигнальных векторов, а на втором – сингулярное разложение матрицы вложенных векторов от первого сингулярного вектора сингулярного разложения первого порядка. показано преимущество отображения фазовых портретов в новом информационном базисе на примере анализа многоканальной электрокардиограммы.

Ключевые слова: многоканальная нелинейная система, сингулярное разложение, адаптивная фильтрация, обобщенный фазовый портрет, электрокардиограмма.

Постановка проблемы. Распространенной методом анализа особенностей одноканальных (однопараметровых) нелинейных динамических систем (в частности, электрокардиограмм (ЭКГ)) является построение их фазовых портретов в системе координат $z(t), dz(t)/dt$ [1, 2]. Поскольку ЭКГ часто зашумлены как измерительными, так и физиологическими шумами, то стандартный способ синтеза фазовых портретов диктует необходимость их предварительной фильтрации [3] в условиях, когда отсутствует априорная информация о спектральных и статистических характеристиках шумов. Более того, учитывая современную тенденцию использования многоканальных ЭКГ, необходимо синтезировать и анализировать большое количество фазовых портретов, что делает использование стандартных методов трудоемким и вычислительно неэффективным.

Данное обстоятельство диктует необходимость разработки новых, помехоустойчивых методов синтеза обобщенных фазовых портретов многоканальных ЭКГ.

Целью работы является демонстрация информационных возможностей нового устойчивого метода синтеза обобщенных фазовых портретов многоканальных нелинейных динамических систем на примерах использования многоканальных ЭКГ.

Основная часть. На рис. 1 представлены фрагменты 15-канальной ЭКГ (диагноз – инфаркт) полученный из открытой базы данных “Physionet” (интервал между отсчетами $T=0.004$ сек.).

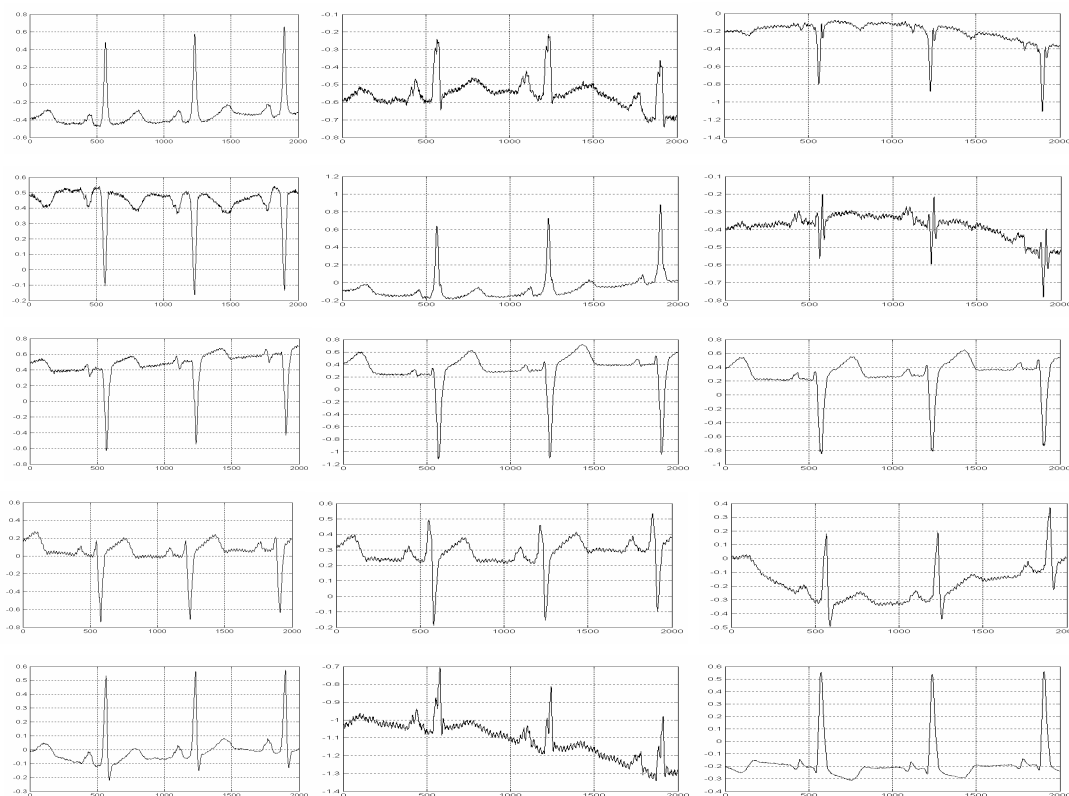


Рисунок 1 – Фрагменты 15-ти канальной ЭКГ
(сигнальная матрица Z)

Визуальный анализ рис. 1 показывает, что измерительные каналы зашумлены и являются сильно коррелированными. Например, если синтезировать фазовый портрет ЭКГ соответствующий второму измерительному каналу, то, как показывает анализ рис. 2, такой прямолинейный подход не дает удовлетворительного результата из-за сильного влияния шумов на результат вычисления первой производной сигнального вектора.

Поэтому, первой задачей построения обобщенного фазового портрета многоканальной ЭКГ является сжатие (устранение избыточности) исходных данных. Возможны различные варианты решения данной задачи, но по нашему мнению наиболее целесообразно здесь использование алгебраического метода сингулярного разложения исходной сигнальной матрицы данных $Z = [z_1, \dots, z_M]$, размерностью $(N * M)$, где N – число отсчетов в каждом сигнальном канале, а M – общее число измерительных каналов.

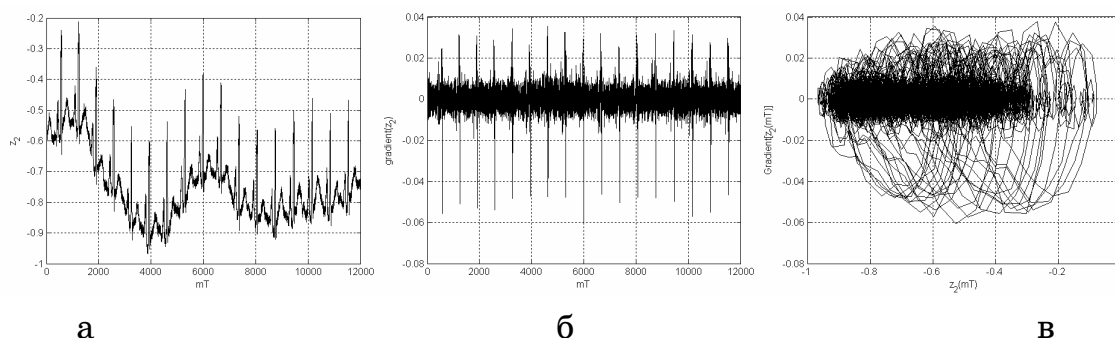


Рисунок 2 – Влияние шума на синтез фазового портрета: а – фрагмент второго измерительного канала $z_2(t)$; б – первая производная $dz_2(t)/dt$; в – фазовый портрет в системе координат $z_2(t) - dz_2(t)/dt$

Сингулярное разложение матрицы Z имеет вид [4]

$$Z = USV^T, \quad (1)$$

где U – ортонормированная матрица сингулярных векторов размерностью $(N * M)$; S – диагональная матрица сингулярных чисел размерностью $(M * M)$, причем $s_1 > s_2 > \dots > s_M$; V – ортонормированная матрица размерностью $(M * M)$. Особое значение имеет анализ матрицы сингулярных чисел S , компоненты которой можно трактовать как относительные величины проекций сигнальной матрицы Z на ортонормированный базис образованный столбцами (сингулярными векторами) матрицы U .

На рис. 3а представлен сингулярный спектр сигнальной матрицы Z .

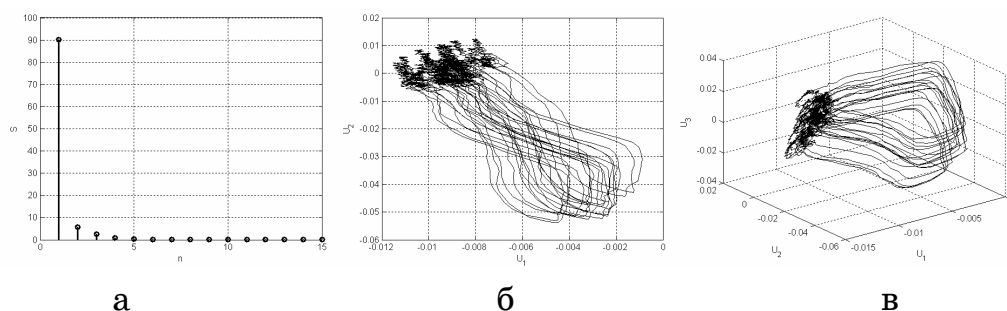


Рисунок 3 – Сингулярное разложение сигнальной матрицы Z : а – нормированный спектр сингулярных чисел; б – фазовый портрет в системе координат $U_1 - U_2$; в – фазовый портрет в системе координат $U_1 - U_2 - U_3$

Из его рассмотрения следует, что первые два сингулярных вектора отображают 96% изменчивости динамики всех 15-ти каналов ЭКГ (рис. 3 б), а три сингулярных вектора – 98,7% всей информации, т.е. синтез фазового портрета в системе координат $U_1 - U_2 - U_3$ можно рассматривать в виде аналога обобщенного фазового портрета анализируемой 15-ти канальной системы (рис. 3 в), что позволяет облегчить анализ и интерпретацию данных.

Каждый сингулярный вектор, представляет собой взвешенную сумму всех исходных векторов, с весами равными порядковому номеру столбцов матрицы V в выражении (1).

На рис. 4 представлены графики первых трех сингулярных векторов, которые и можно рассматривать в качестве нового информационного базиса.

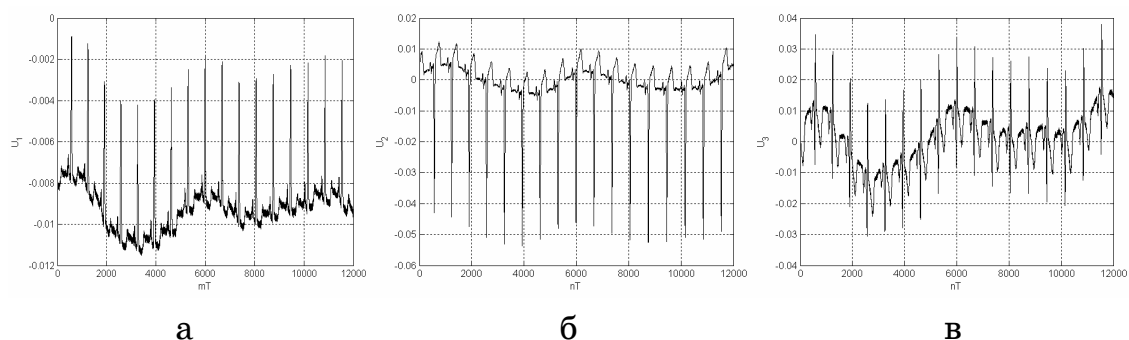


Рисунок 4 – Первые три сингулярных вектора сигнальной матрицы Z : а – U_1 ; б – U_2 ; в – U_3

Анализ рис. 3 показывает, что в обобщенных фазовых портретах все же присутствует шумовой фактор. В [5] были рассмотрены

информационные возможности метода адаптивной фильтрации одно-канальных ЭКГ, не требующего априорной информации о характеристиках шумов. С учетом того, что первый сингулярный вектор (рис.3а) содержит 92% всей информации, то в целях его фильтрации и синтеза более качественных (отфильтрованных) фазовых портретов, возможно обобщение метода [5] на область многоканальных динамических систем при условии, что метод сингулярного разложения вложенных векторов применяется к первому сингулярному вектору (рис.4а) разложения (1), т.е. синтезируется новая матрица вида

$$U_1 \Rightarrow Q = \begin{bmatrix} u_1(t_1) & u_2(t_2) & \dots & u_1(t_L) \\ u_1(t_2) & u_1(t_3) & \dots & u_1(t_{L+1}) \\ \cdot & \cdot & \dots & \cdot \\ u_1(t_{N-L+1}) & u_1(t_{N-L+2}) & \dots & u_1(t_N) \end{bmatrix}, \quad (2)$$

где $N \gg L$, а L характеризует глубину вложения.

Затем производится сингулярное разложение матрицы Q (т.е., фактически, сингулярное разложение второго порядка) как

$$Q = PRD^T, \quad (3)$$

с последующим синтезом обобщенного фазового портрета уже в системе координат $P_1 - P_2$ или $P_1 - P_2 - P_3$ (рис. 5).

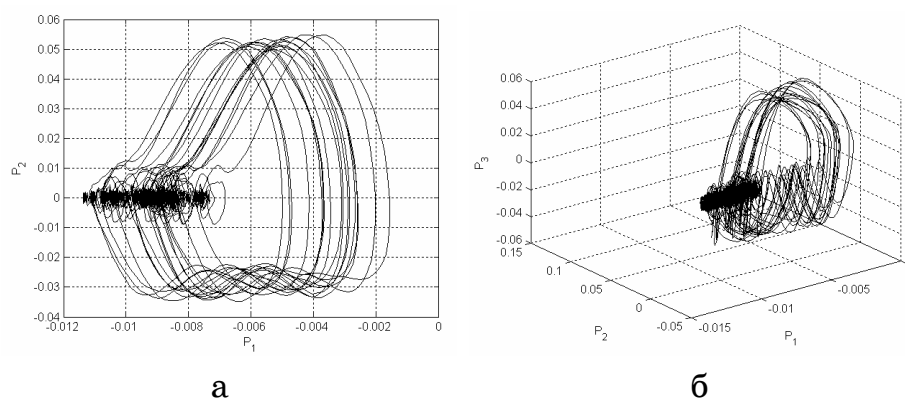


Рисунок 5 – Обобщенный фазовый портрет второго порядка сигнальной матрицы Z в системах координат:

а – $[P_1 - P_2]$; б – $[P_1 - P_2 - P_3]$ (глубина вложения $L = 20$ отсчетов)

Сопоставление рис. 3 б, в и рис. 5 свидетельствует о том, что синтез обобщенных фазовых портретов в сингулярном базисе вложенных векторов от первого сингулярного вектора сингулярного разложения сигнальной матрицы Z на основе предварительного использования выражения (1), позволяет нейтрализовать влияние шумов (P_3

фактически соответствует шумовой компоненте (рис. 6 б), поэтому синтез трехмерного фазового портрета (рис.3в), с практической точки зрения, в данном случае следует признать нецелесообразным) и повысить «качество» визуального анализа топологических особенностей фазовых портретов многоканальных нелинейных динамических систем.

Поскольку в векторе P_2 (рис. 6 б) присутствует шумовая компонента, то для дополнительной нейтрализации влияния шумовых факторов возможен и вариант построения обобщенного фазового портрета многоканальной системы на основе использования лишь первого сингулярного вектора $P_1(t)$ матрицы вложенных векторов (рис. 6 а).

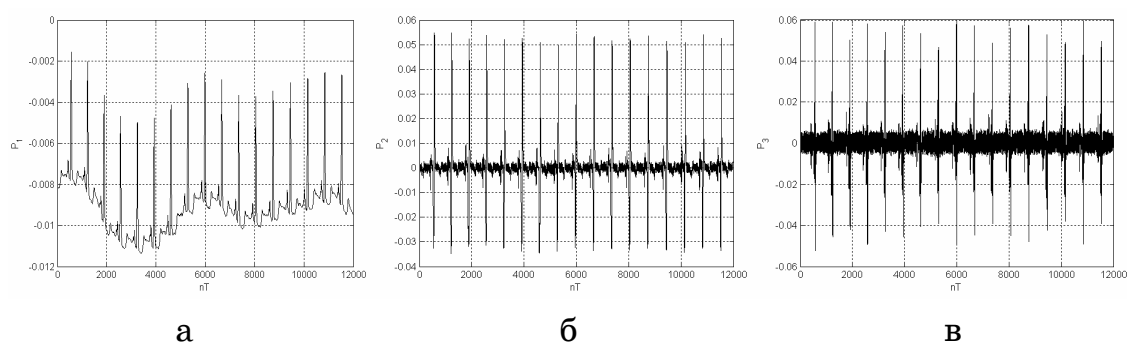


Рисунок 6 – Первые три сингулярных вектора матрицы вложенных векторов Q от первого сингулярного вектора U_1 на рис.4а:

а – P_1 ; б – P_2 ; в – P_3

Здесь возможно использование двух вариантов. Первый – синтез фазового портрета в системе координат $P_1(t) - dP_1(t)/dt$ (рис.7а).

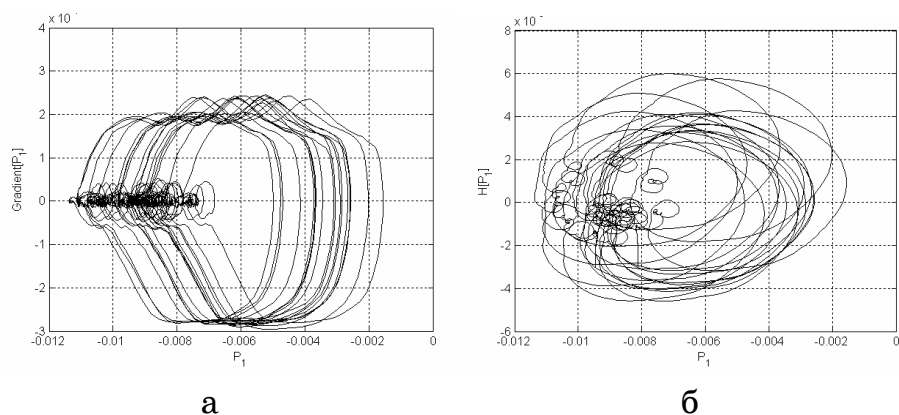


Рисунок 7 – Фазовые портреты на основе использования только первого вектора сингулярного разложения матрицы вложенных векторов в системах координат: а – $P_1(t) - dP_1(t)/dt$; б – преобразование Гильберта от $P_1(t)$

Второй вариант базируется на использовании преобразования Гильберта от $P_1(t)$ на основе выражения [6]

$$H[P_1(t)] = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{P_1(t-\tau)}{\tau} d\tau, \quad (4)$$

с последующим формированием комплексного аналитического сигнала вида

$$G(t) = P_1(t) + jH[P_1(t)] = A(t)\exp[jF(t)], \quad (5)$$

$$\text{где } A(t) = \sqrt{P_1^2(t) + \{H[P_1(t)]\}^2}; F(t) = \arctg[H[P_1(t)]]/P_1(t). \quad (6)$$

На рис. 8 отображены все три характеристики: $H[P_1(t)]$, $A(t)$ и развернутая

фазовая характеристика $F(t)$. Поскольку $P_1(t)$ и $H[P_1(t)]$ ортогональны, то это и позволяет синтезировать фазовый портрет системы (рис. 7 б).

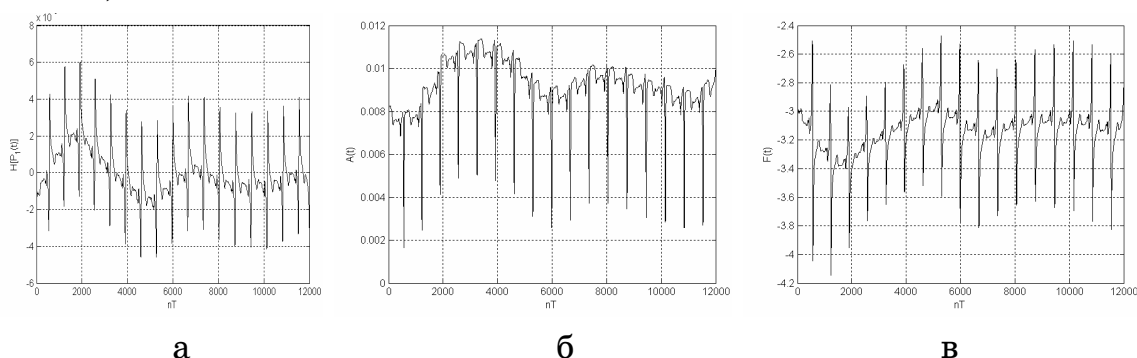


Рисунок 8 – Преобразования Гильберта от $P_1(t)$:

а – $H[P_1(t)]$; б – $A(t)$; в – $F(t)$

Сопоставляя рис. 5 а и рис. 7 следует отметить, что отображение динамики системы в координатах $P_1(t) - P_2(t)$ и $P_1(t) - dP_1(t)/dt$ качественно похожи друг на друга, в то время как синтез фазового портрета в системе координат $P_1(t) - H[P_1(t)]$ дает иной результат (рис. 7 б). С теоретической точки зрения, данный факт можно объяснить тем обстоятельством, что использование преобразования Гильберта не приводит к увеличению влияния шума, тогда как операция вычисления производной относится к классу некорректных задач [7].

Для демонстрации эффективности предлагаемого подхода, на рис. 9 представлены обобщенные фазовые портреты 15-ти канальной ЭКГ с диагнозом «аритмия».

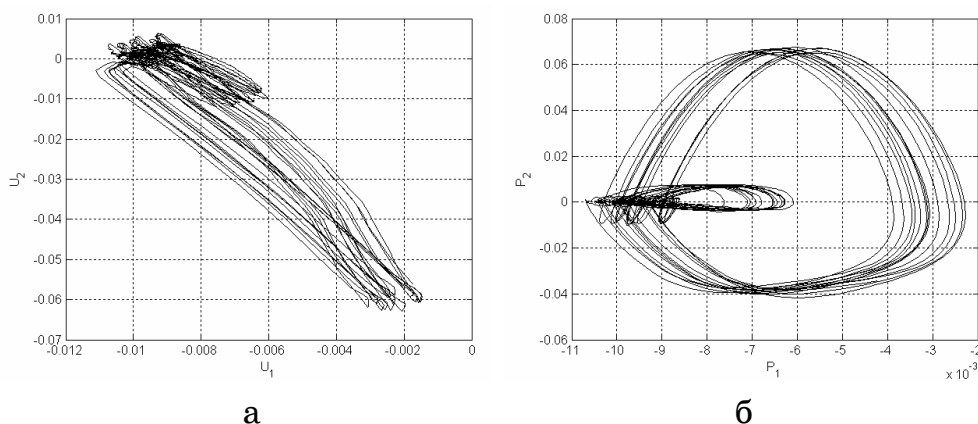


Рисунок 9 – Обобщенные фазовые портреты 15-ти канальной ЭКГ с диагнозом «аритмия» в системах координат:

а – $[U_1(t) - U_2(t)]$; б – $[P_1(t) - P_2(t)]$

Данный пример представляется особенно показательным, поскольку отображение фазового портрета в системе координат $[U_1(t) - U_2(t)]$ (рис. 9 а) не позволяет однозначно выявить характерные особенности данного типа заболевания, а отображение фазового портрета в системе координат $[P_1(t) - P_2(t)]$ (рис. 9 б) эти особенности, связанные с характерным расслоением траекторий, четко выявляет.

Выводы.

1. Синтез обобщенного фазового портрета многоканальной нелинейной динамической системы базируется на идее двукратного использования алгоритма сингулярного разложения.

2. Первый этап – сингулярное разложение сигнальной матрицы в целях сокращения избыточности данных.

3. Второй этап – сингулярное разложение матрицы вложенных векторов, полученной на основе первого сингулярного вектора сингулярного разложения сигнальной матрицы, в целях нейтрализации влияния измерительного шума на характерные особенности синтезируемых обобщенных фазовых портретов.

4. В качестве основы построения фазового портрета используется первый сингулярный вектор сингулярное разложение матрицы вложенных векторов, причем здесь возможно использование двух различных методов основанных на использовании преобразования Гильберта или взятия производной от первого сингулярного вектора.

ЛИТЕРАТУРА

1. Файнзильберг Л.С. Информационные технологии обработки сигналов сложной формы: теория и практика / Файнзильберг Л.С.; – Киев, Наукова думка, 2008. – 333 с.
2. Perc M. Nonlinear time series analysis of the human electrocardiogram // European Journal of Physics. – 2005. – Vol. 26. – P. 757–768.
3. Рангайян Р.М. Анализ биомедицинских сигналов: практический подход / Рангайян Р.М.; [пер. с англ. А.П. Немирко].– М.:, Физматлит, 2007.– 440 с.
4. Беклемишев Д.В. Дополнительные главы линейной алгебры /– М.: Наука, 1983. – 455 с.
5. Алексеев М.А. Адаптивная фильтрация и синтез фазовых портретов нелинейных динамических систем в сингулярном базисе вложенных векторов /Алексеев М.А., Ахметшин К.А.// Системные технологии, 2012, Днепропетровск, Вип.1(78), Т.2, С.127–132.
6. Баскаков С.И. Радиотехнические цепи и сигналы / Баскаков С.И. – М.: Высшая школа, 1987. – 207 с.
7. Тихонов А.Н. Методы решения некорректных задач / Тихонов А.Н., Арсенин В.Я. – М.: Наука, 1986. – 224 с.

Л.Г. Ахметшина, А.М. Ахметшин, И.М. Удовик
**МЕТОД ИНТЕРПОЛЯЦИИ НЕРАВНОМЕРНЫХ
ПРОСТРАНСТВЕННЫХ ДАННЫХ
НА ОСНОВЕ СЕТИ КОХОНЕНА**

Аннотация. Рассмотрены информативные возможности метода интерполяции пространственных данных заданных на неравномерной сетке, основанной на результатах кластеризации с использованием самоорганизующейся сети Кохонена. Представлены результаты проверки метода на модельных и реальных данных.

Ключевые слова: самоорганизующаяся карта Кохонена, параметры обучения, пространственные данные, интерполяция, неравномерная сетка.

Постановка проблемы. Построение моделей пространственного распределения различных свойств природных явлений требуется при решении прикладных задач, например, в геосистемах при оценке границ возможных зон подтопления и загрязнения, определения объемов рудного тела, прогнозном картировании.

Наиболее сложной является задача построения поверхности по малому конечному набору исходных значений с неравномерной плотностью сетки мониторинга, которые обычно и являются результатом реальных измерений: (результаты бурения, измерение уровня вод в контрольных скважинах и др). Использование традиционных математических подходов проблематично, а единственным способом оценки достоверности прогнозирования является эмпирическое тестирование на модельных данных.

Современный подход решения таких задач возможен на базе нечеткого и нейросетевого подходов [1, 2]. Преимущество нечетких моделей связано с возможностью использования малых объемов исходной информации, причем она может носить приближенный характер. В работе [3] демонстрируется возможность интерполяции экспериментальных пространственных данных заданных на неравномерной сетке методом двумерного проецирования нечетких кластеров.

Недостатком метода является значительная ошибка на границе области достоверности и сложность настройки вида функций принадлежности

Использование нейронных сетей обычно требует значительного объема исходных данных, которые необходимы для их обучения и тестирования. Однако, самоорганизующаяся нейронная сеть Кохонена (СКК) позволяет работать с небольшими объемами данных исходных данных и имеет ряд преимуществ перед другими архитектурами – отражает их топологические особенности, что особенно важно при обработке пространственных данных, и обладает прогнозирующими свойствами.

Целью данной работы является описание информативных возможностей метода интерполяции пространственных данных с использованием самоорганизующейся сети Кохонена, который позволяет в условиях малого объема нерегулярно расположенных исходных данных интерполировать поверхность на регулярную сетку произвольного масштаба.

Основная часть. На рис. 1 показана архитектура нейронной сети СКК – самоорганизующейся карты, которая является относительно новым методом вычислительного интеллекта [4].

$(x_1, x_2, \dots, x_n)$ – входной вектор

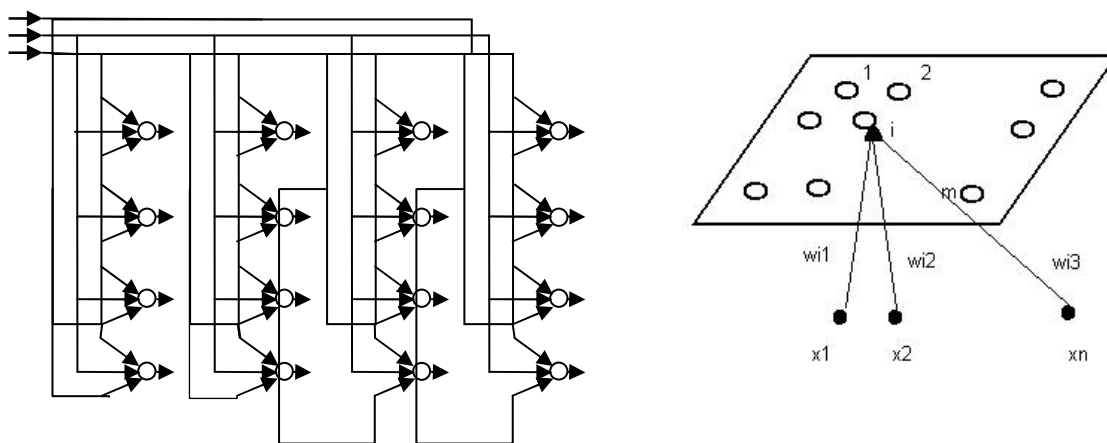


Рисунок 1 - Структурная схема карты Кохонена: $(n_x \times n_y)$ – общее число узлов в карте; $(x_1, x_2, \dots, x_n)$ – входы;

$W_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$ – вектор весов для узла i

Нейроны сети образуют двумерную решетку и связаны со всеми узлами входного слоя: n_x, n_y – число узлов в карте по соответствующим координатным осям; $(x_1, x_2, \dots, x_n)$ – исходные данные размерности n ; $W_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$ – весовые коэффициенты i -го узла. При использовании метода «интеллектуальной» аппроксимации на основе данной архитектуры вид функциональной зависимости автоматически определяется данными и некоторыми заранее заданными ограничениями.

Обеспечение процедуры самоорганизации происходит за счет использования трех процессов:

конкуренция – рассчитывается значение дискриминантной функции, на основе которой определяется нейрон-победитель – имеющий наименьшее различие с рассматриваемым образцом;

кооперация – победивший нейрон определяет пространственное положение топологической окрестности нейронов-соседей, параметры которых подлежат корректировке;

синаптическая адаптация – усиление значений нейронов-соседей посредством корректировки их синаптических весов (самоусиление) (рис. 2).

Кратко структуру алгоритма можно представить в следующем виде [1] .

1. Задание значений n_x, n_y и инициализация, например, случайным образом, начальных значений исходных векторов синаптических весов w_j для каждого нейрона m_j .

2. Выбор вектора X из входного пространства.

3. Поиск нейрона-победителя (ближайшего узла) m_c для вектора X с использованием критерия подобия. Для евклидова расстояния, он определяется по формуле $\arg(\min \|x - w_j\|), j = 1, \dots, k$, где $k = n_x \times n_y$ – количество нейронов сети.

4. Коррекция синаптических весов всех нейронов-соседей (рис. 2): $w_j(t+1) = w_j(t) + \eta(t)h_{j,i(x)}(t)(x - w_j(t))$, а для остальных узлов $w_j(t+1) = w_j(t)$, где t – шаг итерации; $0 < \eta(t) < 1$ – параметр скорости обучения, убывающий с каждым шагом; $h_{i,j(x)}$ – функция окрестности с центром победившего нейрона, определяющая силу взаимного влияния любой пары узлов сети, т.е. расстояние, на которое переда-

ется коррекция. Эти параметры динамически изменяются во время обучения с целью получения лучшего результата.

5. Если разность целевых функций текущего и предыдущего меньше ε , где ε – пороговое значение, то переход к пункту 2.

После построения СКК на нее отображаются входные данные. Результат – исходные многомерные данные аппроксимируются к $n_x \times n_y$ узлам сети, что позволяет получить проекцию всего пространства наблюдений на двумерную плоскость и отобразить важные статистические характеристики входного пространства, которые можно использовать при решении задачи прогноза неопределенных значений.

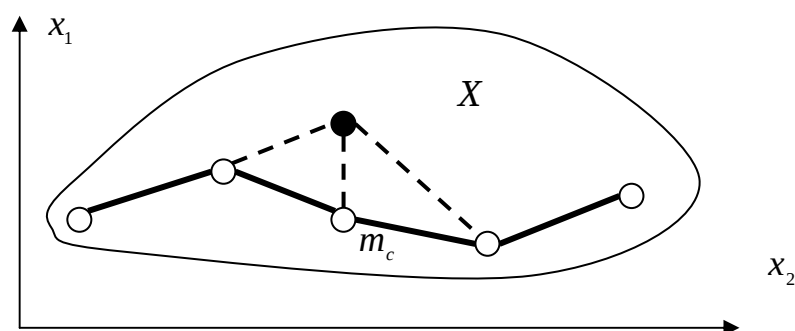


Рисунок 2 – Шаг коррекции синаптических весов
для двумерного входного вектора $X = (x_1, x_2)$

После построения СКК на нее отображаются входные данные. Результат – исходные многомерные данные аппроксимируются к $n_x \times n_y$ узлам сети, что дискриминантных функций позволяет получить проекцию всего пространства наблюдений на двумерную плоскость и отобразить важные статистические характеристики входного пространства, которые можно использовать при решении задачи прогноза неопределенных значений.

При постановке задачи интерполяции пространственных данных исходная информация рассматривается в виде равнозначных результатов измерений, в которых выделяются две части X , Y – пространственные координаты и $Z_i, i=1, \dots, n$ – любая другая характеристика, восстановление значения которой необходимо произвести в точках равномерной координатной сетки. Обработка таких данных предполагает необходимость сохранения отношения соседства, обеспечения однозначности и непрерывности при преобразовании на рав-

номерную сетку системы координат и возможности работы с различными масштабами (в зависимости от задачи – оценочная, уточняющая).

Применение для решения такой задачи СКК показало следующее:

- карта не покрывает всей области значений измерений, что приводит к существенным ошибкам для координат, находящихся за ее пределами;

- при существенном различии диапазона изменения компонент исходных векторов карта имеет существенно скошенный вид, усложняющий формирование на выходе равномерной сетки (рис. 1 а);

- за счет отсутствия всех соседей у граничных нейронов за ее пределами наблюдается значительное увеличение ошибки интерполяции (рис. 1 б).

Для устранения этих недостатков предлагается выполнение дополнительных операций.

- расширение области определения модели за границы области определения системы;
- инициализация начальных значений нейронов для компонент X , Y на равномерной координатной сетке;
- инициализация начальных значений исходных векторов синаптических весов w_j для каждого нейрона на основе анализа значений зависимой переменной;
- определение лучшего аппроксимирующего вектора только по значениям координат (пространственная близость).

Экспериментальные результаты были получены с использованием СКК, архитектура которой выбиралась с учетом того, что на результаты кластеризации существенное влияние оказывает как общее число нейронов в сети, так и соотношение количества нейронов n_x , n_y по осям X , Y .

В качестве исходных для интерполяции модельных данных выбирались значения известного гравитационного поля участка Земной поверхности $Z = f(x, y)$ (рис. 2 а) в координатах (x_i, y_i) $i = 1, \dots, 57$, соответствующих координатам реальных разведочных скважин, предназначенных для оценки залегания угольных пластов (рис. 2 б).

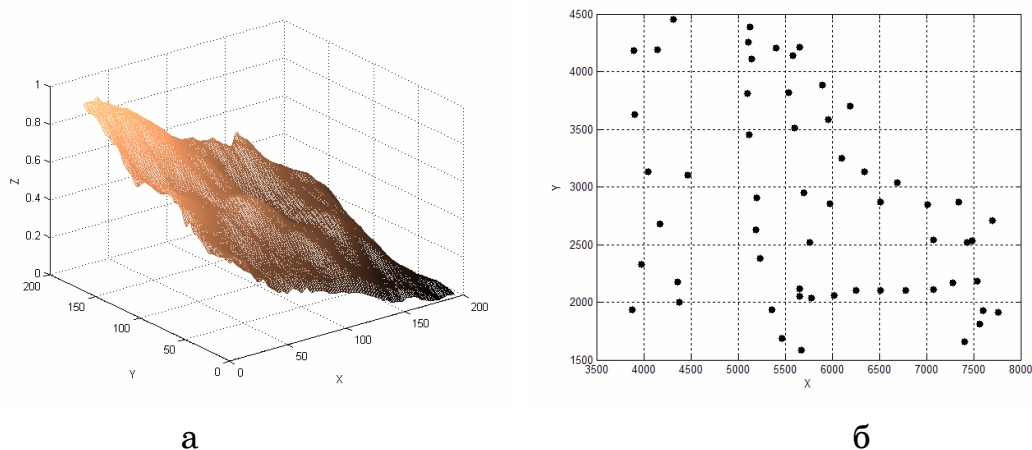


Рисунок 3 - а – геофизическое поле, используемое для формирования модельных данных ;б – неравномерно расположенные экспериментальные данные, используемые при интерполяции

После обучения сети, целевое значение, которое подлежит восстановлению, ассоциируется с ближайшим узлом сетки, что означает замену точки на центр локального сгущения точек данных в пространстве, которое хранится в кодовом блокноте (codebook). Для оценочного моделирования вполне достаточно непосредственная визуализация последнего.

Существенное значение на точность и гладкость результирующей модельной поверхности оказывает величина используемого радиуса соседства. При увеличении его значения повышается гладкость поверхности решения, при этом сужается область интерполяции, увеличивается ошибка в граничной зоне. Уменьшение радиуса соседства с одной стороны, повышает точность аппроксимации каждого экспериментального значения (в пределе при значении равном 1, ошибка стремится к нулю), однако поверхность приобретает ступенчатый вид.

Точность восстановления пропущенных данных можно также повысить, за счет увеличения размерности карты. Такой подход имеет существенные вычислительные ограничения и, кроме того, фактически начинает моделировать случайный шум.

На рис. 3 приведен результат интерполяции модели модельной поверхности с расширенной областью определения и реальных данных – интерполяция угольного пласта по результатам бурения. Целевыми значениями являются глубина его залегания и толщина слоя.

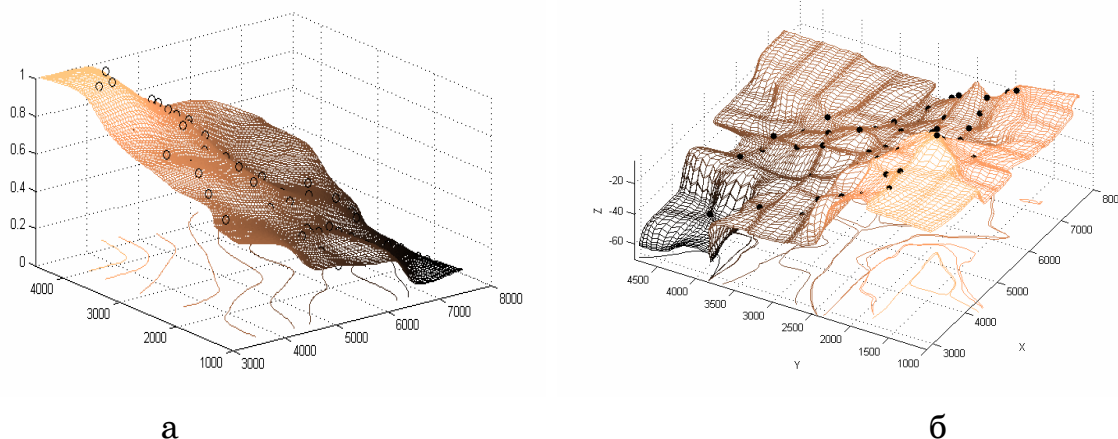


Рисунок 4 – Результаты интерполяции:
а – геофизическое поле ;б – угольный пласт

Выводы.

1. Важнейшей свойством самоорганизующихся карт Кохонена, которое обуславливает их эффективность при обработке именно пространственных данных, является то, что они обеспечивают адаптивное топологическое отображение входного пространства признаков произвольного размера в двумерное представление, в качестве которого может рассматриваться система координат.

2. Точность моделирования реальных систем на основе экспериментальных данных существенно зависит от их пространственного распределения, при этом единственным способом оценки правдоподобности процедуры восстановления является эмпирическое тестирование модели.

3. Для оценочного моделирования поверхности достаточно непосредственная визуализация кодового блокнота.

4. При малом значении неравномерных экспериментальных данных и значительном различии расстояний между ними, важно исследование на результат влияния метода их предобработки, способа инициализации начальных значений, параметров обучения сети.

ЛИТЕРАТУРА

1. Кохонен Т. Самоорганизующиеся карты / Кохонен Т. [пер. 3-го англ. изд. В.Н. Агеева под ред. Ю.В. Тюменцева]. – М.: Бином. Лаборатория знаний, 2008. – 665 с.
2. Рутковский Л. Методы и технологии искусственного интеллекта / Рутковский Л. – М., 2010. – 600 с.
3. Л.Г. Ахметшина., Т.С. Ямнич. Интерполяция пространственных данных методом двумерного проецирования нечетких кластеров // Искусственный интеллект, –2010. № 3. – С. – 433 –438.

**СИСТЕМА МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ
МЕХАНИЧЕСКОГО РАЗРУШЕНИЯ
НЕМЕТАЛЛИЧЕСКИХ ПОКРЫТИЙ ЭЛЕКТРИЧЕСКИХ
ПРОВОДОВ**

Аннотация. Описано использование системы математического моделирования (которая включает математическое моделирование, разработку конструктивно-технологических предложений и оптимизацию параметров) при исследовании механического разрушения неметаллических покрытий электрических проводов.

Утилизация электрических проводов, которые имеют неметаллические изоляционные покрытия, является актуальной хозяйственной и экологической проблемой [1]. Используемые для утилизации методы предполагают разделение металлических и неметаллических составляющих проводов и последующую их переработку для вторичного применения материалов. Большинство способов являются комбинациями различных методов [1], например, химического и механического воздействий. Применение химических воздействий требует дополнительных защитных мероприятий, которые отсутствуют при механическом разделении (разрушении покрытия).

Для исследования возможностей механического разрушения неметаллических покрытий электрических проводов целесообразно применить систему математического моделирования [2], которая включает

- математическое моделирование (целеполагание, идеализация, формализация, идентификация, проверка адекватности);
- разработку конструктивно-технологических предложений;
- оптимизацию параметров конструктивно-технологических предложений.

Рассмотрим решение такой задачи на примере механического разрушения неметаллических покрытий электрических проводов диаметром (145) мм.

Математическое моделирование**Целеполагание**

Создать математические модели (ММ), учитывающие основные физико-механические и геометрические особенности неметаллических покрытий электрических проводов, для определения оптимальных параметров процесса их механического разрушения. Погрешность моделирования не более 5%. Стоимость моделирования не должна превышать 4,75% стоимости проводов определенного вида.

Идеализация

Упрощенная схема разделения полимерной изоляции (рис. 1) предполагает воздействие сжимающей нагрузки, величина которой достаточна для пластического (необратимого) деформирования изоляции 1, при этом металлический провод 2 не деформируется.

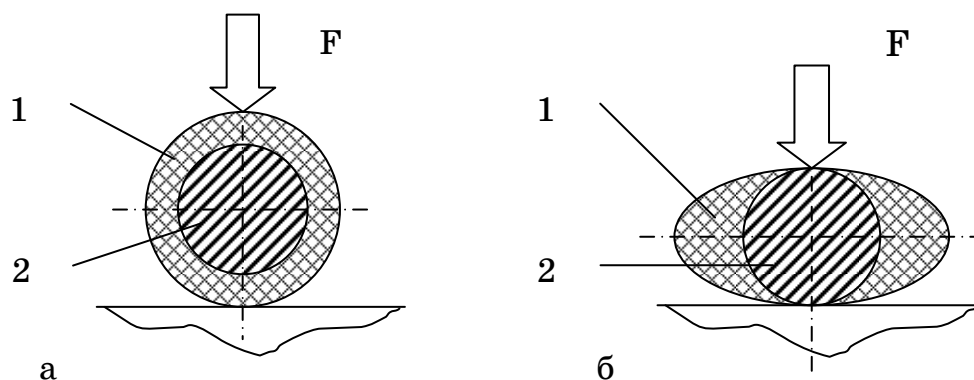


Рисунок 1 - Схема разделения изоляции
(а - начало процесса; б - окончание процесса)

Эта схема разделения имеет основные характерные свойства [3, 4] продольной симметричной прокатки. Отличительными особенностями предлагаемой схемы является комбинация признаков продольной прокатки полос [3] (валки имеют цилиндрическую форму) и продольной прокатки труб [4] (обрабатываемый материал можно рассматривать как полимерную "трубу" на цилиндрической "оправке" - металлическом проводе).

Указанная аналогия кинематических, энергосиловых и деформационных процессов позволяет применять основные результаты теории прокатки для исследования процесса разделения полимерной изоляции проводов.

Формализация

Для принятой схемы разделения в качестве основных входных

переменных примем (рис. 2) радиус (R) и частоту вращения (n) рабочих валков, а в качестве выходных переменных необходимую для реализации процесса мощность (W), усилие прижима валков (F), производительность процесса (V), а также степень деформации валков (Δ).

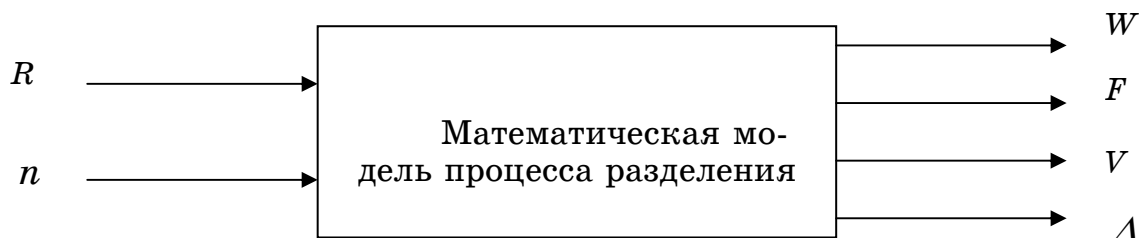


Рисунок 2 - Общий вид математической модели.

Оценкой мощности, необходимой для реализации процесса, может быть выражение [3] для мощности, расходуемой на бочке валков:

$$W_{np} = \frac{2M_{np}}{R} \cdot v_b \quad (1)$$

где M_{np} – крутящий момент для одного валка;
 v_b – окружная скорость валков.

Крутящий момент для одного валка можно определить по формуле [3]:

$$M_{np} = p_{cp} \psi b R \Delta h \quad (2)$$

где p_{cp} – среднее контактное давление;
 ψ – коэффициент плеча момента;
 b – ширина полосы;
 Δh – абсолютное обжатие.

При равномерном распределении давлений по дуге контакта рекомендуется [3] коэффициент плеча момента принимать $\psi=0,5$. Учитывая комбинированный характер (сочетание прокатки полосы и трубы) технологической схемы, примем, что "ширина полосы" (b) оценивается величиной, равной половине исходного диаметра изоляции. Абсолютное обжатие (Δh) определяется как разность между диаметрами недеформированной изоляции (d_0) и металлического провода (d_1):

$$\Delta h = d_0 - d_1. \quad (3)$$

Выражения для среднего контактного давления по дуге контакта [3], полученные Целиковым А.И. (4) и Чекмаревым А.П. (5), содержат большое количество переменных.

$$p_{cp} = \sigma_T^* \frac{2h_1}{\Delta h(\delta - 1)} \left(\frac{h\gamma}{h_1} \right) \left[\left(\frac{h\gamma}{h_1} \right)^\delta - 1 \right] \quad (4)$$

$$p_{cp} = \sigma_T^* \left\{ 1 + \frac{R \cdot \alpha}{2h_{cp}} \left[f_\sigma - \alpha \left(\frac{1}{3} + \frac{\alpha}{4f_\sigma} \right) \right] \right\} \quad (5)$$

Для определения величины среднего контактного давления используем более простое выражение [3]:

$$p_{cp} = 1,15 \sigma_T n_\sigma \quad (6)$$

где n_σ - коэффициент напряженного состояния;

σ_m - предел текучести материала.

Значения пределов текучести для основных видов полимерной изоляции приведены в таблице 2.2. Для определения величины коэффициента напряженного состояния (n_σ) рекомендуется [3] использовать диаграмму и две предварительно рассчитанные величины:

$$\delta = \frac{f_y}{\operatorname{tg}(\alpha/2)}, \quad \varepsilon = \frac{\Delta h}{d_0}, \quad (7)$$

где $\alpha = \sqrt{\Delta h/R}$ - угол контакта валка и прокатываемого материала.

f_y - коэффициент трения при установившемся режиме прокатки.

кружная скорость валков (v_b) определяется выражением:

$$v_b = \frac{2 \cdot \pi \cdot R \cdot n}{60}. \quad (8)$$

Усилие прижима валков (F), которое определяет возможность реализации разделения полимерной изоляции по принятой технологической схеме, обуславливается величиной усилия прокатки [3]:

$$F \geq P = p_{cp} \cdot l_d \cdot b, \quad (9)$$

где $l_d = \sqrt{R \cdot \Delta h}$ - длина очага деформации.

Производительность процесса разделения (V) определяется количеством обработанной кабельной продукции в единицу времени и

зависит от окружной скорости (v_b) валков (эффектом "опережения" пренебрегаем) и площади сечения металлического провода с диаметром d_1 :

$$V = v_b \cdot \frac{\pi \cdot d_1^2}{4}. \quad (10)$$

Степень деформации валков (Δ) определяется величиной различия длины очага деформации с учетом эффекта "сплющивания" валков:

$$\Delta = l_d - l_{dc}, \quad (11)$$

где l_{dc} - длина очага деформации с учетом "сплющивания"

$$l_{dc} = \sqrt{R\Delta h + (cmRp_{cp})^2} + cmRp_{cp}; \quad (12)$$

$$m = \frac{8(1 - \mu^2)}{\pi E}, \quad (13)$$

μ и E - коэффициент Пуассона и модуль упругости материала валков; $c = 1,375$ - поправочный коэффициент.

Математическая модель процесса разделения полимерного материала изоляции включает формулы (1) - (13), описывающие зависимость выходных переменных процесса (W , F , V , Δ) от входных переменных (R , n) и физико-механических свойств (σ_T , μ , E) и геометрических характеристик обрабатываемого материала (d_0 , d_1).

Идентификация

Для определения величин предельных сил, при которых происходит гарантированное разделение полимерной изоляции в соответствии со схемой разделения (рис. 1), выполнены экспериментальные исследования для нескольких основных разновидностей изоляции проводов.

Эксперименты проведены на испытательной установке ИМ-4Р предназначенной для определения механических характеристик материалов.

Для проведения экспериментов использованы по 4 образца каждого вида проводников длиной 30 мм (размер определяется рабочей зоной испытательной установки). Один образец служил для предварительного определения величины усилия разделения (без деформирования металлического провода). Основные эксперименты проводи-

лись на трех образцах проводников. Величина усилия разделения полимерной изоляции фиксировалась с помощью механического динамометра с ценой деления 50Н по шкале 20000Н. Результаты измерения, обработанные традиционными статистическими приемами (расчет оценок математического ожидания и среднего квадратического отклонения), приведены в таблице 1.

Величины усилий разделения для разных групп материалов (резина, поливинилхлоридный пластикат, полиэтилен) существенно различаются, что объясняется их физико-механическими свойствами [5] (таблица 2).

Между усилием разделения (F_p) и показателями механических свойств, в частности, пределом прочности (σ_b), существует сильная корреляционная связь (коэффициент линейной корреляции составляет 0,997). Линейная регрессионная модель (коэффициенты регрессии определены методом наименьших квадратов с помощью стандартных программ для ЭВМ), описывающая эту связь, имеет вид:

$$F_p = -9705.125 + 695.875 \cdot \sigma_b. \quad (14)$$

Таблица 1

Результаты экспериментов

Номер вида образца	Вид изоляции	Усилие разделения, Н	
		Среднее	Среднее квадратическое отклонение
1	Резина	2333	471,4
2	Резина	1166	235,7
3	Резина	2583	311,8
4	Поливинилхлоридный пластикат	5333	623,6
5	Поливинилхлоридный пластикат	3667	311,8
6	Резина	1833	235,7
7	Резина	1750	204,1
8	Полиэтилен	7500	408,2
9	Полиэтилен	8917	360,0
10	Полиэтилен	7500	408,2
11	Полиэтилен	9250	204,1

Основные свойства полимерных материалов изоляции [5]

Вид изоляции	Модуль упругости, МПа	Предел прочно- сти при разрыве, МПа
<i>Полиэтилен</i>	550 - 800	22 – 30
Полипропилен	1050 - 1260	30 – 35
Полистирол	2800 - 3500	35 – 60
Фторопласт	470 – 850	16 – 25
Полиамид	1600	80 - 110
Поливинилхлоридный пластикат	10 – 20	18 – 23
Резина	5 - 10	15 - 20

С помощью зависимости (14) можно рассчитывать усилия разделения для рассмотренных групп полимерных материалов.

Проверка адекватности

Перед использованием математической модели для исследования процесса разделения и оптимизации его параметров необходимо проверить ее соответствие оригиналу [6]. Учитывая, что одной из основных выходных переменных модели и оригинала является усилие разделения полимерного материала, проверку соответствия выполним по этому показателю. Для этого используем результаты экспериментального определения величин усилий разделения полимерных материалов.

Предельные допустимые отклонения значений выходных переменных оригинала ξ_{max} и ξ_{min} определены [6] по формулам (15, 16) с использованием статистических оценок математических ожиданий ξ и средних квадратических отклонений s экспериментальных данных для групп материалов. Значения статистики $\zeta(n, Q)$ определены по таблице 4.8в [8] при $Q=5\%$.

$$\xi_{max} = \xi + \zeta(n, Q) \cdot s \quad (15)$$

$$\xi_{min} = \xi - \zeta(n, Q) \cdot s \quad (16)$$

Соотношение рассчитанных с помощью математической модели и экспериментальных значений усилий разделения (таблица 3) показывает, что гипотеза о соответствии модели и оригинала не может

быть отклонена [6].

Таблица 3

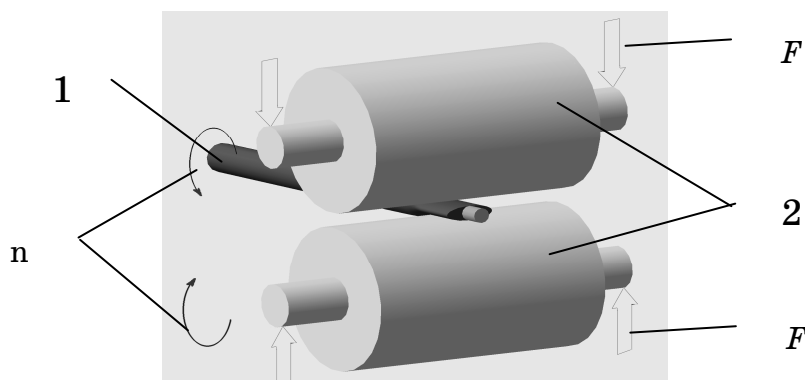
Проверка соответствия модели и оригинала

Вид полимера	Математическая модель разделения	ξ_{max}	ξ_{min}
Резина	2708.93	2989.21	876.79
Поливинилхлоридный	3186.97	6165.75	2834.25
Полиэтилен	6845	9872.28	6711.22

Разработка конструктивно-технологических предложений

Для реализации схемы разделения (рис. 1) с учетом обеспечения требуемой производительности процесса предполагается исходный проводниковый материал (1) пропускать через два вращающихся с одинаковой скоростью валка (2) равного диаметра (рис. 3).

Рисунок 3 - Технологическая схема разделения изоляции (общий вид)



Оптимизация параметров конструктивно-технологических предложений

Критерий оптимизации и ограничения формируются из основных выходных переменных математической модели процесса разделения. Учитывая общие тенденции создания технических систем и особую важность уменьшения затрат энергии при их функционировании, примем в качестве критерия оптимизации мощность (W), необходимую для реализации процесса разделения. Другие выходные переменные (F , V , Δ) образуют систему ограничений в формулировке задачи оптимизации (17).

Усилие прижима валков должно быть не меньше усилия прокатки (P), что обеспечивает гарантированную величину расстояния (равное диаметру металлического провода) между рабочими поверх-

ностями валков. Производительность процесса не должна быть меньше заданной производственной программы ($[V]$), а степень деформации валков ограничена заданной предельной величиной ($[\Delta]$).

$$\begin{aligned} W &\Rightarrow \min \\ F &\geq P \\ V &\geq [V] \\ \Delta &\leq [\Delta] \end{aligned} \quad (17)$$

Применим известный [7] алгоритм для решения задачи оптимизации в формулировке (17) для конкретных условий обработки.

Рассмотрим процесс разделения одного из наиболее прочных видов изоляции - полиэтилена (свойства в таблице 2) для медного проводника диаметром 1 мм (толщина изоляции 1,5 мм). Примем, что допустимая степень деформации валков составляет 400 мкм, которая незначительно влияет на процесс разделения. Величина усилия прижима валков задается большей, чем соответствующая величина усилия прокатки. Предельная производительность процесса (3,5 тонны в месяц, что соответствует для медного проводника величине объемной производительности $5,462 \cdot 10^{-7}$ м³/с) принята, исходя из прогнозных данных об объемах перерабатываемой кабельной продукции. Начальная область (таблица 4) для входных переменных (диаметра валка - D и частоты его вращения - n) принята с учетом имеющегося производственного опыта.

Формулировка задачи оптимизации для этих условий имеет вид (18):

$$\begin{aligned} W &\Rightarrow \min \\ F &\geq P \\ V &\geq 5,462 \cdot 10^{-7}; \\ \Delta &\leq 0,0004. \end{aligned} \quad (18)$$

Для определения параметров градиента критерия оптимизации проведены расчеты с помощью ЭВМ по формулам (1) – (13), результаты которых приведены в таблице 4. Шаг движения в направлении градиента критерия оптимизации уменьшен в 50 раз для обеих вход-

ных переменных для обеспечения возможности изменения величин.

После нескольких шагов обнаруживается (строка 8 таблицы 4) нарушение ограничения по производительности процесса. В соответствии с алгоритмом оптимизации это является признаком достижения оптимума. В этом случае оптимальными признается сочетание значений входных переменных для предшествующего шага (строка 7 таблицы 4), а именно:

- диаметр валка $D = 0,43$ м;
- частота вращения валка $n = 31,4$ об/мин.

Использование системы математического моделирования, которая включает

- математическое моделирование (целеполагание, идеализация, формализация, идентификация, проверка адекватности);
- разработку конструктивно-технологических предложений;
- оптимизацию параметров конструктивно-технологических предложений,

позволяет эффективно решать практические задачи повышения эффективности механического разрушения неметаллических покрытий электрических проводов.

Таблица 4

Ход и результаты оптимизации

	Диаметр валков (D), мм	Частота вращения валков (n), об/мин	Мощность (W), Вт	Усилие (F), Н	Производительность (V), м ³ /с	Степень деформации (Δ), м
Верхний уровень	0,55	45				
Нижний уровень	0,5	40				
1	+	+	1081,08	8026,552	1,01475E-06	0,000467
2	-	+	982,8	7653,017	9,225E-07	0,000416
3	+	-	960,96	8026,552	0,000000902	0,000467
4	-	-	873,6	7653,017	0,00000082	0,000416
Параметры градиента						
	46,41	57,33				
Шаг движения по градиенту						

1 (84) 2013 «Системные технологии»

	0,023205	2,8665				
Движение по градиенту						
5	0,476795	37,1335	773,357	7473, 319	7,25908E -07	0,000392
6	0,45359	34,267	678,926	7289, 192	6,3727E- 07	0,000368
7	0,430385	31,4005	590,305	7100, 292	5,54086E -07	0,000345
8	0,40718	28,534	507,495	6906, 227	4,76357E -07	0,000321

ЛИТЕРАТУРА

1. Материалы Международной конференции «Сотрудничество для решения проблемы отходов», 28–31 марта 2012 г., г. Харьков, Украина
2. Ясев А.Г. Комплексный метод совершенствования технологической оснастки процессов обработки деталей металлургических машин // Сучасні проблеми металургії. Наукові праці. Том 10. Дніпропетровськ: Системні технології, 2007. – с. 68-82
3. Грудев А.П. Теория прокатки. – М.: Металлургия, 1988, 240с.
4. Данченко В.Н., Чус В.С. Продольная прокатка труб. - М.: «Металлургия». 1984. 136с.
5. Гуль В.Е., Кулезнев В.Н. Структура и механические свойства полимеров. М., «Высшая школа», 1972.
6. Ясев А.Г. Соответствие математических моделей и технологических процессов в металлургии и машиностроении. - Днепропетровск: Днепр-VAL, 2001. - 237 с.
7. Ясев А.Г. Применение методов оптимизации технологических процессов. С. 163-217 // Физико-математическая теория процессов обработки материалов и технологии машиностроения / Под общей редакцией Ф.В. Новикова и А.В. Якимова. В десяти томах. - Т. 8: «Оптимизация технологических процессов в машиностроении» - Одесса: ОНПУ, 2004. - 509 с.
8. Большев Л.Н., Смирнов Н.В. Таблицы математической статистики.- М.: Наука, 1983. - 416 с.

В.М. Малкіна, А.О. Бездітний

ВИКОРИСТАННЯ MECHANICAL APDL ДЛЯ РОЗРАХУНКУ ВЛАСНИХ КОЛИВАНЬ ЛОПАТОК ТУРБІН У СИСТЕМІ КІНЦЕВО-ЕЛЕМЕНТНОГО АНАЛІЗУ ANSYS WORKBENCH

Анотація. У статті наводиться процес проведення модального аналізу радіальної крильчатки засобами Ansys Workbench з використанням параметричної мови програмування APDL (Ansys Parametric Design Language).

Ключові слова: параметрична мова програмування APDL, крильчатка, Ansys Workbench.

Постановка проблеми

Особливості геометрії лопаток турбіни необхідність точної її передачі роблять цей виріб надзвичайно складним для проведення моделювальних робіт. Мова параметричного програмування Ansys APDL для програми Ansys Mechanical є основним засобом розрахунку турбінних лопаток завдяки широкому функціональному набору, опрацьованості і можливості застосування потужних інструментів. У статті наводиться приклад дослідження крильчатки засобами однієї з останніх версій зазначеного програмного комплексу.

Формулювання цілей статті

Метою статті є проведення модального аналізу лопатки турбіни засобами програмного комплексу Ansys з використанням параметричної мови програмування APDL.

Основна частина

APDL – це параметрична мова програмування універсальної програмної системи кінцево-елементного аналізу Ansys, мова сценаріїв, яка дає можливість автоматизувати стандартні задачі та створювати власні моделі, виражені через параметри (змінні). APDL також охоплює широкий діапазон можливостей – таких, як повторення команди, макросу, виконання переходу "якщо - тоді-інакше", створення циклів, а також скалярних, векторних і матричних операцій. APDL

також є основою для складних завдань, типу оптимізаційного розрахунку та адаптивної побудови сітки.

Проведемо модальний аналіз лопатки турбіни засобами AnsysWorkbench. Аналіз вільних механічних коливань (модальний аналіз) проводиться без врахування динамічних навантажень, але є першим і обов'язковим кроком під час розв'язання більш складних динамічних задач.

Для досягнення поставленої мети створимо проект у середовищі AnsysWorkbench, оберемо модальний тип аналізу, імпортуємо тривимірну модель лопатки (рис.1), створимо сітку на цій моделі та налаштуємо закріплення.

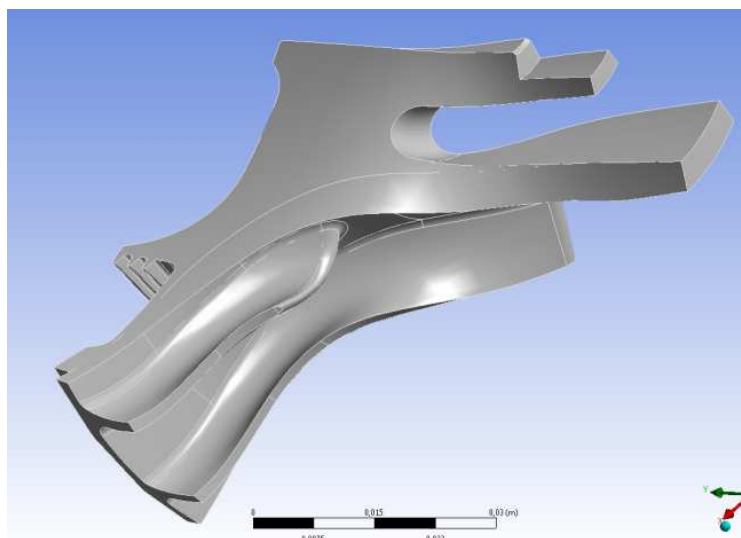


Рисунок 1 - Модель досліджуваної лопатки

Додаємо команду «CYCLIC» до модального аналізу (активує метод циклічного розв'язання симетричних задач у Mechanical APDL). Вводимо команди до меню Solver:

```
/view,1,0,-1,0  
/show,png  
/graph,power  
/efacet,2  
/plopt,info,3  
/udoc,1,date,0  
/rgb,index,100,100,100, 0  
/rgb,index, 80, 80, 80,13  
/rgb,index, 60, 60, 60,14  
/rgb,index, 0, 0, 0,15
```

/сусехpand,1,on

чим налаштуємо відображення графіки для побудови діаграм та включаємо відображення повної моделі (360°). Проводимо розрахунки, після чого передаємо отримані результати до Mechanical APDL (рис.2).

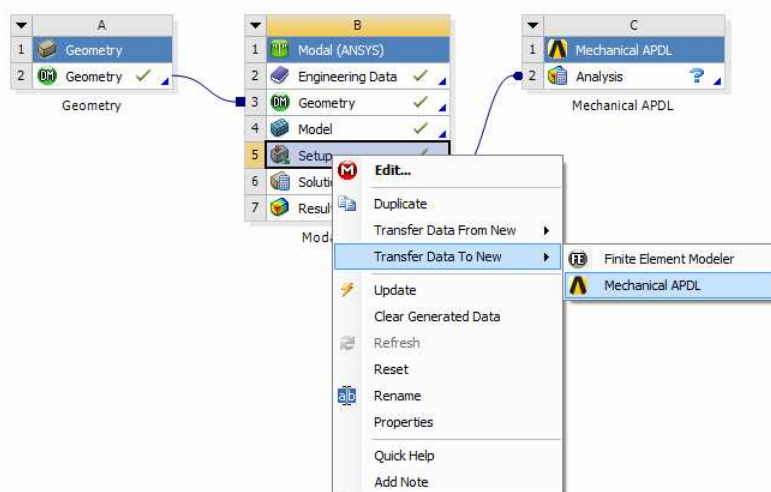


Рисунок 2 - Схема проекту у Ansys Workbench

Після запуску Mechanical APDL активуємо відображення повної моделі крильчатки.

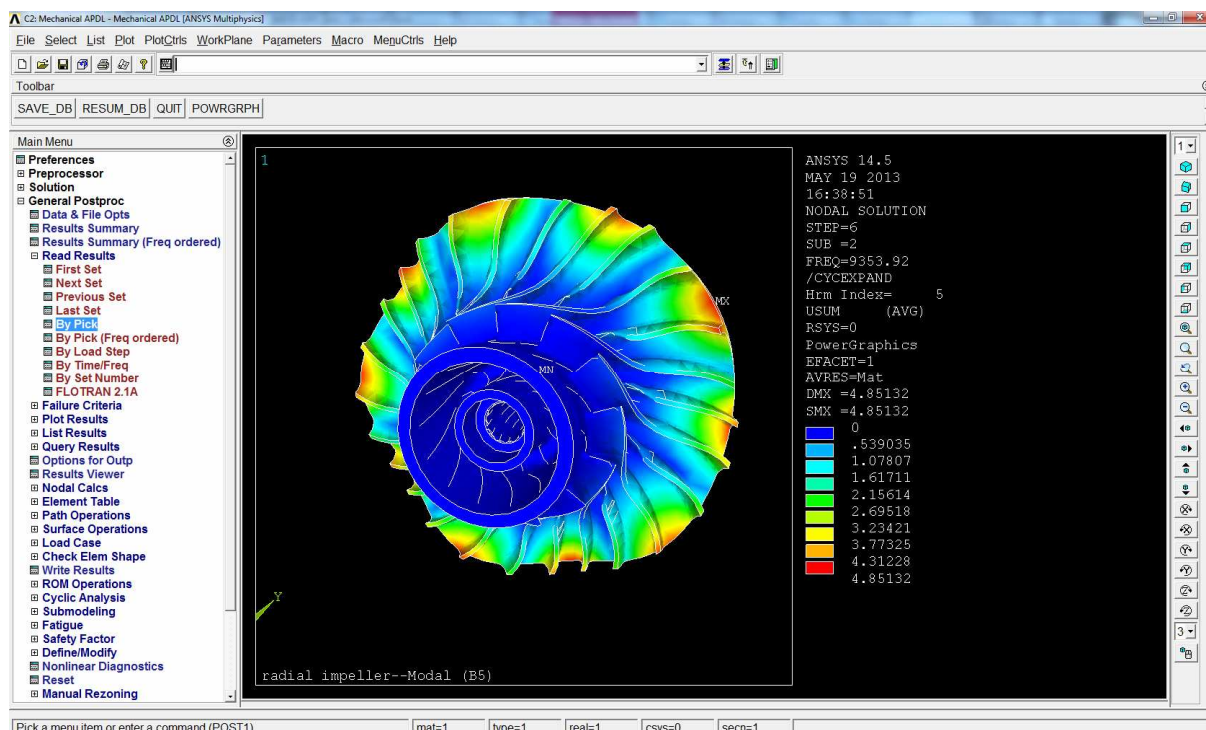


Рисунок 3 - Відображення крильчатки у Mechanical APDL

Далі ми переходимо до відображення результатів моделювання, генерації кольорових діаграм (рис.3) та створенню анімації процесу деформації крильчатки у результаті дії на неї внутрішніх коливань у постпроцесорі Mechanical APDL.

Висновок.

В роботі запропоновано варіант розрахунку вільних механічних коливань лопатки турбіни засобами параметричної мови програмування APDL на базі Ansys Modal, що дозволило отримати власні частоти та форми коливань виробу. Можна зробити висновок, що використана система кінцево-елементного моделювання цілком придатна для застосування її при розв'язанні задач моделювання та дослідження виробів, що мають складну геометрію.

ЛІТЕРАТУРА

1. Басов К.А. ANSYS: справочник пользователя / К.А. Басов // М.: ДМК пресс, 2005. – 640 с.
2. Конюхов А.В. Основы анализа конструкции в ANSYS / А.В. Конюхов // 2001. – с. 102.
3. Бруйка В.А. Инженерный анализ в Ansys Workbench / В.А. Бруйка, В.Г. Фокин, Е.А. Солдусова, Н.А. Глазунова, И.Е. Адеянов. // Самара: Самар. гос. техн. ун-т, 2010. – 271 с.
4. Барулина М.А. Использование ANSYS WORKBENCH для работы с геометрическими моделями. Монография / М.А. Барулина // М.: Эдитус, 2012. – 316 с.

В.В. Спірінцев, І.В. Пихтеева, Ю.О. Дмитрієв
**РОЗРОБКА ФУНКЦІОНАЛЬНОЇ СХЕМИ ПРОЦЕСУ
АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ**

Анотація. В роботі розглянуто питання розробки функціональної схеми процесу автоматизованого проектування з врахуванням сучасних умов виробництва.

Постановка проблеми

Зараз на світовому ринку наукомістких промислових виробів чітко спостерігаються три основні тенденції: підвищення складності й ресурсоемності виробів, підвищення конкуренції на ринку й розвиток кооперації між учасниками життєвого циклу виробу. Найбільш прогресивною та перспективною умовою удосконалення процесу проектування є створення і впровадження в практику систем автоматизованого проектування (САПР), забезпечених сучасними ПЕОМ з розвиненими термінальними системами. Автоматизація підготовки виробництва дає можливість підприємствам швидко реагувати на зміну попиту, у короткий термін випускати нові види продукції, швидко модернізувати випускаєму продукцію, відслідковувати життєвий цикл виробів, ефективно підвищувати якість.

Аналіз останніх досліджень

Сучасний ринок програмного забезпечення автоматизації підготовки виробництва насичений найрізноманітнішими універсальними САПР, що здатні істотно полегшити роботу проектувальника. Разом з тим, не дивлячись на величезну кількість такого виду інструментальних засобів автоматизації інженерної діяльності, універсальні системи часто недостатньо ефективні для вирішення конкретного завдання користувача, зачасту інформаційно несумісні. Виникає необхідність в розробці узагальненої функціональної схеми процесу автоматизованого проектування.

Формулювання цілей статті

Метою статті є розробка функціональної схеми процесу автоматизованого проектування з врахуванням сучасних умов виробництва.

Основна частина

У відповідності з прагненням підприємств досягнути значних фінансових результатів, підвищити прибутковість, розширити та поглибити ринок збуту своєї продукції – основною метою для них є виготовлення продукції з максимальними споживацькими якостями при зниженні собівартості продукції та скороченні термінів проектування і виробництва нової продукції.

В даній роботі запропонована функціональна схема автоматизованого проектування (рис.1), що дозволяє підвищити продуктивність праці проектувальників, скоротити терміни проектування та витрати на розробку технічної документації, збільшити кількість річних проєктів.



Рисунок 1 – Функціональна схема автоматизованого проектування

Дієздатність запропонованої підсистеми проектування (зокрема підсистеми проектування конструкторської документації) розглянемо на прикладі деталі Пуансон.

Після створення 3-D моделі деталі Пуансон проведемо статичний аналіз її напружено-деформованого стану за допомогою програмної системи кінцево-елементного аналізу ANSYS. Використовуючи майстер розрахунків, проведемо конструкційний аналіз, знайдемо напруги, деформацію й запас міцності.

Починаємо роботу з визначення матеріалу аналізованої деталі - у нашому випадку це сталь. Далі імпортуємо модель у форматі *.x_t в ANSYS і розбиваємо її на кінцеві елементи із заданою щільністю й величиною елемента (рис.2). Далі варто визначити закріплюючі області Пуансона (рис.3). У нашому випадку це основа деталі.

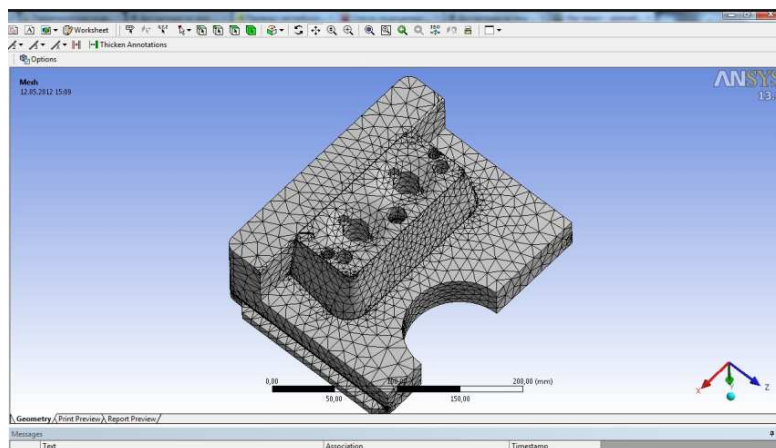


Рисунок 2 – Кінцево-елементна модель

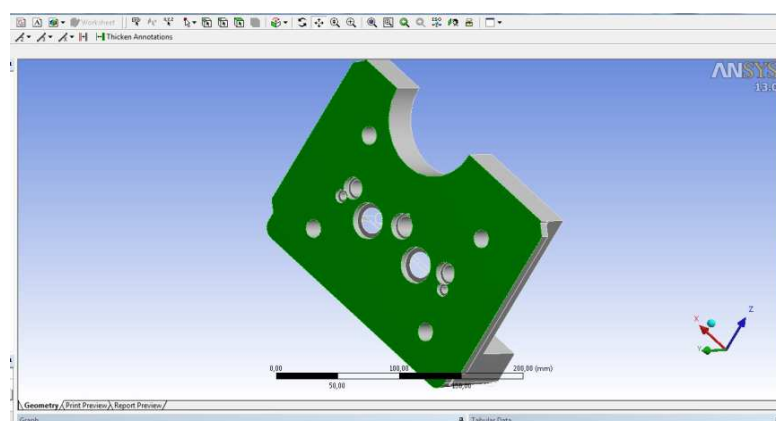


Рисунок 3 - Визначення закріплення

Представимо взаємодію деталі із зовнішніми об'єктами, використовуючи тип без тертя (frictionless support). При закріпленні без тертя забороняється переміщення по нормалі до поверхні контакту. Дозволено переміщення у всіх напрямках за винятком нормалі до площини закріплення. В області контакту передбачається дія сил без тертя, тобто консервативне поведіння системи (без розсіювання енергії).

Тепер варто скласти схему навантаження Пуансона (рис.4). Варто думати, що основні навантаження будуть доводитися на верхні поверхні Пуансона (по нормалі до основи). Виходячи із цього, задамо

тип навантаження тиск (pressure) на всі верхні поверхні Пуансона. Залишається лише додати інформацію про те, які розрахункові параметри ми хотіли б переглянути в графічному вікні. У нашому випадку досить буде деформації (total deformation) – рис.5.

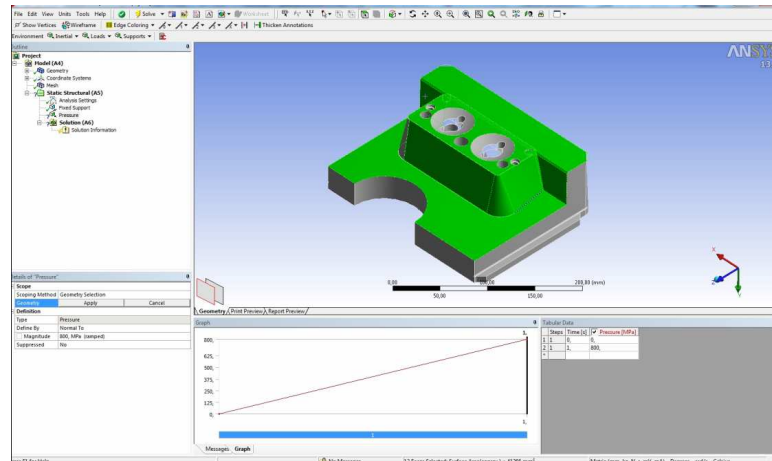


Рисунок 4 – Схема навантаження

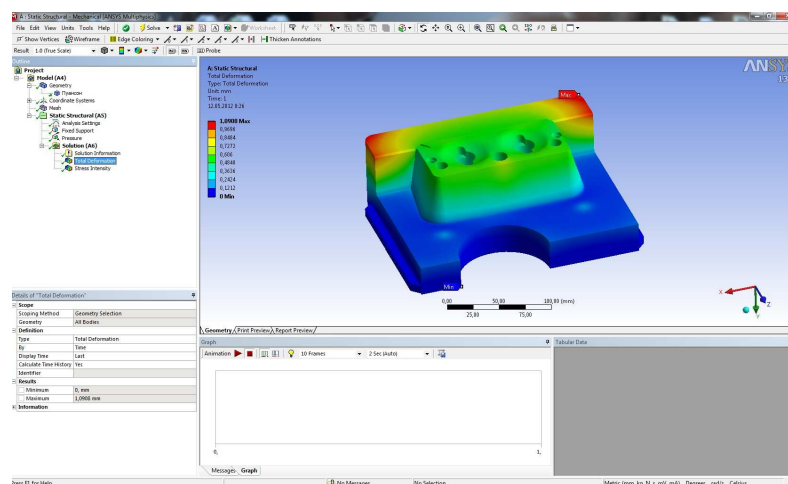


Рисунок 5 – Total deformation

В результаті розрахунків було виявлено, що гранична деформація Пуансона (тильна поверхня) в 1 мм досягається в результаті додавання до його тильної поверхні навантаження в 800 мПа. Тому було вирішено потовщити на 2 мм (рис.6) та випробувати модифіковану конструкції деталі Пуансон.

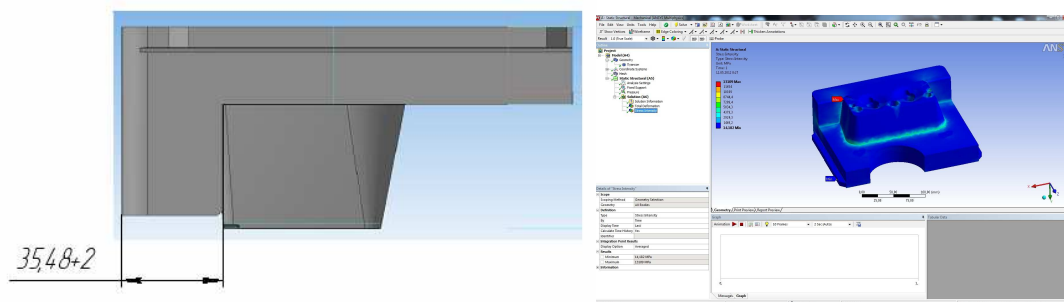


Рисунок 6 – Результат випробувань модифікованої конструкції деталі

Більшість застосовуваних у промисловості тривимірних САПР зазвичай вирішують задачі усередненого підприємства без врахування його специфіки. Оскільки номенклатура виробів постійно оновлюється на базі вже розроблених вузлів, то в більшості випадків робота проектувальника зводиться до модифікації раніше створеної геометрії вузла відповідно до нових розрахункових даних. Тому для ефективної роботи підприємства необхідно розширити бібліотеку нових виробів за допомогою технології API (Application Program Interface). В даному випадку спеціалізована САПР повністю виконує завдання автоматизації конструкторської праці, виконуючи й розрахунок, і побудову моделі.

Згідно із запропонованою методикою, що була розглянута в роботах [2-4] розроблена спеціалізована САПР за допомогою спеціально створеного програмного модуля на базі мови програмування Delphi 7 з використання технології API КОМПАС (рис.7) та API AutoCAD (рис.8) для розширення бібліотеки нових виробів в системі автоматизованого проектування та підвищення автоматизації проектних процедур.

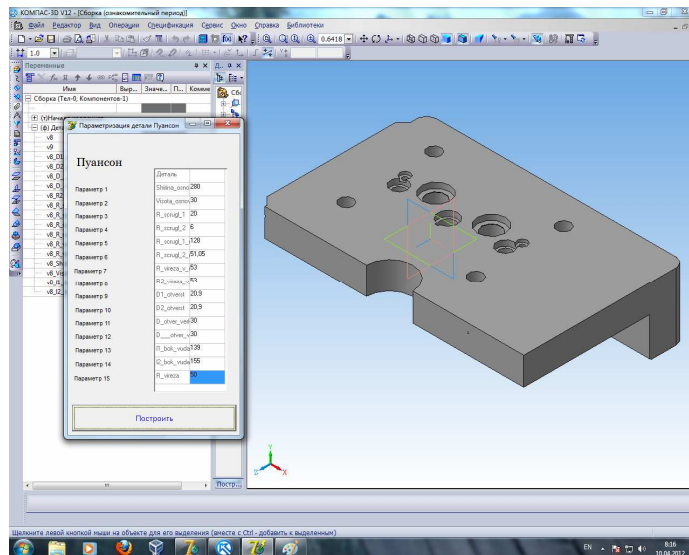


Рисунок 7 – Вікно програмного модулю на базі API КОМПАС

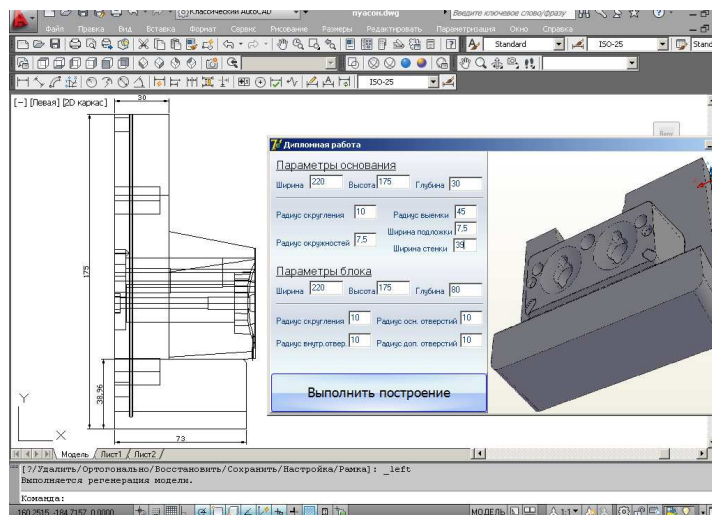


Рисунок 8 - Работа программного модуля на базе API AutoCAD

Висновок. В роботі запропоновано функціональну схему процесу автоматизованого проектування (з врахуванням сучасних умов виробництва), що дозволяє підвищити продуктивність праці проектувальників, скоротити терміни проектування та витрати на розробку технічної документації, збільшити кількість річних проектів. На прикладі деталі Пуансон було перевірено дієздатність запропонованої підсистеми проектування конструкторської документації, зокрема: проведемо статичний аналіз напружено-деформованого стану за допомогою програмної системи кінцево-елементного аналізу ANSYS та розроблено спеціалізовану САПР з використання технології API КОМПАС та API AutoCAD.

ЛІТЕРАТУРА

1. Троицкий Д.И. Методические указания для выполнения лабораторных работ «Создание САПР на базе КОМПАС 3D»/ Д.И. Троицкий//Т.: ГОУ ВПО ТГУ, 2007.
2. Спирінцев В.В. Розробка спеціалізованого програмного модуля для проектування типових деталей/ В.В. Спирінцев// Збірник праць XIV Міжнародної науково-практичної конференції «Сучасні проблеми геометричного моделювання» – Мелітополь: ТДАТУ, 2012. – с. 103-107.
3. Радченко А.К., Пихтеева І.В. Автоматизація процесу побудови моделі на базі створення АРІ програми / А.К.Радченко, І.В.Пихтеева // Інформаційні технології в прикладній геометрії. Праці ТДАТУ– Вип.5, т. 6. – Мелітополь: ТДАТУ, 2013. – с. 125-131.

МОДЕЛИРОВАНИЕ ПРОЦЕССА ФОРМИРОВАНИЯ МИКРОСТРУКТУРЫ МЕТАЛЛИЧЕСКИХ СПЛАВОВ

Аннотация. Работа посвящена имитационному моделированию процесса формирования микроструктуры металлических сплавов. Исследуется процесс формирования микроструктуры и влияние формы микрокристалла на структурные характеристики металлических сплавов.

Ключевые слова: моделирование, форма микрокристалла, металлические сплавы, фрактальная размерность.

Постановка проблемы. На современном этапе развития техники при производстве металла для изделий ответственного назначения уже недостаточно обеспечить высокий уровень чистоты металлов и сплавов по вредным примесям. Актуальной становится задача получения слитков с высокой физической и структурной однородностью. Решение которой, осуществляется на этапе разлива и кристаллизации металла.

В процессе производства крупных отливок и слитков, для подавления и предотвращения образования дефектов кристаллизационного, усадочного и ликвационного характера используют технические приемы, которые позволяют управлять качеством металла непосредственно в процессе затвердевания [1].

Разработана имитационная модель процесса формирования микроструктуры металлических сплавов, описывается следующим выражением:

$$m\ddot{x} + c_0\dot{x} = F(x, u), \quad (1)$$

где x - координаты центров кристаллизации зародышей, $F(x, u)$ - сумма всех сил действующих на частицу, u - вектор входных параметров, c_0 - коэффициент сопротивления – величина обратная подвижности B .

Сумма сил, действующих на микрокристалл:

$$F = F_T + F_A + F_U, \quad (2)$$

где F_T – сила тяжести, действующая на микрористалл, F_A – сила Архимеда, F_U – сила потенциального взаимодействия, действующая между микрористаллами.

Образование микрористаллов происходит на первом этапе процесса кристаллизации жидких металлов. При охлаждении жидкого металла начинается образование центров кристаллизации, в которых атомы металла образуют микрористалл, имеющий кубическую, призматическую, ромбоэдрическую форму.

Если увеличить скорость охлаждения металлов, больше возникнет центров кристаллизации и тем меньше будут их размеры. И наоборот, чем медленнее остывает металл (например: при кристаллизации в песчано-глинистых формах) тем зерна будут крупнее – до нескольких сантиметров. На рис. 1 представлены три основные формы микрористаллов.

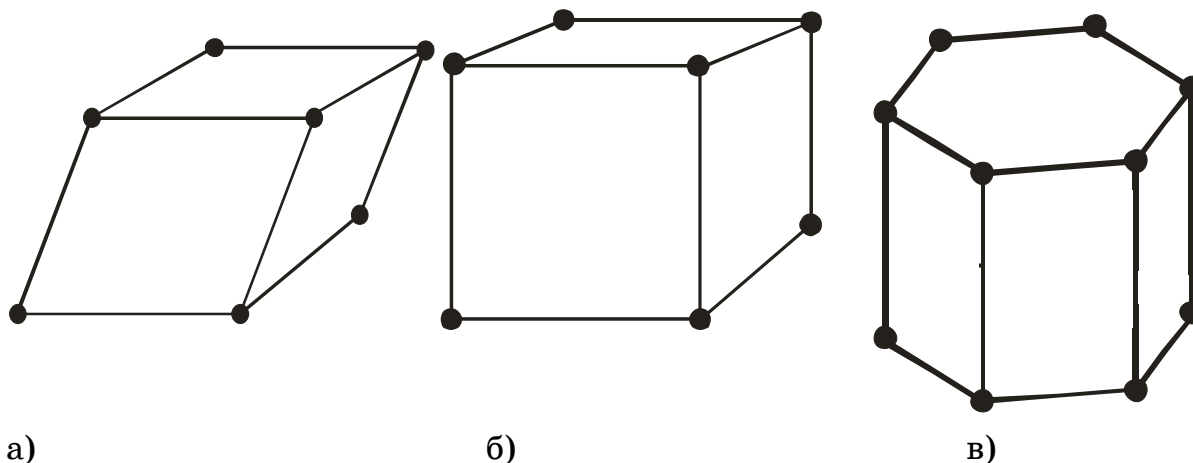


Рисунок 1 - Основные формы микрористаллов а) ромбоэдрическая, б) кубическая, в) гексагональная

Каждое кристаллическое вещество имеет определенную свойственную ему внешнюю форму микрористалла. Например, для хлорида натрия и микрористаллов железа эта форма – куб, для алюмокалиевых квасцов – октаэдр. Если процесс кристаллизации идет не слишком быстро, а частицы обладают удобной для укладки формой и высокой подвижностью, они легко находят свое место. Если же резко снизить подвижность частиц с низкой симметрией, то они застывают произвольно, образуя прозрачную массу, похожую на стекло.

Для проведения моделирования процесса структурообразования был разработан программный продукт «Colcryst – Моделирование та дослідження структури металевих сплавів». С помощью программного продукта «Colcryst» проводилось исследование процесса структурообразования металлических сплавов. «Colcryst» позволяет задавать форму микрокристалла, массу и температуру металлического сплава, и другие параметры.

По результатам исследований микроструктур металлических сплавов было определено, что форма микрокристалла влияет на структурные характеристики металлических сплавов. Учет формы микрокристалла обязателен при моделировании процесса формирования микроструктуры металлических сплавов. С помощью программного продукта «Colcryst» были проведены исследования для трех форм микрокристаллов, которые показали, что при различных формах микрокристаллов меняется интегральная характеристика - значение фрактальной размерности.

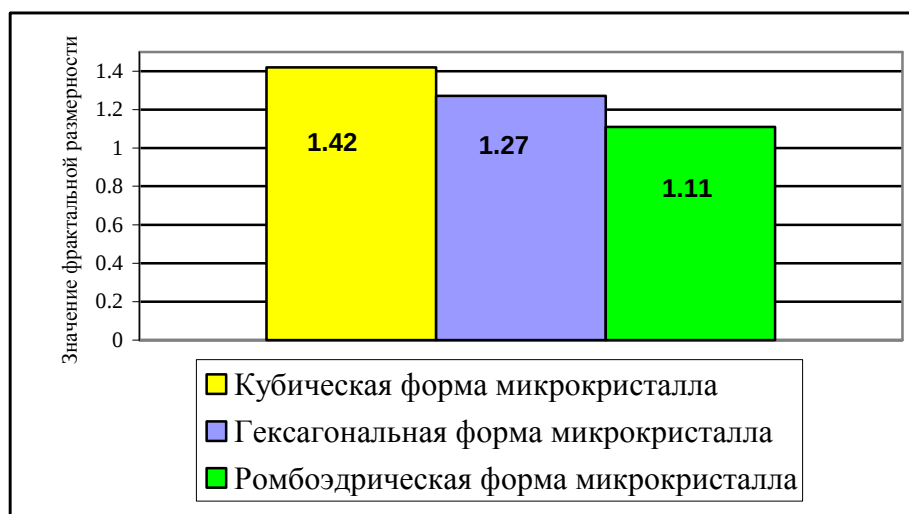


Рисунок 2 - Значение фрактальной размерности при разных формах микрокристаллов

Для трех основных форм микрокристаллов при одинаковых условиях, исследована зависимость в общем объеме количества микрокристаллов от количества микрокристаллов в образовавшихся дендритах. Исследования показали, что в зависимости от формы микрокристалла происходит формирование разного количества дендритов. На рис. 3 представлена зависимость количества микрокристаллов от количества микрокристаллов в дендрите.

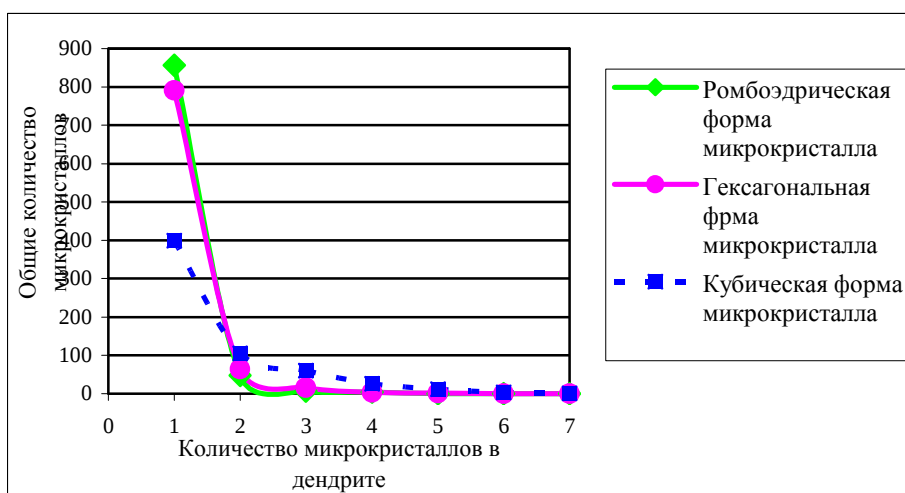
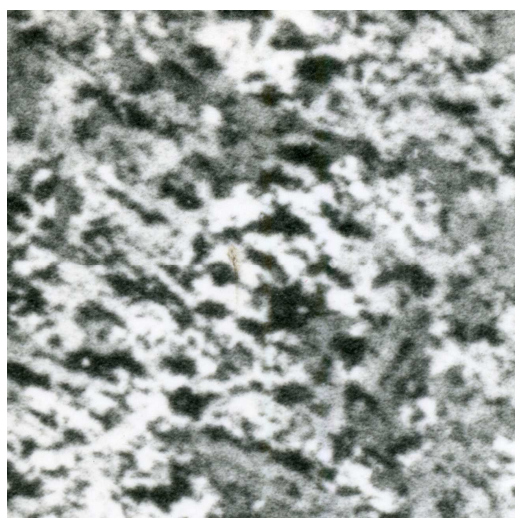


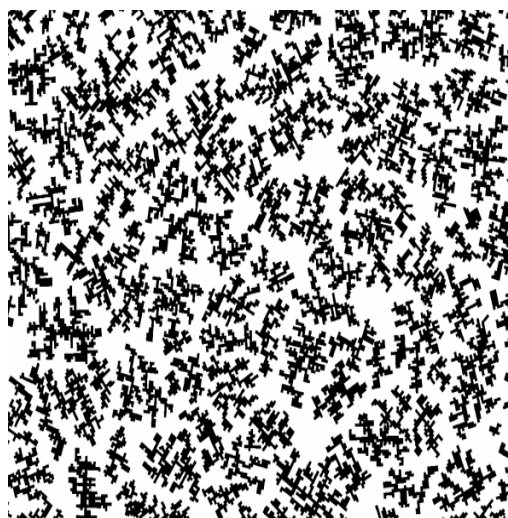
Рисунок 3 - Зависимость общего количества микрокристаллов от количества микрокристаллов в дендрите для разных форм микрокристаллов

На рисунке 4 представлены экспериментальное изображение микроструктуры металлических сплавов и изображение, полученное численным моделированием.

По серии экспериментальных изображений с помощью метода BOX COUNTING, оценены значения фрактальной размерности. Значения фрактальной размерности определялись с помощью программного продукта «MFMet» [2].



а)



б)

Рисунок 4 - Микроструктура металлического сплава: а) экспериментальное изображение, б) изображение, полученное моделированием

По значениям фрактальной размерности для экспериментальных изображений и изображений, полученных путем моделирования, построены вариационные ряды, по которым статистическими методами определен критерий χ^2 , для каждого вариационного ряда соответственно.

Определен доверительный интервал [1,682; 1,718] для экспериментальных изображений значений фрактальной размерности и [1,675; 1,697] – по результатам численного моделирования. Таким образом, пересечение этих доверительных интервалов находится в диапазоне [1,682; 1,697].

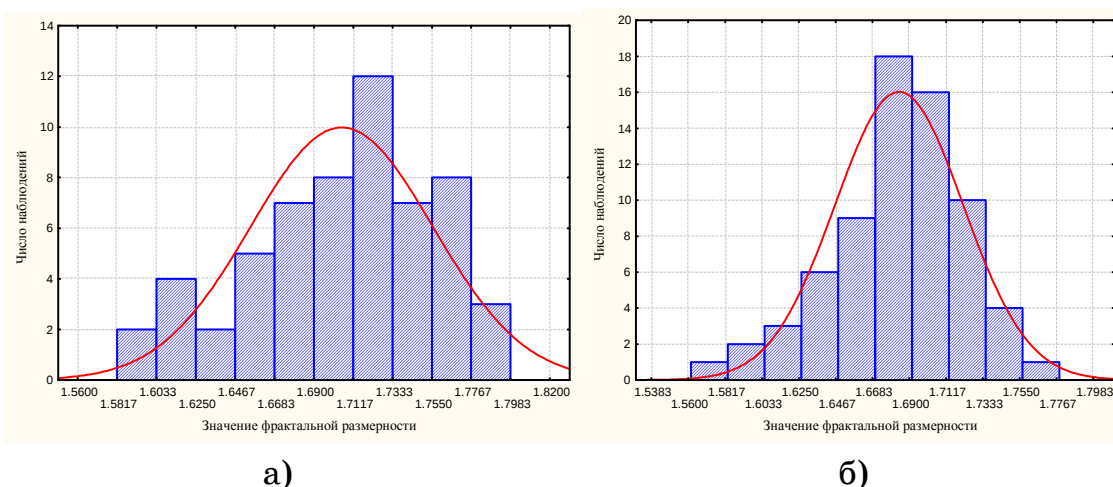


Рисунок 5 - Оценка нормальности распределения по критерию χ^2 для значений фрактальных размерностей полученных:

- а) по экспериментальным изображениям металлических сплавов,
- б) по результатам численного моделирования

Выводы и перспективы дальнейших исследований. В работе рассматривается моделирование процесса формирования микроструктуры металлических сплавов. Разработанная модель позволяет учитывать форму базовой частицы (микрокристалла) процесса структурообразования.

ЛИТЕРАТУРА

1. Ю. С. Пройдак, А. И. Дерев'янко, А. А. Кавац, Э. Б. Гальченко // «Исследование процесса формирования микроструктуры металлических сплавов с применением виброобработки» // Praca zbiorowa pod redakcją naukową Prof.dr.hab.inz. Ryszarda Budzika seria Metalurgia nr 56, Czestochowa, 2011 стр. 40-44.
2. Кавац О. О. Влияние вибровоздействия на плотность металлического сплава. / О. О. Кавац, О. И. Дерев'янко // Міжнародна наукова конференція „Математичні проблеми технічної механіки - 2009” – Дніпропетровськ - Дніпродзержинськ – 2009 – с. 229.

УДК 621.39

В.В. Гнатушенко, Пройдак Ю.С., Данладі Алі

ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ САМОПОДІБНОГО ТРАФІКА ПРИ ЗМІНЕНІ ФОРМИ ПАРЕТО-РОЗПОДІЛЕННЯ ІНФОРМАЦІЙНОГО ПОТОКУ

Анотація. Створено імітаційну модель передачі даних від джерела з самоподібним трафіком та проведені дослідження впливу параметра форми розподілення Парето.

Ключові слова: самоподібність, ON/OFF модель, інформація, трафік.

Постановка проблеми

Для ефективного вибору параметрів архітектури мережі, розробки швидких алгоритмів аналізу пакетів, оптимального використання ресурсів мережевого процесора важливим є дослідження властивостей вхідних потоків, що надходять у систему на обробку. Дослідження показують, що потоки в сучасних мережах не є найпростішими, мають істотну післядію й самоподібність. Необхідність забезпечення різних категорій мережевих додатків високою якістю обслуговування, урахування періодично виникаючих затримок у передачі даних і втрати пакетів при недостатній продуктивності й обмежених ресурсах пам'яті роблять дослідження властивостей самоподібного трафіка дійсно актуальними. У зв'язку з цим важливим питанням є визначення параметрів розподілу інформаційного потоку даних і дослідження властивостей трафіка [1].

Аналіз останніх досліджень

В області трафікового моделювання проведено багато досліджень [1-4], в яких інформаційні потоки аналізуються на наявність, ідентифікацію і якісну оцінку ряду характеристик. В [5] показано, що трафік практично всіх сучасних додатків має самоподібні властивості, а характерними законами розподілу є розподіли з «важкими хвостами».

Формулювання цілей статті (постановка завдання)

Мета роботи полягає в розробці імітаційної моделі мережі з властивостями самоподібності, дослідження трафіка при зміні форми розподілення потоку.

Основна частина

Технологія передачі інформації передбачає, що потік байтів розбивається на окремі пакети фіксованої довжини й інформація далі передається на пакетному рівні по дуплексному каналу взаємодії. На прийомному кінці дані знову збираються в потік байтів. Для надійної (безпомилкової) доставки пакетів необхідно затратити додатковий час на передачу пакетів підтвердження й повторення передачі загублених пакетів. На зазначеному відрізку часу процес передачі інформації блокується. Відзначимо також, що через нерегулярний вплив перерахованих факторів при передачі й розподілі інформації поведіння мережевого графіка здобуває випадковий характер, тобто трафік у мережі формується випадковим образом. Для наочної інтерпретації вищевказаних особливостей поведіння процесів у мережі й пошуку шляхів параметризації цих процесів найбільш придатним є моделювання мережевого трафіка режимом ON/OFF.

Найважливішим параметром, що характеризує ступінь самоподібності, є параметр Херста H , обумовлений для часового ряду $x(t_i)$, t_i – дискретні моменти часу, $1 \leq i \leq N$. Параметр Херста описується емпіричним співвідношенням

$$M \left[\frac{R}{S} \right] \sim a x(t)^H, \quad (1)$$

де a – константа, R – розмах відхилення значень ряду x , S – стандартне відхилення x .

Нехай середнє значення випадкової величини

$$\overline{x(N)} = \frac{1}{N} \sum_{i=1}^N x(t_i). \quad (2)$$

Стандартне відхилення x визначається з формули

$$S(N) = \sqrt{\frac{1}{N} \sum_{i=1}^N [x(t_i) - \overline{x(N)}]^2}. \quad (3)$$

Позначимо через

$$X(t, N) = \sum_{u=1}^t [x(u) - \overline{x(N)}] \quad (4)$$

накопичене відхилення значень випадкової величини $x(t)$ від її середнього значення за час t .

Різниця між максимальним і мінімальним значеннями $X(t, N)$ називається розмахом

$$R(N) = \max X(t, N) - \min X(t, N), \quad (5)$$

де $1 \leq t \leq N$.

Трафік практично всіх сучасних мереж має самоподібні властивості, а характерними законами розподілу є розподіли з «важкими хвостами», такі як, наприклад, розподіл Вейбулла й Парето. У роботі було проведене дослідження вплив параметра форми α розподілу Парето, що має функцію розподілу:

$$F(x) = 1 - \left(\frac{\beta}{x} \right)^\alpha \quad (6)$$

де α – параметр форми, β – мінімальне значення випадкової величини x .

Параметр α пов'язаний з показником Херста H виразом (7)[8]:

$$\alpha = 3 - 2H \quad (7)$$

При значеннях H в інтервалі від 0,5 до 1 трафік є самоподібним, а, отже, згідно (7) α повинне мати значення від 1 до 2. При цьому ступінь самоподібності стає тим більше, чим ближче параметр форми до одиниці. Як порівняння були обрані значення параметра форми розподілу Парето.

Різні типи потоків даних мають властивості самоподібності. Ethernet трафік є самоподібним зі значеннями параметра Херста H між 0.7 й 0.9 та розподілом Парето від 1.2 [6]. Трафік браузера WWW (відправлений й отриманий, обмірюваний як packets/s або bits/s) є також самоподібним, при цьому розподіл щільності може бути заданий за допомогою розподілу Парето між 1.16 й 1.5 [7].

Можна використовувати різні генератори трафіка при моделюванні передачі потоку даних, що розподілені за певним законом. У даній роботі мережевий трафік моделюється у системі OPNET окремими джерелами, які передають дані з однаковою інтенсивністю, але з тривалостями, що мають розподіл з «важким хвостом».

Тривалості ON і OFF періодів розглядаються як випадкові величини з розподілом Парето. Параметр Херста дорівнює 0.7 і середній час посилки пакетів 10.000 packets/sec.

FOST (Fractal Onset time Scale) - фрактальний масштаб часу настання використовується з параметром Херста для визначення параметра зсуву розподілу Парето.

Source activity - визначає відсоток часу, коли хоча б одне з незалежних ON-OFF джерел трафіка є активним.

Варіації цих параметрів наведені у таблиці 1.

Таблиця 1

FOTS	source activity	№ моделі
0,0001	50	1
	75	2
	90	3
0,001	50	4
	75	5
	90	6
0,01	50	7
	75	8
	90	9

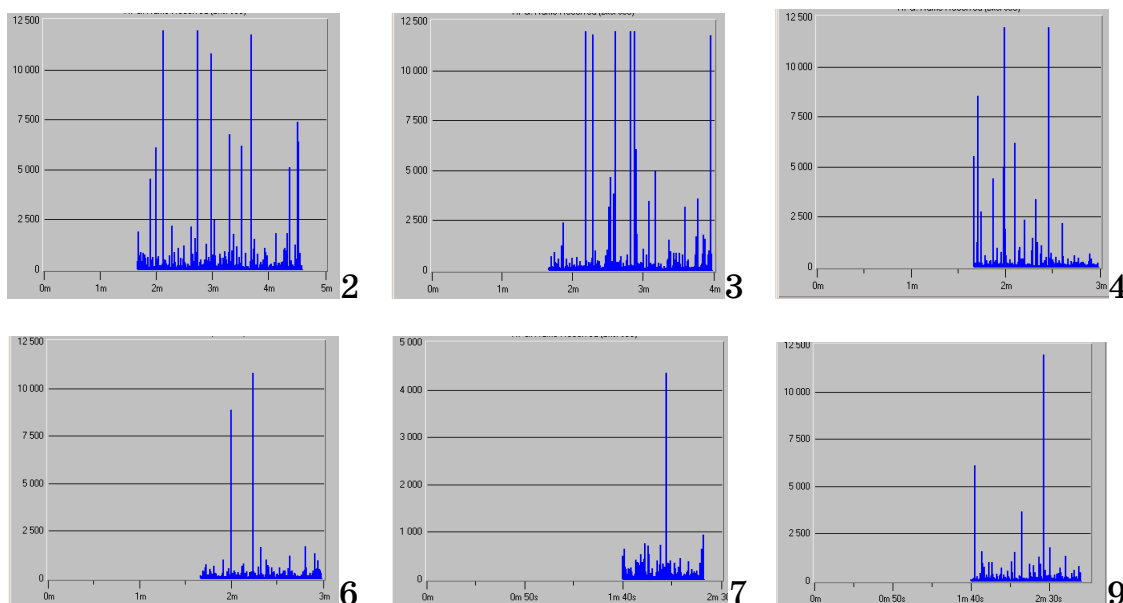


Рисунок 1 - Кількість відправлених пакетів для різних значень FOTS у відповідності за номером моделі

Розглянемо випадок, коли один сервер посилає повідомлення тільки одному клієнту (рис. 2). В іншому випадку використовуються два сервери, кожен з яких з'єднаний зі світчем, який посилає інформацію одному клієнту (рис. 3). Мережа містить вузли-генератори самоподібного трафіка, а також вузли приймача, які з'єднані між собою перемикачами. Імітаційне моделювання здійснено у системі створення прототипу комп'ютерної мережі OPNET.

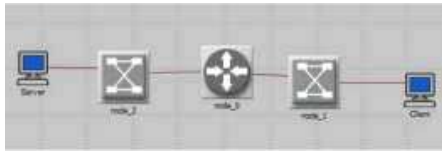


Рисунок 2 - Модель мережі 1

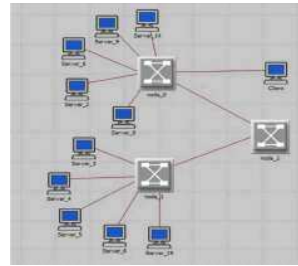


Рисунок 3 - Модель мережі 2

Досліджуємо ефект самоподібності, змінюючи параметр Парето (рис.4).

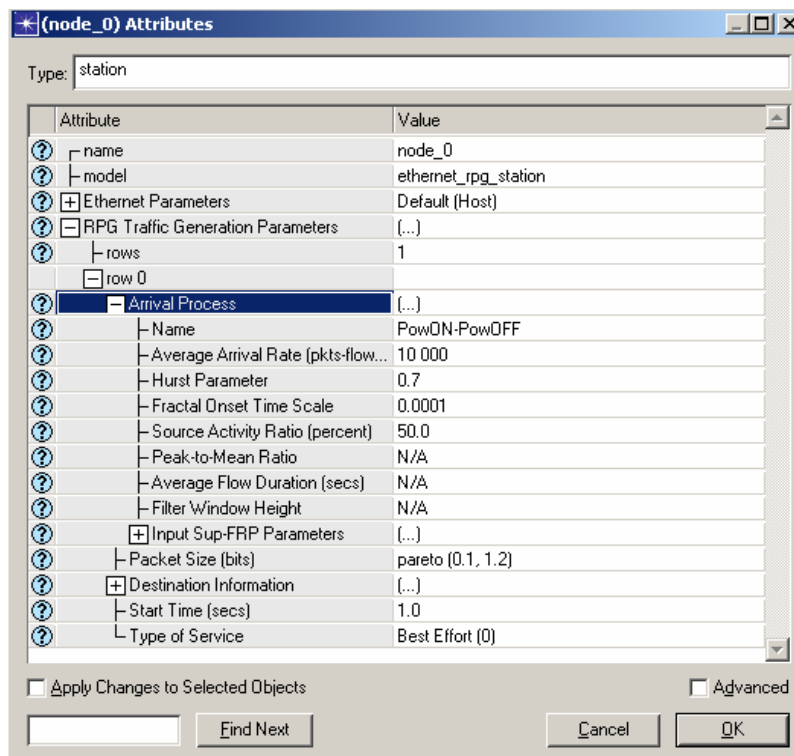


Рисунок 4 - Параметри джерела трафіка у OPNET

Таблиця 2

Моделювання мережі 1 та розрахунок оцінок

	Параметр періода ON/OFF	Aggregate оцінка	R/S оцінка	Periodogram оцінка	Whittle оцінка	Середнє значення Н	Теоретичне значення Н
1	Pareto(10,0.8)	0. 203	0. 496	0. 416	0. 514	0.41	
2	Pareto(10,1)	0. 558	0. 498	0. 454	0.500	0.50	
3	Pareto(10,1.2)	0. 555	0. 508	0. 425	0. 502	0.50	0.90
4	Pareto(10,1.4)	0. 601	0. 527	0. 476	0. 509	0.53	0.80
5	Pareto(10,1.6)	0. 573	0.55	0. 541	0. 538	0.55	0.70
6	Pareto(10,1.8)	0.64	0. 528	0. 805	0. 561	0.63	0.60

Моделювання мережі 2 та розрахунок оцінок

	Параметр періода ON/OFF	Aggregate оцінка	R/S оцінка	Periodogram оцінка	Whittle оцінка	Середнє
1	Pareto(10,0.8)	0. 459	0. 485	0. 315	0. 502	0.44
2	Pareto(10,1)	0. 397	0. 426	0. 786	0. 514	0.53
3	Pareto(10,1.2)	0. 458	0. 419	0.75	0. 537	0.54
4	Pareto(10,1.4)	0. 489	0. 278	0.77	0. 582	0.53
5	Pareto(10,1.6)	0. 613	0. 253	0.89	0.71	0.62
6	Pareto(10,1.8)	0. 627	0. 241	1. 069	0. 789	0.68

Відомо, що самоподібність залежить від характеристик періоду ON/OFF, а також від α (трафік володіє більш вираженими властивостями самоподібності при значеннях α більше 1). У проведених дослідженнях виявлена інша залежність: з більш високими α трафік є більш самоподібним. На іншому етапі моделювання було змінено кількість джерел від 5 до 40 у моделі мережі 2. Перевірено взаємозв'язок між самоподібністю трафіка й кількістю джерел. При збільшенні кількості ON/OFF станцій, H також збільшується. З аналізу проведених досліджень можна зробити висновок, що існує мінімальна залежність самоподібності від кількості ON / OFF джерел.

Висновки та перспективи подальших досліджень

За результатами проведених досліджень можна зробити висновок про те, що при найбільшому ступеню самоподібності переданого мережевого трафіка (при $H \rightarrow 0,7$), величина затримки мінімізується. Для того, щоб параметри трафіка, який передається, задовольняли необхідній якості обслуговування, варто враховувати не тільки розміри пакета й швидкість передачі, але також і характер розподілу. З одного боку, значення Херста самоподібного трафіку та його оцінки не стабільні й відрізняється від теорії. З іншого боку, збільшення кількості джерел не мають великого впливу на самоподібність трафіку.

ЛІТЕРАТУРА

1. Городецкий А.Я., Заборовский В.С. Информатика. Фрактальные процессы в компьютерных сетях. – С.-Пб.: СПбГТУ, 2000. – 102с.
2. Олійник В.Ф. Основи теорії зв'язку. – К.: Техніка, 2000. – 150с.
3. Королёв А.В., Кучук Г.А., Пашнев А.А. Адаптивная маршрутизация в корпоративных сетях. – Х.: ХВУ, 2003. – 224с.
4. В.В. Гнатушенко Дослідження управління трафіком GPRS-мережі за допомоги імітаційної моделі / В.В. Гнатушенко, Данладі Алі, О.І. Михальов // Системні технологій. Регіональний міжвузівський збірник наукових праць. – Випуск 2 (80). – Дніпропетровськ, 2012. – С. 67-78
5. Шелухин О.И., Тенякшев А.М., Осин А.В. Фрактальные процессы в телекоммуникациях. – М.: Радиотехника, 2003. – 480 с.
6. Leland W. (February 1994). On the selfsimilar nature of ethernet traffic (extended version). IEEE/ acm transactions on networking, Vol. 2, No 1. P. 1-15.
7. Norros, I. A storage model with self-similar input / I. Norros // Queueing System. – 1994. – V. 8.
8. Влияние параметра формы распределения Парето на временные характеристики трафика [Электронный ресурс] /Поздняк И.С., Буранова М.А.// – Режим доступа: <http://www.sworld.com.ua>

Р.В. Кирия, Т.Ф. Мищенко, Л.В. Камкина, Ю.В. Бабенко

**РАЗРАБОТКА БЫСТРОГО АЛГОРИТМА ОПРЕДЕЛЕНИЯ
ПРОПУСКНОЙ СПОСОБНОСТИ СИСТЕМЫ
«КОНВЕЙЕР – БУНКЕР – КОНВЕЙЕР»**

Аннотация. Разработана упрощенная марковская модель функционирования системы «конвейер – бункер – конвейер», позволяющая определить пропускную способность этой системы при различных соотношениях производительностей надбункерного и подбункерного конвейеров. Полученные результаты теоретических исследований достаточно хорошо совпали с результатами имитационного моделирования.

Ключевые слова: быстрый алгоритм, марковская модель, система, бункер, конвейер, пропускная способность, имитационное моделирование.

Вопросами исследования функционирования системы «конвейер – бункер – конвейер» занимались многие исследователи [1–3]. Полученные в этих работах результаты применимы для частных случаев режимов функционирования аккумулирующего бункера. Разработанную в работе [4] математическую модель функционирования системы «конвейер – бункер – конвейер» можно использовать в инженерных расчетах только в частном случае равенства поступающего в бункер и разгружаемого из бункера грузопотоков. В остальных случаях получены сложные зависимости, которые очень плохо поддаются инженерному анализу.

Поэтому разработка простых или «быстрых алгоритмов» определения пропускной способности системы «конвейер – бункер – конвейер» является актуальной задачей [5].

Предположим, что в системе «конвейер – бункер – конвейер» параметры потоков отказов и восстановлений надбункерного и подбункерного конвейеров равны λ_1 , μ_1 и λ_2 , μ_2 соответственно. Производительность надбункерного конвейера и питателя соответственно равны m_Q и Q_n . Объем бункера равен V (рис. 1).

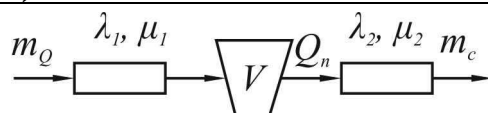


Рисунок 1 - Расчетная схема функционирования системы «конвейер – бункер – конвейер»

Рассмотрим сначала случай $m_Q > Q_n$.

Предположим, что подбункерный конвейер функционирует без остановок, т.е. $\lambda_2 = \mu_2 = 0$. Тогда граф состояния такой системы имеет вид (рис. 2).

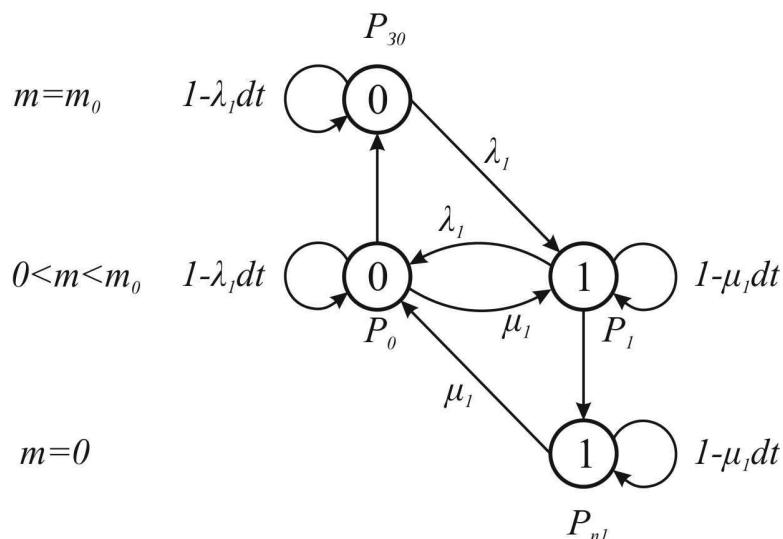


Рисунок 2 - Граф состояний системы «конвейер – бункер – конвейер» при $m_Q > Q_n$

На рис. 2 индекс «0» обозначает состояние системы, при котором надбункерный конвейер работает, индекс «1» обозначает состояние системы, при котором надбункерный конвейер не работает.

Тогда в этом случае обозначим через $P_0(m, t)$ и $P_1(m, t)$ соответственно вероятности нахождения системы в состояниях «0» и «1» при количестве груза в бункере, равном m , а через $P_{30}(t)$ – вероятность нахождения системы в состоянии «0», при этом бункер заполнен ($m = m_0$), и $P_{n1}(t)$ – вероятность нахождения системы в состоянии «1», при этом бункер пуст ($m = 0$).

Система уравнений Колмогорова [6], описывающая процесс функционирования бункеров, в этом случае примет вид:

$$\begin{cases} \frac{\partial P_0}{\partial t} + (m_Q - Q_n) \frac{\partial P_0}{\partial m} = -\lambda_1 P_0(m, t) + \mu_1 P_1(m, t), \\ \frac{\partial P_0}{\partial t} - Q_n \frac{\partial P_1}{\partial m} = \lambda_1 P_0(m, t) - \mu_1 P_1(m, t), \\ \frac{dP_{30}}{dt} = -\lambda_1 P_{30}(t) + (m_Q - Q_n) P_0(m_0, t), \\ \frac{dP_{n1}}{dt} = -\mu_1 P_{n1}(t) + Q_n P_1(0, t). \end{cases} \quad (1)$$

Кроме того, должны выполняться:

начальные условия:

$$\text{при } t = 0 \quad P_0(m, 0) = P_1(m, 0) = P_{30}(0) = 0, \quad P_{n1}(0) = 1; \quad (2)$$

граничные условия:

$$\text{при } m = m_0 \quad Q_n P_1(m_0, t) = \lambda_1 P_{30}(t); \quad (3)$$

$$\text{при } m = 0 \quad (m_Q - Q_n) P_0(0, t) = \mu_1 P_{n1}(t), \quad (4)$$

а также условие нормирования

$$P_{30}(t) + P_{n1}(t) + \int_0^{m_0} P_0(m, t) dm + \int_0^{m_0} P_1(m, t) dm = 1, \quad (5)$$

где m – текущее значение количества груза в бункере, т; m_0 – максимальное количество груза в бункере, т.

Для стационарного случая, т.е. при $t \rightarrow \infty$, система уравнений (1) с учетом граничных условий (2)–(4) примет вид:

$$\begin{cases} (m_Q - Q_n) \frac{dP_0}{dm} = -\lambda_1 P_0(m) + \mu_1 P_1(m), \\ -Q_n \frac{dP_1}{dm} = \lambda_1 P_0(m) - \mu_1 P_1(m), \\ -\lambda_1 P_{30} + (m_Q - Q_n) P_0(m_0) = 0, \\ -\mu_1 P_{n1} + Q_n P_1(0) = 0, \\ Q_n P_1(m_0) = \lambda_1 P_{30}, \\ (m_Q - Q_n) P_0(0, 0) = \mu_1 P_{n1}, \end{cases} \quad (6)$$

где $P_0(m)$, $P_1(m)$ – значения $P_0(m, t)$ и $P_1(m, t)$ при $t \rightarrow \infty$; P_{30} и P_{n1} – значения $P_{30}(t)$ и $P_{n1}(t)$ при $t \rightarrow \infty$.

Пропускную способность системы «конвейер – бункер – конвейер» в этом случае определится из выражения:

$$m_c = \left[P_{30} + \int_0^{m_0} P_0(m) dm + \int_0^{m_0} P_1(m) dm \right] Q_n. \quad (7)$$

Подставляя решение уравнений (6) в (7), получим пропускную способность системы «конвейер – бункер – конвейер» в виде:

$$m_c = \left[\frac{\frac{e^{A_1 m_0}}{\gamma_1} + \frac{\bar{m}_Q}{(\bar{m}_Q - Q_n)} (e^{A_1 m_0} - 1)}{1 + \frac{e^{A_1 m_0}}{\gamma_1} + \frac{\bar{m}_Q}{(\bar{m}_Q - Q_n)} (e^{A_1 m_0} - 1)} \right] Q_n, \quad (8)$$

где

$$A_1 = \frac{\mu_1 [m_Q - (1 + \gamma_1) Q_n]}{(m_Q - Q_n) Q_n}; \quad \bar{m}_Q = \frac{m_Q}{1 + \gamma_1}; \quad \gamma_1 = \frac{\lambda_1}{\mu_1}; \quad m_0 = \gamma V.$$

Здесь γ_1 – коэффициент аварийности надбункерного конвейера;
 γ – удельный вес груза.

Если подбункерный конвейер функционирует с простоями, т.е. $\lambda_2 \neq 0$, $\mu_2 \neq 0$, то, подставив в формулу (8) вместо Q_n его среднее значение, равное

$$\bar{Q}_n = \frac{Q_n}{1 + \gamma_2},$$

где $\gamma_2 = \frac{\lambda_2}{\mu_2}$ – коэффициент аварийности подбункерного конвейера,

получим пропускную способность системы «конвейер – бункер – конвейер» без дополнительных ограничений, т.е. при $m_Q > Q_n$:

$$m_c = \left[\frac{\frac{e^{A_1 \gamma V}}{\gamma_1} + \frac{\bar{m}_Q}{(\bar{m}_Q - \bar{Q}_n)} (e^{A_1 \gamma V} - 1)}{1 + \frac{e^{A_1 \gamma V}}{\gamma_1} + \frac{\bar{m}_Q}{(\bar{m}_Q - \bar{Q}_n)} (e^{A_1 \gamma V} - 1)} \right] \bar{Q}_n, \quad (9)$$

где

$$A_1 = \frac{\mu_1 [m_Q - (1 + \gamma_1) \bar{Q}_n]}{(m_Q - \bar{Q}_n) \bar{Q}_n}.$$

Рассмотрим случай $m_Q < Q_n$.

Предположим, что в системе «конвейер – бункер – конвейер» надбункерный конвейер функционирует без остановок, т.е. $\lambda_1 = \mu_1 = 0$. Тогда граф состояний этой системы имеет, представленный на рис. 3.

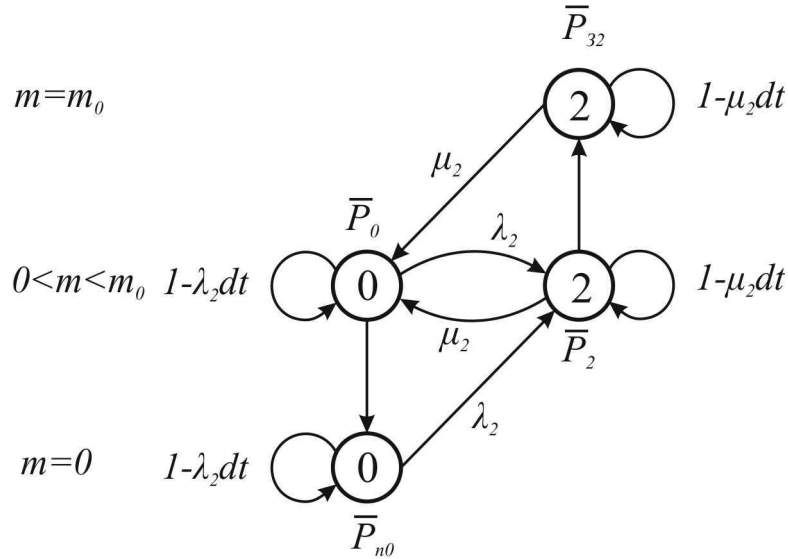


Рисунок 3 - Граф состояний системы
«конвейер – бункер – конвейер» при $m_Q \leq Q_n$

На рис. 3 индекс «0» обозначает состояние системы, при котором подбункерный конвейер работает, индекс «2» обозначает состояние системы, при котором подбункерный конвейер не работает.

Тогда в этом случае обозначим через $\bar{P}_0(m, t)$ и $\bar{P}_2(m, t)$ соответственно вероятности нахождения системы в состояниях «0» и «1» при условии, что количестве груза в бункере равно m , а через $\bar{P}_{32}(t)$ вероятность нахождения системы в состоянии «0», при этом бункер заполнен ($m = m_0$), и $\bar{P}_{n0}(t)$ – вероятность нахождения системы в состоянии «2», при этом бункер пуст ($m = 0$).

Система уравнений Колмогорова, описывающая процесс функционирования бункеров, в этом случае примет вид:

$$\begin{cases} \frac{\partial \bar{P}_0}{\partial t} + (m_Q - Q_n) \frac{\partial \bar{P}_0}{\partial m} = -\lambda_2 \bar{P}_0(m, t) + \mu_2 \bar{P}_2(m, t), \\ \frac{\partial \bar{P}_2}{\partial t} + m_Q \frac{\partial \bar{P}_2}{\partial m} = \lambda_2 \bar{P}_0(m, t) - \mu_2 \bar{P}_2(m, t), \\ \frac{d \bar{P}_{32}}{dt} = -\mu_2 \bar{P}_{32}(t) + m_Q \bar{P}_2(m_0, t), \\ \frac{d \bar{P}_{n0}}{dt} = -\lambda_2 \bar{P}_{n0}(t) + (m_Q - Q_n) \bar{P}_2(0, t). \end{cases} \quad (10)$$

Кроме того, должны выполняться:

начальные условия:

$$\text{при } t = 0 \quad \bar{P}_0(m, 0) = \bar{P}_2(m, 0) = \bar{P}_{32}(0) = 0, \quad \bar{P}_{n0} = 1; \quad (11)$$

граничные условия:

$$\text{при } m = m_0 \quad (Q_n - m_Q)\bar{P}_0(m_0, t) = \mu_2 \bar{P}_{32}(t); \quad (12)$$

$$\text{при } m = 0 \quad \bar{P}_2(0, t) = \lambda_2 \bar{P}_{n0}(t), \quad (13)$$

а также условие нормирования

$$\bar{P}_{32} + \bar{P}_{n0} + \int_0^{m_0} \bar{P}_0(m, t) dm + \int_0^{m_0} \bar{P}_2(m, t) dm = 1. \quad (14)$$

Для стационарного случая, т.е. при $t \rightarrow \infty$, система уравнений (10) с учетом условий (11)-(13) примет вид:

$$\begin{cases} (m_Q - Q_n) \frac{d\bar{P}_0}{dm} = -\lambda_2 \bar{P}_0(m) + \mu_2 \bar{P}_2(m), \\ m_Q \frac{d\bar{P}_2}{dm} = \lambda_2 \bar{P}_0(m) - \mu_2 \bar{P}_2(m), \\ -\mu_2 \bar{P}_{32} + m_Q \bar{P}_2(m_0) = 0, \\ -\lambda_2 \bar{P}_{n0} + (Q_n - m_Q) \bar{P}_0(0) = 0, \\ (Q_n - m_Q) \frac{d\bar{P}_0}{dm} = \mu_2 \bar{P}_{32}, \\ m_Q \frac{d\bar{P}_2}{dm} = \lambda_2 \bar{P}_{n0}, \end{cases} \quad (15)$$

где $\bar{P}_0(m)$, $\bar{P}_2(m)$ – значения $\bar{P}_0(m, t)$ и $\bar{P}_2(m, t)$ при $t \rightarrow \infty$; $\bar{P}_{30}$ и $\bar{P}_{n1}$ – значения $\bar{P}_{32}(t)$ и $\bar{P}_{n0}(t)$ при $t \rightarrow \infty$.

Пропускная способность в этом случае определяется из выражения:

$$m_c = m_Q \bar{P}_{n0} + Q_n \int_0^{m_0} \bar{P}_0(m) dm. \quad (16)$$

Подставляя решение уравнений (15) в (16), получим пропускную способность в виде:

$$m_c = \left[\frac{1 + \frac{(Q_n - \bar{Q}_n)}{(\bar{Q}_n - m_Q)} (1 - e^{A_2 \gamma V})}{1 + \gamma_2 e^{A_2 \gamma V} + \frac{(Q_n - \bar{Q}_n)}{(\bar{Q}_n - m_Q)} (1 - e^{A_2 \gamma V})} \right] m_Q, \quad (17)$$

где

$$A_2 = \frac{\mu_2 [(1 + \gamma_2) m_Q - Q_n]}{m_Q (Q_n - m_Q)}.$$

Если надбункерный конвейер функционирует с простоями, т.е. $\lambda_1 \neq 0$, $\mu_1 \neq 0$, то подставив в последнюю формулу вместо m_Q его сред-

нее значение, равное $\bar{m}_Q = \frac{m_Q}{1 + \gamma_1}$, получим пропускную способность

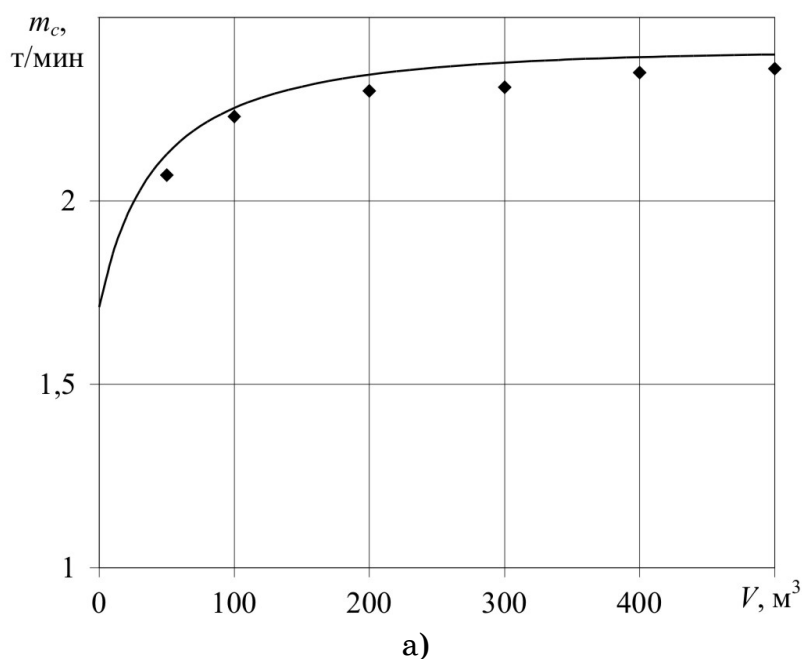
системы «конвейер – бункер – конвейер» в случае $m_Q \leq Q_n$ без дополнительных ограничений в виде:

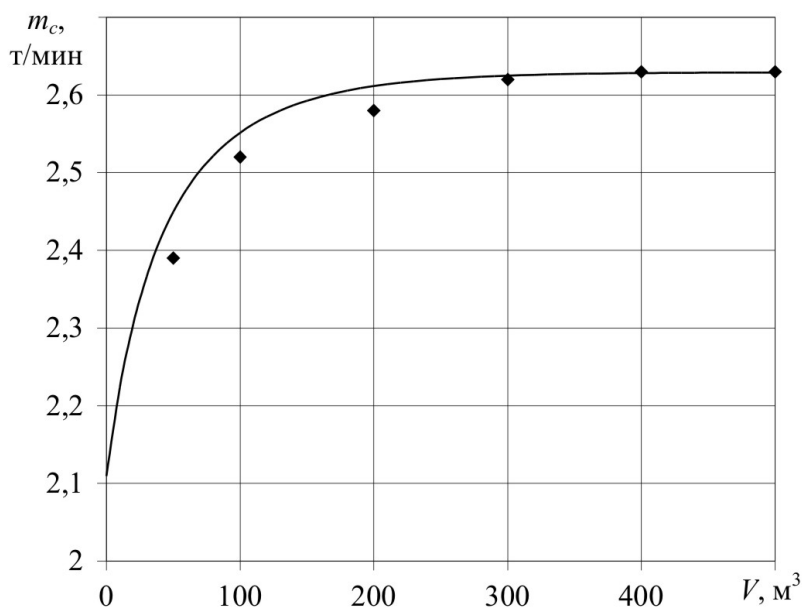
$$m_c = \left[\frac{1 + \frac{(Q_n - \bar{Q}_n)(1 - e^{A_2 \gamma V})}{(\bar{Q}_n - \bar{m}_Q)}}{1 + \gamma_2 e^{A_2 \gamma V} + \frac{(Q_n - \bar{Q}_n)(1 - e^{A_2 \gamma V})}{(\bar{Q}_n - \bar{m}_Q)}} \right] \bar{m}_Q, \quad (18)$$

где

$$A_2 = \frac{\mu_2 [\bar{m}_Q (1 + \gamma_2) - Q_n]}{\bar{m}_Q (Q_n - \bar{m}_Q)}.$$

На рис. 4 показаны графики зависимости средней производительности m_c системы «конвейер – бункер – конвейер» от объема бункера V при различных значениях поступающего m_Q и разгружаемого Q_n грузопотоков, построенные согласно формул (9) и (18).





б)

а) – $m_Q > Q_n$; б) $m_Q < Q_n$

Рисунок 4 - Графики зависимости средней производительности m_c системы «конвейер – бункер – конвейер» от объема бункера V

При этом, для всех случаев средняя производительность поступающего грузопотока $m_Q = 3,7$ т/мин, а параметры потоков отказов и восстановлений надбункерного и подбункерного конвейеров принимали значения соответственно $\lambda_1 = 0,025$ мин⁻¹, $\mu_1 = 0,0614$ мин⁻¹; $\lambda_2 = 0,017$ мин⁻¹, $\mu_2 = 0,069$ мин⁻¹; удельный вес груза $\gamma = 1$ т/м³.

На рис. 4,а сплошная кривая соответствует формуле (9) ($m_Q > Q_n$) при $Q_n = 1$ т/мин.

На рис. 4,б сплошная кривая соответствует формуле (18) при ($m_Q \leq Q_n$) $Q_n = 4$ т/мин.

Кроме того, на рис. 4 точками показаны результаты имитационного моделирования.

Из рис. 4,а,б видно, что при любых соотношениях значений m_Q и Q_n с увеличением объема бункера V пропускная способность m_c системы «конвейер – бункер – конвейер» сначала увеличивается, а затем с увеличением объема бункера V пропускная способность m_c не изменяется и принимает постоянное значение.

Выводы. На основе теории марковских процессов с непрерывным временем и дискретным состоянием получены упрощенные модели, описывающие процесс функционирования системы «конвейер – бункер – конвейер».

Анализ процесса функционирования системы «конвейер – бункер – конвейер» показал, что при любых соотношениях производительностей надбункерного и подбункерного конвейеров с увеличением объема бункера пропускная способность системы сначала увеличивается, а затем принимает постоянное значение.

Полученные теоретические результаты достаточно хорошо совпали с результатами имитационного моделирования.

ЛИТЕРАТУРА

1. Владзиевский Д. П. Автоматические линии в машиностроении. Книга 1. / Д. П. Владзиевский. – М.: Машгиз, 1958. – 430 с.
2. Шахмейстер, Л. Г. Расчет осредняющей емкости у лавы методами теории массового обслуживания / Л. Г. Шахмейстер, П. И. Ярошевский // Уголь Украины. – 1967. – №8. – С. 66–68.
- Алотин, Л. М. Исследование и обоснование увеличения угледобычи при использовании аккумулирующего бункера в транспортной линии / Л. М. Алотин, В. Л. Белгородский // Физико-технические проблемы разработки полезных ископаемых. – 1970. – № 6. – С. 108–111.
3. Черкесов, Г. Н. Надежность технических систем с временной избыточностью / Г. Н. Черкесов. – М.: Советское радио, 1974. – 296 с.
4. Моисеев Н. Н. Математические задачи системного анализа / Н. Н. Моисеев. – М.: Наука, 1981. – 488 с.
5. Вентцель, Е.С. Теория случайных процессов и ее инженерные приложения: учебное пособие / Е.С. Вентцель, Л.А. Овчаров. – М.: КНОРУС, 2011. – 448 с.

E.A. Gavrilenko, V.V. Gnatushenko, Yu.V. Kholodnyak

**DETERMINE OF GEOMETRICAL CHARACTERISTICS
RANGES OF THE DISCRETELY PRESENTED CURVE WITH
LAW OF CURVATURE CHANGE**

Summary. The conditions modeling of curve with a monotonous change of curvature and the scheme of the destination of the provisions tangents at which the problem of modeling the contours with a monotonous change of curvature has a solution is considered in this article.

Keywords: discretely represented curve (DRC), the tangent, normal, basic triangle, radius of curvature, monotonous curvature change.

Problem statement

The main functional characteristic of dynamic surfaces is regulated nature of flow of the medium. At designing dynamic surfaces are important lines which connect surfaces geometry with the physical side of its functional purpose. Most often it is necessary to ensure the second order of the smoothness and regular change of curvature along the contours of which are elements of the surfaces carcass. In this case it is advisable to modeling the curve locally in areas which are limited to the initial points. The resulting areas of monotonous curves connecting together with the order of smoothness are not lower than the second. In order to ensure a given nature of the change curvature along the contour it is necessary to assign position of the tangent in the initial points at which the problem of modeling the curve with a monotonous curvature change has a solution.

Analysis of the previous researches

The method of determining the possibility of modeling of the curve with monotonous change of curvature based on a given points set is developed in the work [1]. Necessary condition is the location of the initial points such that the radiuses of the circles which define three successive points, change monotonically along the row.

Methods for formation of curves segment along which the curvature changes monotonically proposed in the works [2, 5]. The conditions

of modeling segment of curve are regular change of curvature, given positions of tangents and radiuses of curvature in the points which limit the segment. The method of determining of possible locations ranges of the thickened point on the segment of the monotonous curve is developed in the article [2]. Previously determined the position of the normal and the center of curvature which correspond to the point of thickening.

The method of determining of thickening point of the monotonous DRC on the given positions of tangents and radiuses of curvature at the initial points is proposed in the article [5]. At formation of the curves segment the point of thickening and the tangent, which passes through it, are defined within the basic triangle. The basic triangle limited to tangents that pass through two consecutive points and the chord that connects these points. The method, which made possible using parameters of the basic triangles for identify the ranges of curvature values of the monotonous curve, is developed in the work [3]. In order to using these methods to modeling a contour of the points it is necessary to assign such characteristics in the initial points (positions of normals, tangents, positions of the centers of curvature, values of the radiuses of curvature), in which the task of forming of segments curves with a monotonous change of curvature and connecting of these segments with the second order smoothness has a solution.

The formulation of purpose of the article (task statement)

The purpose of the article is to develop a method for determining of the range positions tangents and values of curvature in the initial points of the monotonous DRC and determining of location area of contours that meets the above specifications.

Basic part

Suppose that monotonous curve is modeled by the method of thickening. Let consider the segment $(i...i+1)$. In the initial points are given positions of tangents t_i , t_{i+1} and normals n_i , n_{i+1} respectively (ill. 1).

Determine the conditions on the location of t_i , t_{i+1} and n_i , n_{i+1} , when it is possible to provide a monotonous increase of the radiuses of curvature at the segment $(i...i+1)$.

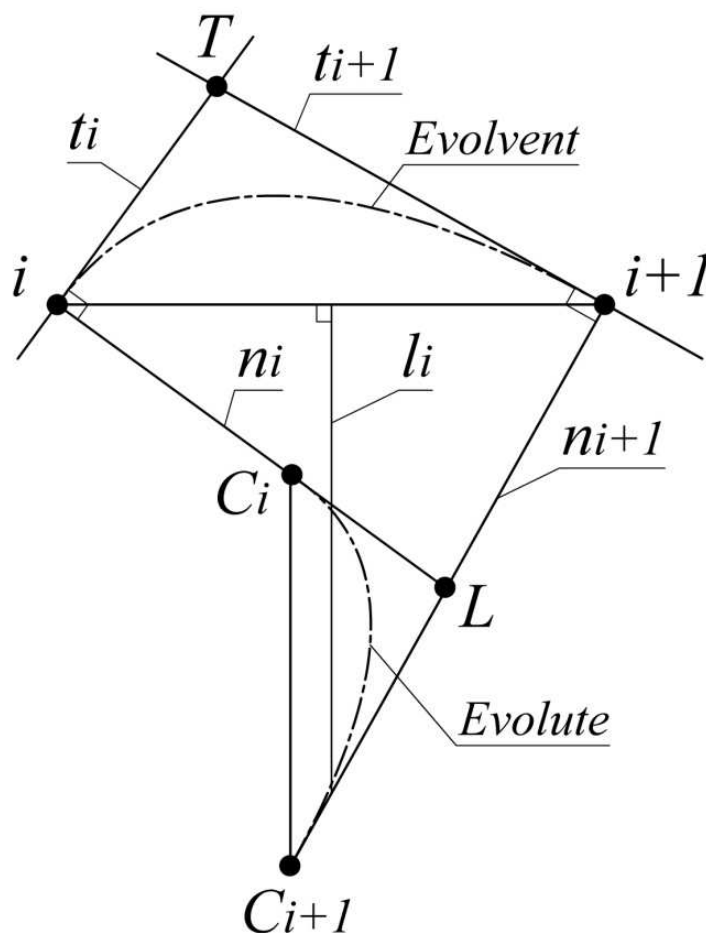


Illustration 1

Let denote the centers of curvature in the points i and $i+1$ as C_i and C_{i+1} respectively, and the point of intersection of normals as L . The triangle $C_i; L; C_{i+1}$ will be called a triangle of the centers of curvature [2].

The radiuses of curvature of the DRC in the points i and $i+1$ are respectively:

$$R_i = |C_i; i|, \quad R_{i+1} = |C_{i+1}; i+1|.$$

If along a curve on a segment $(i...i+1)$ radiuses of curvature are monotonically increasing, the evolute of the curve $(C_i...C_{i+1})$ is convex curve which is tangent to n_i and n_{i+1} in the points C_i and C_{i+1} respectively [4].

At the same time the length of the evolute (ΔR) is equal to the difference between the radius of the curvature at the initial points:

$$\Delta R = R_{i+1} - R_i.$$

Thus evolutes located inside the triangle $C_i; L; C_{i+1}$ and the value of its length satisfies the following conditions:

$$\Delta R > |C_i; C_{i+1}|; \quad (1)$$

$$\Delta R < |C_i; L| + |L; C_{i+1}|, \quad (2)$$

$|C_i; C_{i+1}|$, $|C_i; L|$, $|L; C_{i+1}|$ – are sides lengths of the triangle of the centers of curvature $C_i; L; C_{i+1}$.

The condition (2) can be written as

$$|C_{i+1}; i+1| - |C_i; i| < |C_i; L| + |L; C_{i+1}|, \\ \text{or } |i; L| > |L; i+1|. \quad (3)$$

With such an arrangement of normals the tangents t_i , t_{i+1} and chord $[i; i+1]$ determines the basic triangle $i; T; i+1$ with an aspect ratio that satisfies the condition:

$$|i; T| < |T; i+1|. \quad (4)$$

Location of normals and tangents in which the following requirements (3) and (4), respectively, are essential for the formation of a curve with a monotonous increase of curvature on the segment $(i \dots i+1)$. Conditions (3) and (4) may be used as a criterion for determining the position of the tangents (normals) in the initial points according to the conditions of the problem.

In modeling of the DRC along the points positions of the tangents in initial points must be assigned in such a way that the condition (4) holds for all segments of the curve. Let's define the range of positions of the tangent t_i .

To do this the analysis phase of the initial points every three consecutive points is held circles. Obtained circles is called as contiguous circles. Suppose that through the point i are passing circles ΠK_{i-1} , ΠK_i and ΠK_{i+1} . In order to modeling the DRC with a monotonous change of curvature, the range of possible locations of the tangent is the angle that is limited by the tangents of the respective contiguous circles. One boundary is tangent to the circle ΠK_i ($t_{\Pi K_i}$) and the second - a tangent that is closer to $t_{\Pi K_i}$: tangent to circle ΠK_{i-1} or to circle ΠK_{i+1} [1].

After determining the ranges of the tangents positions at all points the minimal of them is chosen. Position of the tangent to the DRC at a point which corresponds to the minimum range (for example,

t_i) is assigned to the center of the range. Positions of the tangents in the previous and next point (t_{i-1} , t_{i+1}) is determined by taking into account the position of the tangent t_i .

The range of position of tangent t_{i+1} is limited by the tangent to the circle ΠK_{i+1} and line $(A; i+1)$. Line $(A; i+1)$ defined by passes through the point A which belongs to the tangent t_i so that the triangle $i; A; i+1$ is isosceles (ill. 2). Ranges of positions of tangents from t_{i-1} to t_i and from t_{i+1} to t_n are defining similarly.

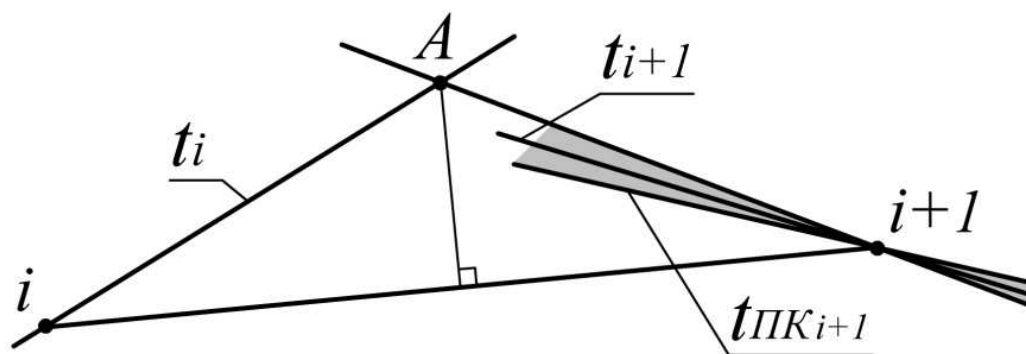


Illustration 2

In the result is a sequence of basic triangles, which allows modeling of contour with a monotonous change of curvature along the all points.

Basic triangles define ranges of the radiuses of curvature in the initial points, which can be provided for modeling a monotonous curve [3]. In order to be able to modeling the DRC with monotonous change of curvature along the all point, it is necessary that the ranges of curvature which determined by previous and next basic triangle were intersected. If these ranges do not intersect, it is necessary to make an adjustment of the positions of tangents in the initial points.

Adjusting of the tangent is performed using its rotation around the corresponding point of initial DRC. The range of radiuses of curvature at the respective point increases when turned clockwise. A boundary is a position of the tangent at which one of the basic triangles, the sides of which determines the tangent is isosceles.

Necessary condition for ensuring crossing the ranges of the radiuses of curvature at the point is the location of the initial points according to the conditions set out in [1].

Conclusions and prospects of further researches

The following results were obtained in this work:

- the condition of the location of the tangents (normals) at the points that limit the segment of monotonous curve is defined. The criterion is ratio of sides of basic triangles or the location of the point of intersection of normals in initial points;

- the method of assigning of the positions of tangents (normals) at the points of curve in which the problem of modeling a contour with a monotonous change of curvature has a solution along the all points is suggested in this article.

The results obtained allow to model DRC along the points set which is composed of any number of points.

LITERATURE

1. Гавриленко Е.А. Дискретное интерполирование плоских одномерных обводов с закономерным изменением кривизны: дис. канд. техн. наук / Е.А. Гавриленко – Мелитополь, 2004. – 182 с.
2. Гавриленко Є.А. Визначення положення точки згущення при моделюванні монотонної дискретно представленої кривої / Є.А. Гавриленко // Прикл геом. та інж. графіка / Праці ТДАТУ – Вип.4, Т.53.–Мелітополь 2012.– с. 29-33.
3. Гавриленко Є.А. Визначення діапазонів геометричних характеристик монотонної дискретно представленої кривої / Є.А. Гавриленко, Ю.В. Холодняк // Прикл. геом. та інж. графіка / Праці ТДАТУ – Вип.4, Т.54. – Мелітополь 2012. – с. 38-42.
4. Рашевский П.К. Курс дифференциальной геометрии / П.К. Рашевский –М., ГИТТЛ, 1956.– 420 с.
5. Холодняк Ю.В. Формування ділянки дискретно представленої кривої із монотонною зміною кривини / Ю.В. Холодняк // Прикл. геом. та інж. графіка / Праці ТДАТУ – Вип.4, Т.57. – Мелітополь, 2013. – с. 211-216

СОДЕРЖАНИЕ

Пономарев И.В., Рыбка Ю.М. Автоматизированное размещение данных на информационных сайтах	3
Герасимов В.В., Пономарьев И.В., Щетинин И.О. Оптимізація процесу розробки інтернет-додатків на платформі Java..	7
Герасимов В.В., Пеньков А.П., Пушков Г.О. Аналіз технологій розробки насичених інтернет-додатків на платформі Java	15
Григорьев В.М. Выбор платформы для реализации лабораторного практикума по компьютерным сетям.....	22
Ивон А.И., Истускин В.Ф. Производительность центрального процессора в режиме троттлинга	31
Матвеева Н.А. Нейронные сети в задачах классификации дефектов	37
Атаманенко І.І., Сівцов Д.П. Використання модулів на асемблері для збільшення продуктивності C / C + + програм симетричного шифрування.....	43
Крохин В.В., Митиков Н.Ю. Об исследовании качества проектирования цифровых фильтров	52
Твердоступ Н.И. Об эффективности алгоритма адаптивного сглаживания в условиях влияния мешающих факторов	58
Волковский О.С., Петренко Я.Ю. Конвертация медиа файлов на примере аудио, в операционной системе Windows 8 ОС использованием Windows Store библиотеки. Сравнение с ffmpeg.....	66
Литвинов А.А., Павленко М.В., Матвеев А.А. Технология анализа работы клиники хирургического профиля	74
Литвинов А.А., Вякилов А.С., Брежнев А.И. Особенности построения медицинской информационной системы ориентированной на стандарты.....	82
Мартинovich Л.Я., Стеценко О.О. Проектування автоматизованої системи складання розкладу занять у коледжі	89
Алексеев М.А., Ахметшин К.А. Синтез обобщенных фазовых портретов многоканальных нелинейных динамических систем в сингулярном базисе вложенных векторов ...	97

Ахметшина Л.Г., Ахметшин А.М., Удовик И.М. Метод интерполяции неравномерных пространственных данных на основе сети Кохонена.....	106
Ясев А.Г. Система математического моделирования механического разрушения неметаллических покрытий электрических проводов	113
Малкіна В.М., Бездітний А.О. Використання Mechanical APDL для розрахунку власних коливань лопаток турбін у системі кінцево-елементного аналізу Ansys Workbench.....	124
Спірінцев В.В., Пихтеева І.В., Дмитрієв Ю.О. Розробка функціональної схеми процесу автоматизованого проектуювання.....	128
Кавац А.А. Моделирование процесса формирования микроструктуры металлических сплавов.....	135
Гнатушенко В.В., Проїдак Ю.С., Данладі Алі Імітаційне моделювання самоподібного трафіка при зміні форми Парето-розподілення інформаційного потоку.....	140
Кирия Р.В., Мищенко Т.Ф., Камкина Л.В., Бабенко Ю.В. Разработка быстрого алгоритма определения пропускной способности системы «конвейер – бункер – конвейер»	147
Gavrilenko E.A., Gnatushenko V.V., Kholodnyak YU.V. Determine of geometrical characteristics ranges of the discretely presented curve with law of curvature change	156

РЕФЕРАТЫ

УДК 621.3

Пономарьов І.В., Рибка Ю.М. **Мовні конструкції для паралельних асинхронних обчислень на реконфігурованих багатопроцесорних системах** // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1(84). – Дніпропетровськ, 2013, - С.3 – 6.

Запропоновано мовні конструкції розділу управління мови програмування паралельних обчислень на реконфігурованих БОС. Налаштування параметрів конфігурації системи при описі завдання дозволяють ефективно виконати обробку системою певної архітектури.

Бібл. 2.

УДК 004.4'23

Герасимов В.В., Пономарев И.В., Щетинин И.О. **Оптимизация процесса разработки Интернет-приложений на платформе Java** // Системные технологии. Региональный межвузовский сборник научных работ. – Випуск 1 (84). – Днепропетровск, 2013. – С. 7 – 14.

Рассмотрен Play Framework, который значительно упрощает и ускоряет процесс разработки Интернет-приложений на платформе Java. Описаны особенности применения и преимущества этой технологии. Представлены результаты сравнения Play Framework с альтернативными технологиями.

Библ. 6, илл. 1, табл. 1.

УДК 004.55

Герасимов В.В., Пеньков А.П., Пушков Г.О. **Аналіз технологій розробки насичених Інтернет-додатків на платформі Java** // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1 (84) – Дніпропетровськ, 2013. – С.15 – 21.

Проведено аналіз технології JavaFX 2, яка активно розвивається в останній час. Наведено особливості цієї технології, розглянуто перспективи її розвитку. Представлені результати порівняння сучасних технологій розробки насичених Інтернет-додатків на різних платформах.

Бібл. 5, іл. 2, табл. 1.

УДК 004.7:004.45

Григор'єв В.М. **Вибір платформи для реалізації віртуальної лабораторії по комп'ютерних мережах** // Системні технології. Регіональний міжвузівський збірник наукових праць. - Випуск 1 (84). - Дніпропетровськ, 2013. - С.22 – 30.

Досліджується час завантаження великого числа віртуальних машин Qemu під управлінням операційної системи RouterOS фірми Mikrotik в рамках хост-машини Ubuntu. Показано, що операційна система RouterOS є адекватним вибором для організації віртуальної лабораторії по комп'ютерних мережах.

Бібл. 1.

УДК 004.318+536.37

Івон О.І., Істушкін В.Ф. **Продуктивність центрального процесора в режимі троттлінгу** // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1(84) – Дніпропетровськ, 2013. – С. 31 – 36.

Виконано аналіз продуктивності процесора в режимі троттлінгу при його апаратній реалізації, за допомогою критичних терморезисторів на основі V02. Показано, що коли середня затрата енергії на одну машинну команду E_{in} не перевищує мінімальну енергію $E_{in\ min}$, що викликає режим троттлінгу, процесор працює з продуктивністю, яка відповідає нормальному тепловому режиму. При $E_{in} \geq E_{in\ min}$ продуктивність процесора зменшується обернено пропорційно значенню E_{in} .

Бібл. 4, іл. 2.

УДК621.3

Матвєєва Н.О. **Нейроні мережі в задачах класифікації дефектів**. // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1(84). – Дніпропетровськ, 2013. – С. 37 – 42.

Розглянуто можливість застосування нейроних мереж Кохонена для класифікації сигналів дефектоскопії. Навчена нейрона мережа промодельована на перевірочній множині зашумованих сигналів.

Бібл. 4, іл. 3.

УДК 004.422.833

Атаманенко И.И., Сивцов Д.П. **Использование модулей на ассемблере для увеличения производительности С/С++ программ симметричного шифрования** // Системные технологии. Региональный межвузовский сборник научных работ. – Випуск 1(84). – Днепропетровск, 2013. – С. 43 – 52.

В статье обсуждается возможность увеличения производительности С/С++ программ для шифрования информации на основе симметричных криптографических алгоритмов. Было установлено, что существенный эффект может быть достигнут путем реализации частей таких программ на ассемблере.

Библ. 7, рис. 3.

УДК 621.391.26(075)

Крохін В.В., Мітіков М.Ю. **Про дослідження якості проектування цифрових фільтрів** // Системные технологии. Региональный межвузовский сборник научных работ. – Випуск 1(84). – Днепропетровск, 2013. – С. 53 – 58.

З метою порівняльного аналізу якості проєктованих цифрових фільтрів разом з іншими характеристиками пропонується використовувати ширину перехідної зони між полосами пропускання та затримання. Наведено методику розрахунку такої характеристики, а також результати аналізу якості проєктованих цифрових фільтрів з використанням 6 різних вагових функцій.

УДК 519.281:620.179(035)

Твердоступ М.І. **Про ефективність алгоритму адаптивного згладжування в умовах впливу заважаючих факторів** // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1(84). – Дніпропетровськ, 2013. – С. 59 – 66.

Досліджено вплив експоненціального згладжування на похибку результату вимірювання індуктивності параметричного датчика, розташованого на поверхні залізорудного штуфа. Встановлено, що згладжування майже на порядок зменшує похибку результату вимірювання при наявності впливу неоднорідності магнітного поля датчика, що заважає.

Бібл. 5, рис. 3

УДК 624.04

Волковський О.С., Петренко Я.Ю. **Конвертація медіа файлів на прикладі аудіо, в операційній системі Windows 8 з використанням Windows Store бібліотеки. Порівняння з ffmpeg** // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1(84). – Дніпропетровськ, 2013. – С. 67 – 74.

Досліджено конвертація аудіо файлів з використанням Windows Store бібліотек, проведено порівняння з ffmpeg бібліотекою. Написана тестова програма, яка може бути використана для створення додатків по обробці медіа файлів. Проведено порівняння Windows Store бібліотеки та ffmpeg бібліотеки при конвертації аудіо.

Бібл. 3, іл. 7, табл. 3.

УДК 614.2+574/578+004.38

Литвинов О.А., Павленко М.В., Матвеев О.А. **Технологія інформаційного аналізу роботи відділення хірургічного профілю** // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1(84). – Дніпропетровськ, 2013. – С. 75 – 82.

В роботі розглянуто стратегія та технологія аналізу роботи клініки хірургічного профілю, яка спрямована на підтримку вирішення задач ефективного планування та розподілення заходів та ресурсів клініки, а також оцінку застосованих стандартів лікування. Розглянуті основні проблеми, які пов'язані з вирішенням задач. Запропонована стратегія аналізу, яка базується на застосуванні спеціального середовища, та розглянуті результати її застосування.

Бібл. 5. рис. 5., табл.2.

УДК 621.3.083.92

Мартинovich Л.Я., Стеценко О.О. **Проектування автоматизованої системи складання розкладу занять у коледжі** // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1 (84) – Дніпропетровськ, 2013. – С. 90 – 97.

Розроблено автономне програмне забезпечення для складання розкладу занять у коледжі, з урахуванням обмежень на кількість щоденного та тижневого навантаження для студентів, неприпустимості пропусків між заняттями для студентів та обмежень тижневого навантаження для викладачів на основі навчального та робочого планів.

Бібл.6, рис.3.

УДК 616.12

Алексеев М.О., Ахметшин К.О. **Синтез узагальнених фазових портретів багатоканальних нелінійних динамічних систем у базисах сингулярного розкладання вкладених векторів** // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1(84) – Дніпропетровськ, 2013. – С. 98 – 106.

Розглянуто інформаційні можливості нового методу аналізу багатовимірних нелінійних динамічних систем. Представлені результати експериментальної перевірки методу.

Бібл. 7, рис. 9.

УДК 616.12

Ахметшина Л.Г., Ахметшин О.М., Удовик І.М. **Метод інтерполяції нерівномірних просторових даних на основі мережі Кохонена** // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1(84) – Дніпропетровськ, 2013. – С. 107 – 113.

Розглянуто інформаційні можливості методу інтерполяції нерівномірних просторових даних на основі мережі Кохонена. Представлено результати моделювання геофізичних поверхонь на модельних та реальних даних.

Бібл.3, рис. 4.

УДК 669.02/09:519.28

Ясев А.Г. **Система математического моделирования механического разрушения неметаллических покрытий электрических проводов** // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1(84) – Дніпропетровськ, 2013. – С. 114 – 124.

Описано использование системы математического моделирования (которая включает математическое моделирование, разработку конструктивно-технологических предложений, оптимизацию параметров) при исследовании механического разрушения неметаллических покрытий электрических проводов.

Библ.8, ил.3, табл.4.

УДК 519.682

Малкина В.М., Бездитный А.А. **Использование Mechanical APDL для расчета собственных колебаний лопаток турбин в системеконечно-элементного анализа AnsysWorkbench** // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1(84) – Дніпропетровськ, 2013. – С. 125 – 128.

В статье рассматривается процесс проведения модального анализа радиальной крыльчатки средствами Ansys Workbench с использованием параметрической языка программирования APDL (AnsysParametricDesignLanguage).

Библ. 4, ил. 3, табл. 0.

УДК 519.682

Спиринцев В.В., Пыхтеева И.В., Дмитриев Ю.А. **Разработка функциональной схемы процесса автоматизированного проектирования** // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1(84) – Дніпропетровськ, 2013. – С. 129 – 135.

В работе рассмотрен вопрос разработки функциональной схемы процесса автоматизированного проектирования с учетом современных условий производства.

Библ. 3, ил. 8, табл. 0.

УДК 528.8(075.8)

Кавац А.А. **Модельювання процесу формування мікроструктури металевих сплавів** // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1(84) – Дніпропетровськ, 2013. – С. 136 – 140.

Стаття присвячена моделюванню процесу формування мікроструктури металевих сплавів. Досліджується процес формування мікроструктури і вплив форми мікрокристалу на структурні характеристики металевих сплавів.

Бібл. 2, рис. 5, табл. 0.

УДК.621.39

Гнатушенко В.В. **Имитационное моделирование процессов передачи трафика в компьютерных сетях** / В.В. Гнатушенко, Ю.С. Пройдак, Данлади Али // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1(84) – Дніпропетровськ, 2013. – С. 141 – 147.

Создана имитационная модель передачи данных от источника с самоподобным трафиком и проведены исследования влияния параметра формы распределения Парето.

Библ.8, ил. 4.

УДК 622.647.82:51.001.57

Кирія Р. В. **Розробка швидкого алгоритму визначення пропускної спроможності системи «конвеєр – бункер – конвеєр»** / Р.В. Кирія, Т.Ф. Міщенко, Л.В. Камкіна, Ю.В. Бабенко // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1(84) – Дніпропетровськ, 2013. – С. 148 – 156.

Розроблено спрощену марківську модель функціонування системи «конвеєр – бункер – конвеєр», яка дозволяє визначити пропускну спроможність цієї системи при різних співвідношеннях продуктивності надбункерного і підбункерного конвеєрів. Одержані результати теоретичних досліджень достатньо добре співпали з результатами імітаційного моделювання.

Бібл. 6, іл. 4

УДК 514.18

Гавриленко Є.А. **Визначення діапазонів геометричних характеристик дискретно представленої кривої із закономірною зміною кривини** / Є.А. Гавриленко, В.В. Гнатушенко, Ю.В. Холодняк // Системні технології. Регіональний міжвузівський збірник наукових праць. – Випуск 1(84) – Дніпропетровськ, 2013. – С. 157 – 162.

Визначено умови моделювання кривої із монотонною зміною кривини та запропоновано схему призначення положень дотичних, при яких задача моделювання обводу із монотонною зміною кривини має розв'язок.

Бібл. 5, іл. 2

UDC 621.3

Ponomarev I.V., Rybka Yu.M. **Language constructions for parallel asynchronous calculations on the reconfigurable multiprocessor systems** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.3 - 6.

Proposed language constructs of the management section the programming language of parallel calculations on reconfigurable MCS. Tuning of parameters of system configuration at job definition allow to execute processing by system of certain architecture effectively.

Bibl. 2.

UDC 004.4'23

Gerasimov V.V., Ponomarev I.V., Shchetinin I.O. **Optimization of the process to develop an Internet-based applications on Java platform** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.7 - 14.

Considered Play Framework, which simplifies and accelerates the development process of Internet applications on Java platform. The features and benefits of this technology described. The results of the comparison the Play Framework with alternative technologies presented.

Bibl. 5, ill. 2, tabl. 1.

UDC 004.55

Gerasimov V.V., Penkov A.P., Pushkov G.O. **Analysis of technologies develop rich Internet applications based on Java** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.15 - 21.

The analysis of JavaFX 2 technology, which is rapidly progressing. The features of this technology and the prospects for its development are given. The results of the comparison of modern technology development of rich Internet applications on different platforms are provided.

Bibl. 5, ill. 2, tabl. 1.

UDC 004.7:004.45

Grigor'yev V.M. **Virtualization laboratory works on computer networks** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.22 - 30.

The load time of a large number of virtual machines is investigate Qemu operating system company RouterOS Mikrotik. within the host Ubuntu. It is shown that the operating system is RouterOS adequately choice for organizing a virtual laboratory for computer networks

UDC 004.318+536.37

Ivon A.I., Istushkin V.F. **Central processor throughput in the throttling mode** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.31 - 36.

The analysis of the processor throughput in the throttling mode realized by means of the VO2 based critical thermistors was performed. As shown, if the average expenditure of energy on the one computer instruction E_{in} does not exceed the minimum energy for activation of the throttling mode $E_{in\ min}$, the processor throughput conforms to a normal thermal condition. At $E_{in} \geq E_{in\ min}$ the processor throughput decreases in inverse proportion to a value of E_{in} .

Bibl. 4, ill. 2

UDC 621.3

Matveyeva N.A. **Neural networks in the problems classification of defects** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.37 - 42.

This paper introduces the results on research using Kohonen neural networks to the problem classification of signals of non-destructive testing. The trained neural network is modelled on the check set of signals with noise.

Bibl. 4, ill. 3.

UDC 004.422,833

Atamanenko I.I., Sivtsov D.P. **The use of assembly language modules to increase performance C / C + + programs symmetric encryption** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.43 - 52.

The article discusses the possibility of increasing the productivity of C / C + + programs for encrypting information based on symmetric cryptography algorithms. It has been found that a significant effect can be achieved by implementing a piece of software in assembly.

Bibl. 7, Fig. 3.

UDK 621.391.26(075)

Krokhin V.V. Mitikov N.Yu. **About the study of the filter`s design quality** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.53 - 58.

A width of the transition band between the transmission and detention zone is suggested to be used in the evaluation of the filter`s design quality as an extra parameter together with the levels of the pulsations in the zones for the purposes of the comparative analysis. A methodology for calculating of the characteristic is given together with the results of analysis of the filter`s design quality using six different weight functions.

UDK 519.281:620.179(035)

Tverdostup N. **Effectiveness of adaptive algorithm smoothing in effect mesh factors** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.59 - 66.

Influence of exponential smoothing on an error of result of measurement of inductance of the parametrical sensor located on a surface iron ore piece is investigated. It is established that smoothing almost much reduces an error of result of measurement at stirring influence of heterogeneity of a magnetic field of the sensor.

Bibl. 5, ill. 3.

UDC 624.04

Volkovsky O.S., Petrenko Y.Y. **Convert media files on example of audio in operating system Windows 8 using Windows Store library. Comparison with ffmpeg** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.67 - 74.

Investigated the conversion of audio files using the Windows Store libraries, compared with ffmpeg library. Written test program that can be used to develop applications for processing media files. Compared Windows Store ffmpeg libraries and library while converting audio.

Bibl. 3, ill. 7, table. 3.

UDC 614.2+574/578+004.38

Litvinov A.A., Pavlenko M.V., Matveev A.A. **A technology for information analysis of surgical clinic** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.75 - 82.

The given work is devoted to a strategy and technology of surgery clinic information analysis which is concentrated on tasks of activities and resources planning by the manager and clinical standards effectiveness estimation by the expert. First, main problems connected with the tasks were stated and discussed. Then, a strategy of analysis was suggested and the results of its application were discussed.

Bibl. 5, ill. 5. tabl.2.

UDC 614.2 +574 / 578 +004.38

Litvinov A.A., Vyakilov A.S., Brezhnev A.I. **A features of construction of medical information system oriented to standards** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.83 - 89.

In this paper, we consider the option of a flexible information system to support the standards. The result of the approach is the generation of standards based on the current practice of treatment, which allows for flexibility of medical information system is an integral part of quality control systems and the cost of treatment.

UDC 621.3.083.92

Martynovych L.Y., Stecenko O.O. **Designing automated scheduling classes in college** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.90 - 97.

A standalone software for scheduling classes in college, subject to the limitations on the number of daily and weekly on-loading for students, non-gaps between classes for students and weekly load limits for teachers based on academic and work plans.

Bibl.6, ill.3.

UDC 616.12

Alekseev M.A., Akhmetshin K.A. **Generalized phase portrait synthesis of multichannel nonlinear systems at singular value decomposition basis of embedded vectors** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.398 - 106.

Information possibilities a new method of nonlinear dynamical systems analysis are considered. Results experimental testing of the method are presented.

Bibl. 7, ill. 9

UDC 616.12

Akhmetshina L.G., Akhmetshin A.M., Udovik I.M. **Method interpolation of spatial distributed data, based on the Kohonen network** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.107 - 113.

It has been researched informative possibilities of the spatial distributed data in the irregular grid, based on the clusterization, used by self-organizing Kohonen network. The results of modeling by the geophysical surfaces on the model and real data has been presented.

Bibl. 3, ill. 4

UDC 669.02/09:519.28

Yasev A.G. **The summary System of mathematical modeling of mechanical destruction of nonmetallic coverings of electric wires** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.114 - 124.

Use of system of mathematical modeling (which includes mathematical modeling, development of is constructive-technological offers, optimization of parameters) at the research of mechanical destruction of nonmetallic coverings of electric wires is described.

UDC 519.682

Malkina V.M., Bezditniy A.A. **The Using of Mechanical APDL fo the calculation of turbineblades'naturalos cillation sat the finite-element analysis system Ansys Workbench** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.125 - 128.

This article provides a process of radial impeller'smod alanalysis in Ansys Workbench using parametr icprogramming language APDL (AnsysParametricDesignLanguage).

Bibl. 4, il. 3, Tab. 0.

UDC 519.682

Sprintsev V.V., Pyhteeva I.V., Dmitriev Y.O. **Working out of the function chart of process of the automated designing** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.129 - 135.

In work the question of working out of a function chart of process of the automated designing taking into account modern conditions of manufacture is considered.

Bibl. 3, il. 8, Tab. 0.

UDC 528.8(075.8)

Kavats A.A. **Simulation of the formation of the microstructure of metal alloys** // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.136 - 140.

The article deals with modeling of the formation of the microstructure of metal alloys. We study the formation of the microstructure and the effect of micro-crystal form on the structural characteristics of metal alloys.

Bibl. 4, Fig. 5, Tabl. 0.

UDK 621.39

Gnatushenko V.V. **Simulation modeling of traffic in computer networks** / V.V. Gnatushenko, Danladi Ali // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.141 - 147.

A simulation model of the data on the source of self-similar traffic impact studies and shape parameter of Pareto distribution.

Bibl. 8, ill. 4.

UDK 622.647.82:51.001.57

Kiriya R. **Develop a fast algorithm for determining the capacity of the "conveyor belt – the bunker – conveyor"** / R. Kiriya, T. Mishenko, Yu. Babenko // System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.148 - 156.

A simplified Markov model of the functioning of the "conveyor belt - the bunker - conveyor", which allows determining the capacity of the system at different ratios of capacities

over a bunker and a bunker conveyors. The results of theoretical research rather well with the results of the simulation.

Refs. 6, ill. 4

UDC 514.18

Gavrilenko E.A., Kholodnyak Yu.V. **Determine of geometrical characteristics ranges of the discretely presented curve with law of curvature change** / E.A. Gavrilenko, V.V. Gnatushenko, Yu.V. Kholodnyak// System technologies. N 1(84) - Dnipropetrovsk, 2013.- P.157 - 162.

The conditions modeling of curve with a monotonous change of curvature and the scheme of the destination of the provisions tangents at which the problem of modeling the contours with a monotonous change of curvature is considered.

Bibl. 5, ill. 2.

АВТОРЫ ВЫПУСКА

Рыбка Ю.М. – Дніпропетровський національний університет імені Олеся Гончара.

Герасимов В.В. – Дніпропетровський національний університет імені Олеся Гончара.

Бездитный Андрей Александрович - к.т.н., ассистент кафедры «Прикладной геометрии и информационных технологий проектирования им. В.Н. Найдыша» Таврического государственного агротехнологического университета.

Спиринцев Вячеслав Васильевич - к.т.н., доцент кафедры «Прикладной геометрии и информационных технологий проектирования им. В.Н. Найдыша» Таврического государственного агротехнологического университета.

Пономарьов І.В. – Дніпропетровський національний університет імені Олеся Гончара.

Щетінін І.О. – Дніпропетровський національний університет імені Олеся Гончара.

Герасимов В.В. – Дніпропетровський національний університет імені Олеся Гончара.

Пеньков А.П. – Дніпропетровський національний університет імені Олеся Гончара.

Гнатушенко Виктория Владимировна — к.т.н., доцент, доцент кафедры информационных технологий и систем Национальной металлургической академии Украины.

Данлади Али — аспирант кафедры информационных технологий и систем Национальной металлургической академии Украины.

Пушков Г.О. – Дніпропетровський національний університет імені Олеся Гончара.

Григорьев В.М. – Дніпропетровський національний університет імені Олеся Гончара.

Ивон А.И. – Дніпропетровський національний університет імені Олеся Гончара.

Истушкин В.Ф. – Дніпропетровський національний університет імені Олеся Гончара.

Матвєєва Н.А. – Дніпропетровський національний університет імені Олеся Гончара.

Крохин В.В. – Дніпропетровський національний університет імені Олеся Гончара.

Митиков Н.Ю. – Дніпропетровський національний університет імені Олеся Гончара.

Атаманенко І.І. – Дніпропетровський національний університет імені Олеся Гончара.

Сівцов Д.П. – Дніпропетровський національний університет імені Олеся Гончара.

Твердоступ Н.И. – Дніпропетровський національний університет імені Олеся Гончара.

Волковский О.С. – Дніпропетровський національний університет імені Олеся Гончара.

Петренко Я.Ю. – Дніпропетровський національний університет імені Олеся Гончара.

Литвинов А.А. – Дніпропетровський національний університет імені Олеся Гончара.

Кирия Руслан Виссарионович - к.т.н., ст. научн. сотрудник, Институт геотехнической механики им. Н.С. Полякова НАН Украины.

Мищенко Тамара Федоровна - ведущий инженер, Институт геотехнической механики им. Н.С. Полякова НАН Украины.

Бабенко Юлия Васильевна – аспирант, Национальная металлургическая академия Украины.

Брежнев А.И. – Дніпропетровський національний університет імені Олеся Гончара.

Вякилов А.С. – Дніпропетровський національний університет імені Олеся Гончара.

Литвинов А.А. – Дніпропетровський національний університет імені Олеся Гончара.

Павленко М.В. – Дніпропетровський національний університет імені Олеся Гончара.

Матвеев А.А. – Дніпропетровський національний університет імені Олеся Гончара.

Мартинович Л.Я. – Дніпропетровський національний університет імені Олеся Гончара.

Стеценко О.О. – Дніпропетровський національний університет імені Олеся Гончара.

Кавац Алена Александровна - доцент кафедры «Информационных технологий и систем». Национальная металлургическая академия Украины.

Алексеев М.А. – Дніпропетровський національний університет імені Олеся Гончара.

Ахметшин К.А. – Дніпропетровський національний університет імені Олеся Гончара.

Ахметшина Л.Г. – Дніпропетровський національний університет імені Олеся Гончара.

Малкина Вера Михайловна - д.т.н., профессор, академик АН ВО Украины. Заведующий кафедрой «Прикладной геометрии и информационных технологий проектирования им. В.Н. Найдыша» Таврического государственного агротехнологического университета. Декан факультета инженерии и компьютерных технологий Таврического государственного агротехнологического университета.

Пыхтеева Ирина Викторовна - к.т.н., доцент кафедры «Прикладной геометрии и информационных технологий проектирования им. В.Н. Найдыша» Таврического государственного агротехнологического университета.

Ахметшин А.М. – Дніпропетровський національний університет імені Олеся Гончара.

Удовик И.М. – Дніпропетровський національний університет імені Олеся Гончара.

Ясев Александр Георгиевич – к.т.н., профессор кафедры прикладной математики и вычислительной техники Национальной металлургической академии Украины.

Дмитриев Юрий Алексеевич - старший преподаватель кафедры «Прикладной геометрии и информационных технологий проектирования им. В.Н. Найдыша» Таврического государственного агротехнологического университета.

Гавриленко Євген Андрійович – к.т.н., докторант, Таврійський державний агротехнологічний університет.

Гнатушенко Володимир Володимирович — д.т.н., професор, завідувач кафедри автоматизованих систем обробки інформації Дніпропетровського національного університету імені Олеся Гончара (м. Дніпропетровськ).

Холодняк Юлія Володимирівна – аспірант, Таврійський державний агротехнологічний університет.