

РЕДАКЦІЙНА КОЛЕГІЯ:

Головний редактор:

проф., д.т.н. Рудь В. Д. (м. Луцьк)

Перший заступник головного редактора:

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

Другий заступник головного редактора:

доц., к.т.н. Лотиш В. В. (м. Луцьк)

доц., к.пед.н. Панасюк Н.Л. (м. Луцьк)

Редактор-коректор:

доц., к.т.н. Мельник К.В. (м. Луцьк)

Члени редакційної колегії:

проф., д.т.н. Божидарнік В. В.

(м. Луцьк)

проф., д.пед.н. Горбатюк Р. М.

(м. Тернопіль)

проф., д.т.н. Мазур М. П. (м. Луцьк)

проф., д.т.н. Мельник А. О. (м.Київ)

проф., д.т.н. Пальчевський Б. О.

(м. Луцьк)

проф., д.пед.н. Поліщук Ю. Й.

(м. Тернопіль)

проф., д.пед.н. Романишина Л. М.

(м. Тернопіль)

проф., д.пед.н. Свистун В. І. (м.Київ)

проф., д.т.н. Сидорчук О. В. (м.Київ)

проф., д.т.н. Тарасенко В. П. (м.Київ)

доц., к.т.н. Гуменюк Л. О. (м. Луцьк)

доц., к.т.н. Кузнецов Р. М. (м. Луцьк)

доц. к.т.н. Пех П. А. (м. Луцьк)

доц. к.т.н. Повстяной О. Ю. (м. Луцьк)

доц. к. пед.н. Потапюк Л. М. (м. Луцьк)

доц. к.т.н. Решетило О. М. (м. Луцьк)

Адреса редколегії:

Луцький національний технічний
університет,
кафедра комп'ютерної інженерії,
вул. Львівська, 75, ауд. 141, м. Луцьк,
43018

тел. (0332) 74-61-15

E-mail: cit@Intu.edu.ua

ekaterinamelnik@gmail.com

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

№14

Журнал засновано у грудні 2010 р.

2014

Свідоцтво про реєстрацію КВ № 16705–5277 Р.

Засновник: Луцький національний технічний університет
Рекомендовано до друку Вченою радою Луцького національного
технічного університету (протокол №8 засідання від 25.03.2014)

Журнал рішенням президії ВАК України від 30 березня 2011
№1-05/3 включено в перелік наукових фахових видань.

Журнал має російський індекс наукового цитування (РІНЦ)

ISSN:978-617-672-040-9

ЗМІСТ

Інформатика та обчислювальна техніка

Веремко П.Д., Мельник К.В., Багнюк Н.В.	
Прогнозування енергетичного комплексу Волинської області на основі нейронних мереж та моделі АРСС.....	5
Говорущенко Т.О. Забезпечення якості програмних продуктів.....	10
Гринюк С.В., Топчевська К.Е., Шолом П.С. Кодування інформації за допомогою самокорегуючого штрихового коду НЕМ-8. В.....	21
Губаль Г. М. Бібліографічні інструменти та деякі правила для створення математичних текстів у видавничій системі L^AT_EX.....	27
Губаль Г. М. Вишукані команди для створення математичних текстів у видавничій системі L^AT_EX.....	31
Губіш С.А., Здолбницька Н.В., Здолбницький А.П.	
Поширення сигналу на відкритому просторі.....	36
Дудік О.В. Аспекти безпеки програмування на РНР.....	41
Здолбницький А.П. Об'єктно-орієнтована програмна реалізація математичної моделі деформування деревини у процесі неізотермічного вологоперенесення.....	47
Здолбницький А.П., Місюк В.В. Система на кристали Amber ARM в базисі ПЛІС.....	57
Коцюба А.Ю., Матвіюк В.В. Імітаційне моделювання симуляції розіграшів національної лотереї “Супер Лото”.....	62
Красій А.В. Моделювання процесу прогнозування характеристик програмного забезпечення на основі аналізу специфікацій.....	66

ЗМІСТ

Інформатика та обчислювальна техніка

Мельник В.А. Аналіз та дослідження ефективності застосування апаратних прискорювачів для побудови реконфігурованих комп'ютерних систем.....	77
Мельник В.М., Мельник К.В., Жигаревич О.К. Мережева емуляція для Java-додатків за допомогою об'єднань сокетів.....	87
Пех П. А., Кузава О. В., Мацібора А. С., Шафета Ю. О. Програмний комплекс для дослідження вхідних потоків систем масового обслуговування (СМО).....	92
Пех П. А., Середа А. О., Калінін Б. Ю., Лавренюк О. С. Програмний комплекс для дослідження процесу оброблення вимог у системах масового обслуговування (СМО).....	100
Савенко О.С., Лисенко С.М., Кришук А.Ф. Ефективність діагностування комп'ютерних систем на наявність бот-мереж антивірусною мультиагентною системою.....	109
Христинець Н.А., Міскевич О.І. Shell-програмування процесів фоновому режиму в ОС Linux.....	115
Шолом П.С., Гринюк С.В., Семчук С.Р. Система контролю просторовим положенням монітора.....	119

Автоматика та управління

Бортник К.Я. Інноваційні технології в енергозбереженні по компенсації реактивної потужності.....	124
Брей В.А., Н.В. Здолбіцька, А.П. Здолбіцький. Звукова сигналізація на PIC16F84 контролері.....	128
Вісин О.О. Аналіз сучасних комп'ютеризованих систем управління охороною праці. Жигаревич О.К. Дослідження складних систем засобами імітаційного моделювання.....	139
Крестьянполь Л.Ю. Розробка комп'ютерної моделі для вирішення задачі оптимізації системи захисту паковань алкогольної продукції.....	144
Кулінченко Г.В., Леонтьєв П.В., Ляпощенко О.О. Ідентифікація моделі процесу низькотемпературної сепарації природного газу.....	149
Полетайкин А.Н., Гузь О.А., Каганюк А.К. Интеллектуальные средства и технологии инфраструктуры городского дорожного движения.....	155
Савеленко Г.В., Ермолаев Ю.О. Дослідження робочого процесу розмірної обробки дугою на верстаті з електромеханічним приводом.....	164
Соколовськія Я.І., Семенюк В.Я., Мокрицька О.В. Математичне та програмне забезпечення моделювання в'язкопружного стану деревини у процесі сушіння з врахуванням багатофазності.....	170
Яремко С.А., Коваленко В.В. Дослідження проблем впровадження сучасних інформаційних систем аудиту в контрольно-ревізійній діяльності.....	179

Педагогічні науки

Панасюк О.П. Теоретичні засади сприйняття інформації про розвиток моральних цінностей.....	183
--	-----

Управління проектами

Сидорчук О.В., Луб П.М., Грабовець В.В., Спічак В.С. Алгоритм статистичної імітаційної моделі ґрунтообробно-посівних процесів літньо-осіннього періоду....	188
Тригуба А.М. Структура системи управління технологічно інтегрованими програмами молочарства.....	193

УДК 004.8

П.Д. Веремко, К.В. Мельник, Багнюк Н.В.
Луцький національний технічний університет

ПРОГНОЗУВАННЯ ЕНЕРГЕТИЧНОГО КОМПЛЕКСУ ВОЛИНСЬКОЇ ОБЛАСТІ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ ТА МОДЕЛІ АРПСС.

Мельник К.В., Веремко П.Д., Багнюк Н.В. Прогнозування енергетичного комплексу Волинської області на основі нейронних мереж та моделі АРПСС. Проведено аналіз та прогнозування споживання природного газу у Волинській області на різні періоди часу на основі нейронних мереж та моделі АРПСС. На основі порівняння відносних похибок прогнозу вибрана оптимальна модель.

Ключові слова: нейронні мережі, багатoshаровий перцептрон, похибка, модель, прогнозування.
Рис. 6. Таб. 2 Літ.6

Мельник К.В., Веремко П.Д., Багнюк Н.В. Исследование и прогнозирование энергетического комплекса Волинской области на основе нейронных сетей. Проведен анализ и прогнозирование потребления природного газа в Волинской области на разные периоды времени на основе нейронных сетей и модели АРПСС. На основе сравнения относительных погрешностей прогноза выбрана оптимальная модель.

Ключевые слова: нейронные сети, многослойный перцептрон, погрешность, модель, прогнозирование.
Рис. 6. Таб. 2 Лит.6

Melnik K.V., Veremko P.D., Bagnyk N.V. Research and Forecasting Energy Complex Volyn region based on neural networks. The analysis and forecasting of natural gas consumption in the Volyn region in different time periods based on neural networks and model ARPSS. On the basis of the relative prediction error is selected optimal model.

Key words: neural networks, multilayer perceptron, error, the model, prediction.
Fig.6 Tabl.2 Ref.6

Постановка проблеми. На даний час в Україні проблема газопостачання стоїть досить гостро. Прогнозування споживання і попиту на газ є основою не тільки для планування закупівель природного газу в газотрейдерів, але, що не менш важливо, для розробки заходів з управління енергоспоживанням у муніципальному господарстві, особливо в опалювальний період. Основним завданням є отримання більш точного прогнозу на певний період часу з урахуванням максимальної кількості чинників, що впливають на попит природного газу. Недооцінка навантаження може призвести до зростання використання пікового обсягу, що, у свою чергу, веде до зниження резервів, а завищений прогноз навантаження може призвести до необґрунтованого збільшення резерву, і, отже, собівартості газу. Тому дуже важливим є оперативне прогнозування навантаження мережі.

Актуальність прогнозування режимів газоспоживання в розрізах часу безсумнівна, так як обґрунтовані прогнози є ефективним інструментом перспективного планування та управління в галузі [1].

Аналіз останніх досліджень і публікацій. В роботі Вороновського Г.К. [2] представлена нейронна модель для короткострокового прогнозування зв'язного споживання теплової та електричної енергії. В ході багатьох експериментів з моделлю була виявлена закономірність: зі збільшенням дефіциту тепла споживання електроенергії збільшується і, навпаки – зі зменшенням дефіциту тепла споживання електроенергії зменшується.

Огляд і опис двох моделей прогнозу викладено в статті Філатова А.Ю., в якій було запропоновано статистичний і евристичний методи, які визначають як впливають ті чи інші фактори на створення нейронної мережі. В роботі зазначалося, що для прогнозування споживання газу не існує загального, єдиного методу: кожне виробництво містить індивідуальні технологічні цикли, які в сумі дають унікальний часовий процес. Існуючі системи не дозволяють з усієї безлічі засобів і методів автоматично вибрати найбільш ефективний шлях вирішення, тому користувач-аналітик сам приймає рішення про вибір методу. Існує ряд переваг при застосуванні різних методів, а саме, можна зробити порівняння їх результатів.

Було зроблено висновок, що робота системи буде ефективна, прогноз достовірний і точний тільки тоді, коли вона буде створюватися під конкретний об'єкт і завдання із застосуванням необхідних оптимальних методів, що враховують всі впливають параметри.

Метою і задачами дослідження є побудова моделі прогнозування з урахуванням максимального числа факторів, що впливають на споживання природного газу.

Для досягнення поставленої мети необхідно було вирішити такі завдання:

1. Збір даних споживання природного газу, їх системний аналіз з метою виявлення особливостей зв'язного споживання.

2. Дослідити теоретичні та методичні підходи до прогнозування споживання природного газу.

3. Виявити і систематизувати фактори, що впливають на споживання природного газу.

4. Здійснити прогнозування часового ряду на основі моделей: експоненціального згладжування, АРПСС і нейронних мереж.

Дослідження проводилося на основі статистичних показників споживання природного газу у Волинській області. Були зафіксовані дані помісячно за період з 2006 по 2013 рр. Отриманий обсяг даних дозволяє побудувати середньостроковий прогноз з максимальної точності. Поставлена задача розділяється на ряд підзадач: провести описовий (графічний) аналіз часового ряду; дослідити часовий ряд на наявності постійних і регулярних компонент; здійснити прогноз часового ряду [3].

Аналіз проводився за допомогою статистичного пакета STATISTICA. Перед прогнозуванням необхідно провести описовий аналіз часового ряду. На рис. 1 представлений графік споживання газу з періодом 1 місяць.

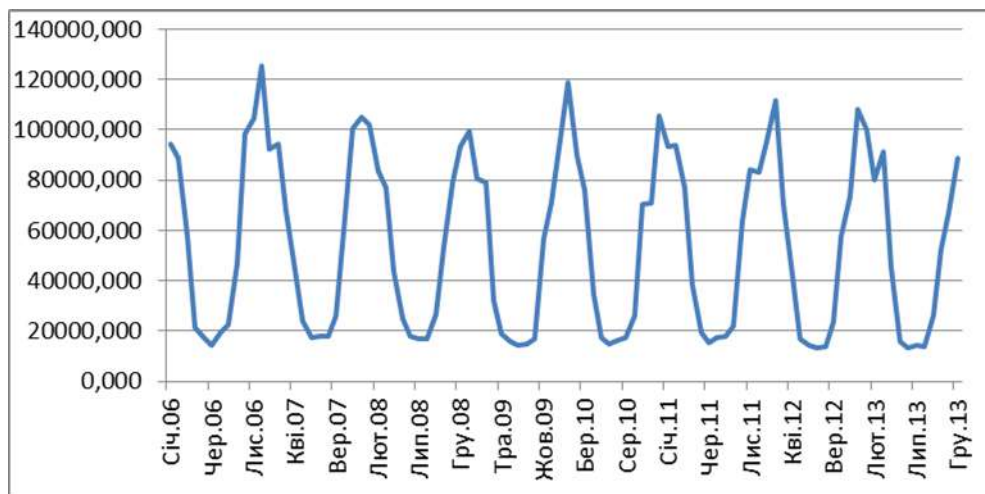


Рис.1. Графік помісячного споживання природного газу у Волинській області

З графіка видно, що часовий ряд є стаціонарним, коливається біля деякого постійного значення, а також спостерігаються явні піки які відповідають сезонності.

Першим кроком буде виділення періоду коливань сезонної компоненти. Це може бути зроблено за допомогою спектрального аналізу. На рис. 2 наведений графік періодограми.

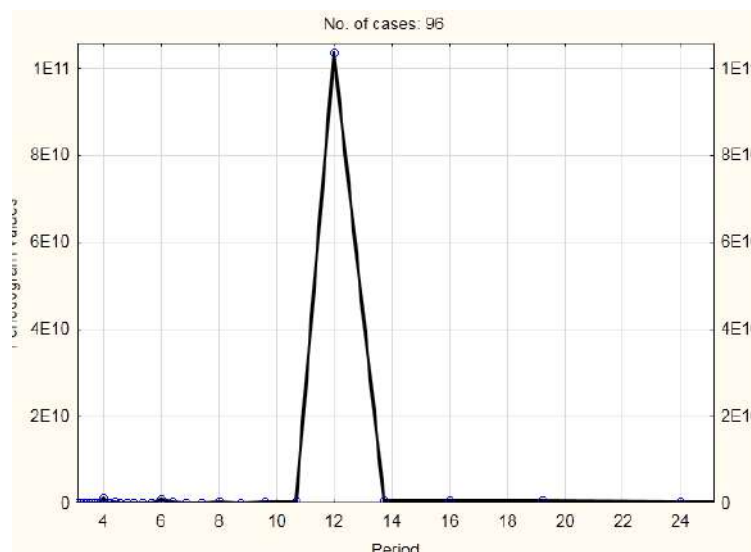


Рис.2. Періодограма часового ряду

Як видно з графіку спостерігається наявний пік, що відповідає 12 місяцям. В середині одного сезонного циклу споживання газу змінюється відповідно за певними законами, які можуть бути виявлені за допомогою класичної сезонної декомпозиції. Результати дослідження наведені на рис. 3.

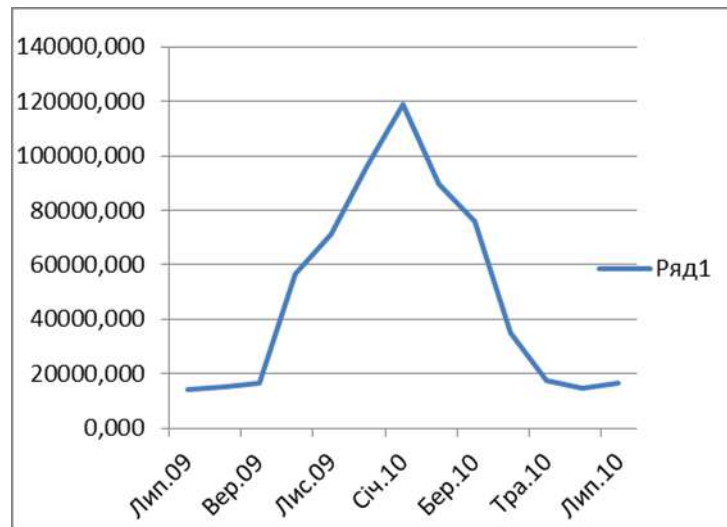


Рис.3. Споживання природного газу у Волинській області за один рік.

Як видно з графіка, споживання газу максимально у січні. При цьому спостерігається трендові падіння загального споживання з кінця лютого місяця, яке пов'язане із зростанням середньодобової температури.

Перший прогноз побудований на основі моделі АРПСС. На рис. 4 наведені основні параметри цієї моделі. Як відомо, модель авторегресії проінтегрованого ковзного середнього, застосовується при оцінці нестационарних часових рядів, які характеризуються математичним сподіванням, дисперсією, автоковаріацією і автокореляцією.

В основі моделі АРПСС лежать два процеси:

- 1) процес авторегресії;
- 2) процес ковзного середнього.[4]

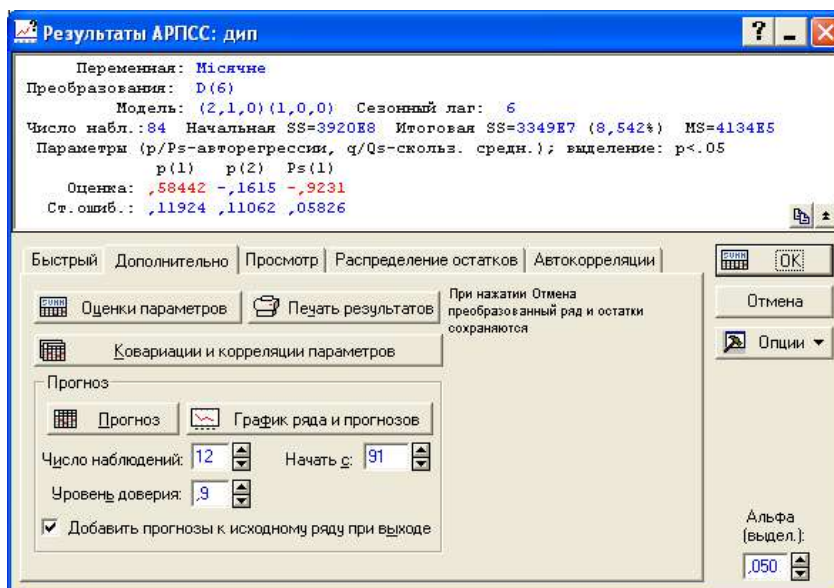


Рис.4 Основні параметри АРПСС моделі

Як видно з табл. 1 АРПСС модель дає високу похибку навіть при короткостроковому прогнозі.

Наступний прогноз будувався на основі нейронних мереж. Для налаштування нейронних мереж навчальна вибірка складала 60%, а контрольна – 40%. Остання вибірка використовувалася для зупинки процесу навчання нейронної мережі, тобто коли похибка на контрольній вибірці починає зростати це є ознакою того що, дана модель перенавчається, і тоді необхідно зупинити навчання. Задано також розмір вікна (12), як правило він задається не меншим ніж період ряду. Розмір вікна - це кількість спостережень, які є входними в нейронну мережу[5].

Таблиця 1. – Похибка моделі АРПСС

	6-місяців	Похибка	12-місяців	Похибка	24-місяців	Похибка
1	14463,8	0,022	108508,3	0,085	74739,45	0,216
2	18273,1	0,331	110608,9	0,214	82175,30	0,266
3	28084,2	0,082	66986,8	0,162	71534,55	0,019
4	57031,9	0,081	42851,4	0,059	40742,71	0,035
5	69564,3	0,036	21035,7	0,187	25625,22	0,530
6	100227,6	0,133	22386,1	0,647	21618,25	0,488
7			21549,2	0,523	22355,75	0,672
8			22361,2	0,628	23361,41	0,669
9			27332,1	0,053	26464,52	0,123
10			56604,5	0,073	61759,36	0,067
11			68957,5	0,027	79127,00	0,077
12			100904,8	0,140	77829,58	0,282
...
24					73390,55	0,171
Середня відносна похибка моделі		10,1%		23,3%		32,1%

Побудовані мережі типу «Багатошаровий перцептрон», в яких функція похибки вибрана сума квадратів. Експериментально встановлено, що найменшу похибку дає мережа, в якій у прихованому шарі має 6 нейронів. В другій таблиці наведена відносна середня похибка прогнозу, який побудовано на основі мережі з різною кількістю нейронів у прихованому шарі.

Таблиця 2. – Середня відносна похибка прогнозу на 12 місяців

	Місячне споживання газу тис.куб.З	М 12-6-1	М 12-2-1	М 12-4-1	М12-5-1
85	100017,346	99395,187	96530,067	101151,386	99941,696
86	91104,744	100653,004	93953,974	100025,537	100467,259
87	79966,928	76576,006	77170,695	76079,400	76077,383
88	45523,491	47488,761	47391,565	46219,625	47365,154
...
96	88498,879	86780,104	91310,159	90833,955	87695,167
Середня відносна похибка методу на 12 місяців		6,8%	10,2%	7,4%	7,3%

На рис. 5 наведені параметри оптимальної мережі, яка має 12 входів, 6 нейронів у прихованому шарі і один вихід. Продуктивність мережі рівна 0,967776. Ця значення є величиною кореляції між вихідним рядом і прогнозованим, відповідно на кожній вибірці. Як відомо, чим ближче це значення до 1, тим точнішим буде прогноз. Також представлені числові значення функціоналу похибки як на навчальній так і на контрольній вибірці, вказано номер ітерації (8), на якій алгоритм навчання зупиняється, а також функція похибки і функції активації[6].

Summary of active networks (дип)											
Include cases: 1:84											
Index	Net. name	Training perf.	Test perf.	Validation perf.	Training error	Test error	Validation error	Training algorithm	Error function	Hidden activation	Output activation
2	MLP 12-6-1	0.967776	0.972060		29872289	27080142		BFGS 8	SOS	Tanh	Identity

Рис. 5 Параметри оптимальної мережі

На рис. 6 штриховою лінією показані прогнозовані значення газу на 12 місяців на основі вибраної оптимальної мережі. Як показано в таб. 2 відносна похибка такого прогнозу становить 6,8%.

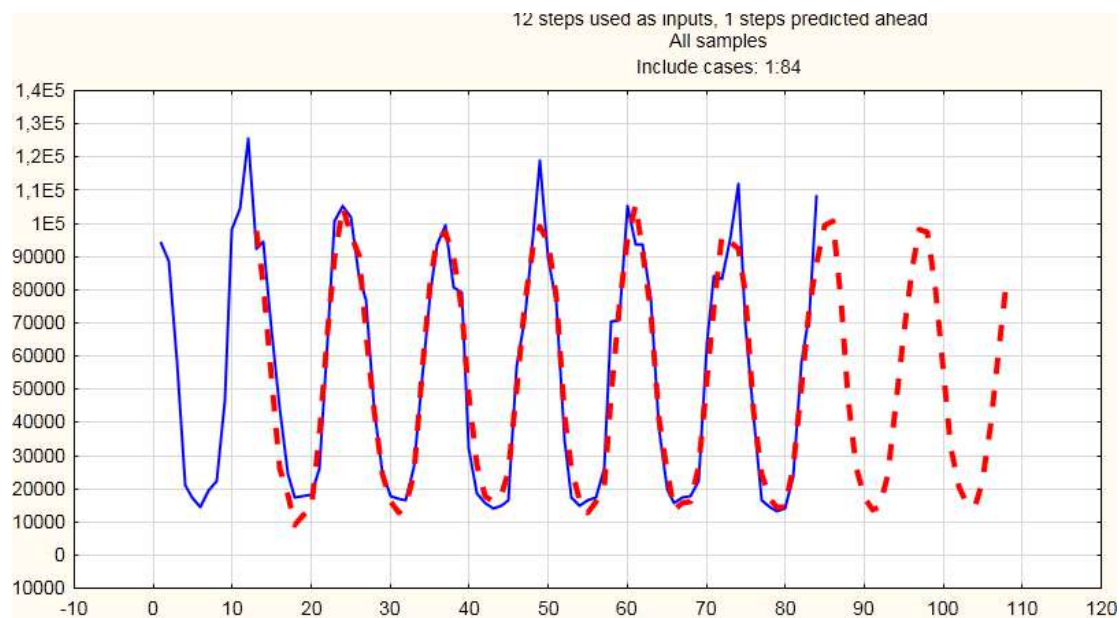


Рис. 6. Графік прогнозу на 12 місяців вперед

Висновки. Отже, порівнюючи відносні похибки прогнозу на основі різних моделей можна зробити висновок, що вдало налаштована нейронна мережа дає більш точний прогноз на довший проміжок часу в порівнянні з класичними статистичними методами. Слід зазначити що, наведений обсяг статистичних даних дозволяє здійснювати середньострокові прогнози. При збільшенні тривалості прогнозу до 24 місяців похибка зростає до 10,5%.

Список використаних джерел

1. Методы прогнозирования URL: <http://www.neuroproject.ru/>
2. Нейросетевая модель связанного потребления тепловой и электрической энергии крупным жилым массивом города /Г.К. Вороновский и др. // Вестник ХГПУ, Серия «Электротехника, электроника и электропривод», Выпуск №113.–Харьков 2000.
3. STATISTICA, аналіз даних, статистичний аналіз, статистичний аналіз даних, візуалізація, прогнозування, моделювання, сегментація, дослідження, аналітичні рішення. URL: <http://www.statsoft.ru>
4. Прогнозування об'ємів споживання електроенергії. URL: http://statsoft.ru/solutions/ExamplesBase/branches/detail.php?ELEMENT_ID=644
5. Вороновский Г. К., «Усовершенствование практики оперативного управления крупными теплофикационными системами в новых экономических условиях»: «Харьков», 2002— 240 с. URL: www.isde.kharkiv.com/Archive/Books/part3.htm
6. Головки В.А. Нейронні мережі: навчання, організація і застосування.// Головки В.А. – М.: Радіотехніка, 2001.- 256 с.

УДК 004.891.3: 004.3

Т.О.Говорущенко

Хмельницький національний університет

ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ПРОГРАМНИХ ПРОДУКТІВ

Т.О.Говорущенко. Забезпечення якості програмних продуктів. Стаття присвячена постановці проблеми в галузі забезпечення якості програмних продуктів. В якості одного з підходів, які дозволяють вирішити поставлену проблему, автор розглядає backcasting-підхід, який допомагає прийняти зважені, фундаментальні, обгрунтовані, легітимні, якісні рішення при виборі програмних проектів, а також може підвищувати спільну відповідальність і усувати конфлікти цілей та інтересів зацікавлених сторін за рахунок проблемної орієнтованості, розвитку майбутнього бачення та активної участі усіх зацікавлених сторін на всіх етапах підходу.

Ключові слова: програмне забезпечення (ПЗ), програмний проект, якість програмного забезпечення, забезпечення якості ПЗ, backcasting-підхід, зацікавлені сторони.

Форм. 0. Табл. 1. Рис. 11. Літ. 6.

Т.А.Говорущенко. Обеспечение качества программных продуктов. Статья посвящена постановке проблемы в отрасли обеспечения качества программных продуктов. В качестве одного из подходов, которые разрешают решить поставленную проблему, автор рассматривает backcasting-подход, который помогает принять взвешенные, фундаментальные, обоснованные, легитимные, качественные решения при выборе программных проектов, а также может повышать совместную ответственность и устранять конфликты целей и интересов заинтересованных сторон благодаря проблемной ориентированности, развитию будущего видения и активному участию всех заинтересованных сторон на всех этапах подхода.

Ключевые слова: программное обеспечение (ПО), программный проект, качество программного обеспечения, обеспечение качества ПО, backcasting-подход, заинтересованные стороны.

Форм. 0. Табл. 1. Рис. 11. Лит. 6.

Т.О.Говорущенко. Software Quality Assurance. The article is devoted to the formulation of the problem of the software quality assurance industry. As one of the approaches, that allow to solve the problem, the author considers the backcasting-approach, that helps to make informed, fundamental, grounded, legitimate, quality decisions at the selecting of software projects, can improve joint responsibility and eliminates the conflicts of stakeholders goals and interests through problem orientation, development of future vision and active participation of all stakeholders at all approach stages.

Key words: software, software project, software quality, software quality assurance, backcasting-approach, stakeholders.

Form. 0. Tab. 1. Fig. 11. Lit. 6.

Вступ. Ключовим фактором забезпечення ефективного застосування програмних продуктів є ретельне оцінювання та досягнення високих значень показників якості. Програмне забезпечення (ПЗ) розробляють вже понад п'ятдесят років, і за цей період задачі, які воно може вирішувати, рівень їх складності та форми представлення отриманих результатів кардинально змінилися. Але й дотепер розробка якісних програмних продуктів не стала нормою. Джерела несправностей сучасного ПЗ вкрай різноманітні, і це лише ускладнює проблему, а також збільшує її масштаб та вартість. Постійне підвищення складності функцій, реалізованих програмами в інформаційних системах, безпосередньо призводить до збільшення їх обсягу та трудомісткості створення.

Звісно, за останні роки програмна індустрія досягла такого рівня розвитку, при якому вимоги до забезпечення якості стали обов'язковим пунктом договорів на предмет розроблення програмних систем. У відповідності до стандартів [1], забезпечення якості - це сукупність планованих та систематичних заходів, необхідних для впевненості в тому, що продукція або процеси задовольняють певні вимоги до якості. Система забезпечення якості стосовно програмних засобів - це сукупність методів та засобів організації керуючих та виконавчих підрозділів підприємства, які беруть участь в проектуванні, розробленні та супроводженні комплексів програм з метою надання їм властивостей, що забезпечують задоволення певних потреб замовників та споживачів при мінімальних або допустимих витратах ресурсів [2]. Гарантування якості ПЗ – проблема, вирішення якої потребує комплексного дослідження за наступними напрямками: 1) розроблення засобів аналізу і оцінювання якості ПЗ на різних етапах його життєвого циклу (ЖЦ); 2) визначення і управління параметрами, які впливають на якість ПЗ на всіх етапах його ЖЦ, із залученням всіх зацікавлених осіб. Потреба у забезпеченні якості ПЗ випливає з двох причин: 1) помилки та відмови критично-важливого ПЗ загрожують катастрофами, які призводять до людських жертв, екологічних катаклізмів, значних економічних та часових втрат; 2) застосування неякісного ПЗ призводить до фінансових збитків.

Постановка проблеми. Наразі істотною і невід'ємною властивістю програмних продуктів є їх складність. Постійне зростання складності функцій ПЗ неминуче призводить до збільшення їх обсягу

та трудомісткості створення. Щільність помилок, поява яких очікується в ПЗ різного розміру, представлена на рис.1 [3].

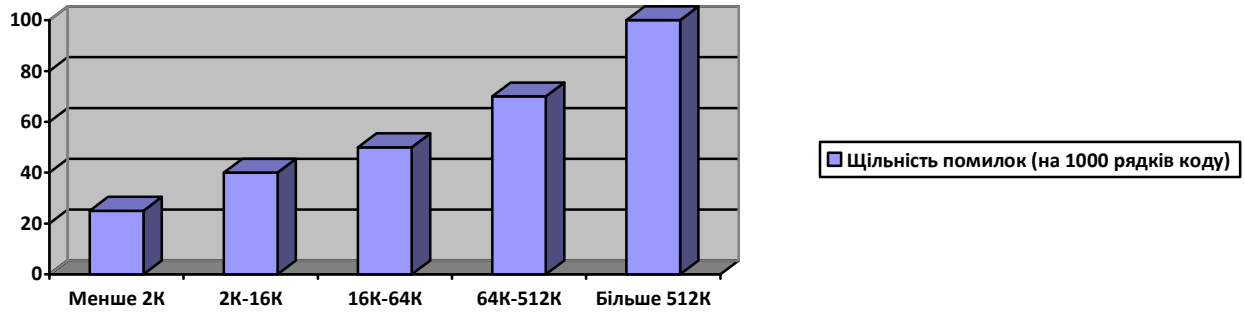


Рис.1. Типова щільність помилок для ПЗ різного розміру

З рис. 1 слідує, що сучасне ПЗ обсягом в мільйони рядків коду в принципі не може бути безпомилковим. Проблема полягає в тому, щоб забезпечити потрібну якість ПЗ з врахуванням того, що деяка невідома кількість помилок та дефектів у ПЗ завжди залишається, а їх негативна дія повинна бути блокована або скорочена до допустимого рівня.

За наближеними оцінками витрати на розроблення ПЗ складають близько 275 мільярдів доларів, але лише 72% програмних проектів досягають етапу впровадження і всього 26% програмних проектів завершуються успіхом [4], тоді існує наступний розподіл витрат – рис.2.

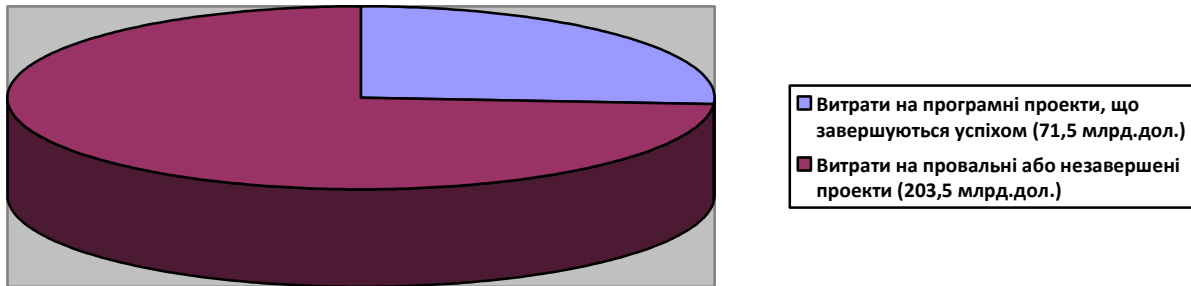


Рис.2. Розподіл витрат на програмні проекти

Програмні проекти часто зазнають невдач через неадекватне формулювання вимог, невдале проектування або неефективне планування, невірне розуміння або недостатній аналіз специфікації та проекту, тобто через помилки на ранніх етапах життєвого циклу ПЗ [3]. Таблиця 1 відображає відсоткову кількість помилок, які виникають на етапах формулювання вимог, проектування та реалізації ПЗ різного обсягу [3].

Таблиця 1. Розподіл помилок, припущених на різних етапах життєвого циклу

Етап ЖЦ	Обсяг ПЗ				
	2К	8К	32К	128К	512К
Формулювання вимог	До 10%	До 15%	До 20%	До 22%	До 23%
Проектування	До 15%	До 19%	До 25%	До 28%	До 32%
Конструювання	До 75%	До 66%	До 55%	До 50%	До 45%

З таблиці 1 видно, що помилки формулювання вимог та проектування складають 25-55% всіх помилок, причому чим більший обсяг ПЗ, тим більше помилок вноситься саме на ранніх етапах.

Отже, забезпечення можливості раннього виявлення помилок та гарантування якості ПЗ на етапі проектування дали б можливість уникнути ряду катастроф та інцидентів, причини яких були внесені на етапах формулювання вимог та проектування.

В основу процесу розроблення ПЗ покладено фундаментальну ідею: проектування ПЗ є формальним процесом, який можна вивчати і вдосконалювати. Завдяки засвоєнню і правильному застосуванню методів і засобів створення ПЗ можна підвищити його якість, забезпечити керованість його процесу проектування і збільшити термін його життя. Але при тому, що складність деяких програмних розробок на сьогодні вже перевищує складність багатьох машинобудівних або інфраструктурних проектів, а наслідки помилок є катастрофічними, процес розроблення ПЗ залишається не забезпеченим фундаментальною теорією та методологією надбання знань і навичок. Всі дослідження у галузі забезпечення якості на ранніх етапах життєвого циклу мають хаотичний, несистематизований характер, хоча, як доведено вище, саме на етапі проектування можна й варто виявляти та усувати до 55% всіх помилок майбутнього ПЗ. Безумовно, є ряд фундаментальних досліджень (роботи Боєма, Дейкстри, Мейєра), але відсутня завершена, протестована та апробована теорія та методологія розроблення якісного ПЗ, а також забезпечення якості ПЗ на ранніх етапах ЖЦ.

Існуючий стан галузі забезпечення якості програмних продуктів представлено на рис.3.

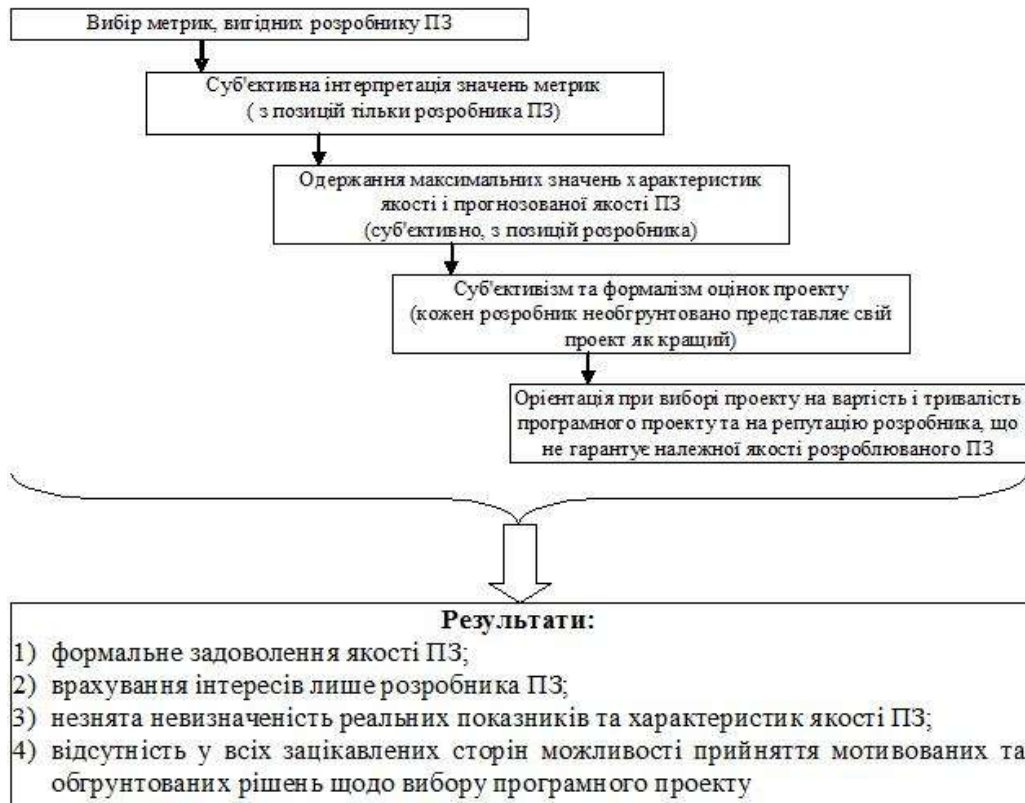


Рис.3. Сучасний стан галузі забезпечення якості ПЗ

Отже, наразі процес забезпечення якості ПЗ є недетермінованим, оскільки відсутні теорія та методологія в галузі оцінювання, прогнозування та забезпечення якості ПЗ, яка б при використанні однакових технологій розроблення із застосуванням однакових стандартів гарантувала створення однаково якісного ПЗ з відповідними витратами та в межах заданого часу. *Актуальною* є проблема створення методів і засобів досягнення необхідного рівня якості для проекту та розроблюваного ПЗ, а *перспективним напрямком досліджень* наразі є забезпечення галузі гарантування якості ПЗ фундаментальними теорією та методологією.

Використання backcasting-підходу для забезпечення якості програмних продуктів. Потіки етапу проектування програмного забезпечення представлені на рис.4.

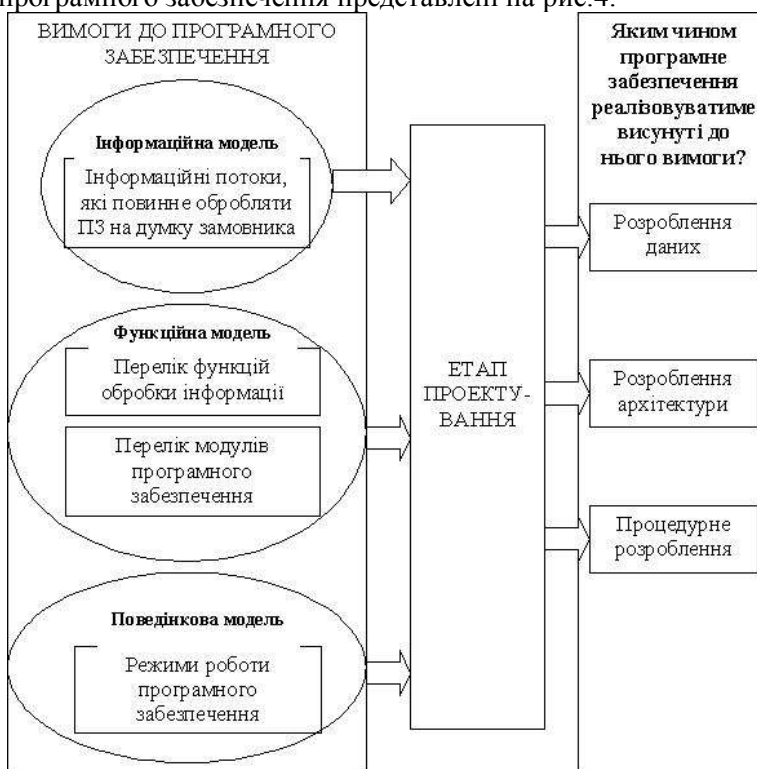


Рис.4. Потіки етапу проектування ПЗ

Аналіз потоків етапу проектування дає можливість визначити основні особливості етапу проектування: значна невизначеність, багатовимірність, велика кількість зацікавлених сторін з різними, часто конфліктними інтересами. При цьому на основі такої інформації необхідно отримати якісні, проблемно-орієнтовані, довготривалі, зважені та ґрунтовні рішення щодо забезпечення якості розроблюваного ПЗ, які враховують цілі та інтереси усіх зацікавлених осіб, а також дозволяють вносити зміни під час наступних етапів життєвого циклу ПЗ, адже великі програмні проекти можуть тривати й декілька років, а вимоги до ПЗ можуть змінюватись під час його проектування, розроблення і навіть супроводу. Одним з підходів, які дозволяють отримати якісні проблемно-орієнтовані рішення на основі невизначених багатовимірних даних із залученням значної кількості зацікавлених сторін, є backcasting-підхід [5, 6].

При використанні backcasting-підходу робота починається із визначення бажаного майбутнього, а потім працює в зворотньому шляху для визначення заходів, які необхідні для досягнення бажаного майбутнього. Основна ідея такого підходу - "якщо ми хочемо досягти певної мети, які дії для цього повинні бути виконані?". Backcasting – це спосіб, в якому передбачаються майбутні бажані умови, а його етапи визначаються з позиції досягнення цих умов. Backcasting-підхід часто є більш ефективним, ніж прогнозування, оскільки прогнозування спирається на відоме сьогодні, але ці знання завжди недосконалі та часто змінюються. Такий підхід наразі посідає центральне місце в стратегічному підході до планування стійкого розвитку та інновацій в багатьох галузях.

В контексті забезпечення якості ПЗ можна уявити велику кількість програмних проектів, призначених для вирішення однієї й тієї ж задачі. Наразі при виборі програмного проекту замовники та розробники орієнтуються на вартість і тривалість програмного проекту, а також на репутацію розробника, що абсолютно не гарантує належного рівня якості. Особливо складною проблема вибору є для проектів, вартість, тривалість та репутація розробника яких є приблизно однаковими. Backcasting-підхід здатен допомогти замовникам та розробникам при виборі програмного проекту

прийняти зважені та ґрунтовні рішення завдяки розвитку майбутнього бачення та врахуванню інтересів усіх зацікавлених сторін.

Backcasting-підхід для проблеми забезпечення якості розроблюваного ПЗ на етапі проектування представимо в наступному вигляді - рис.5.

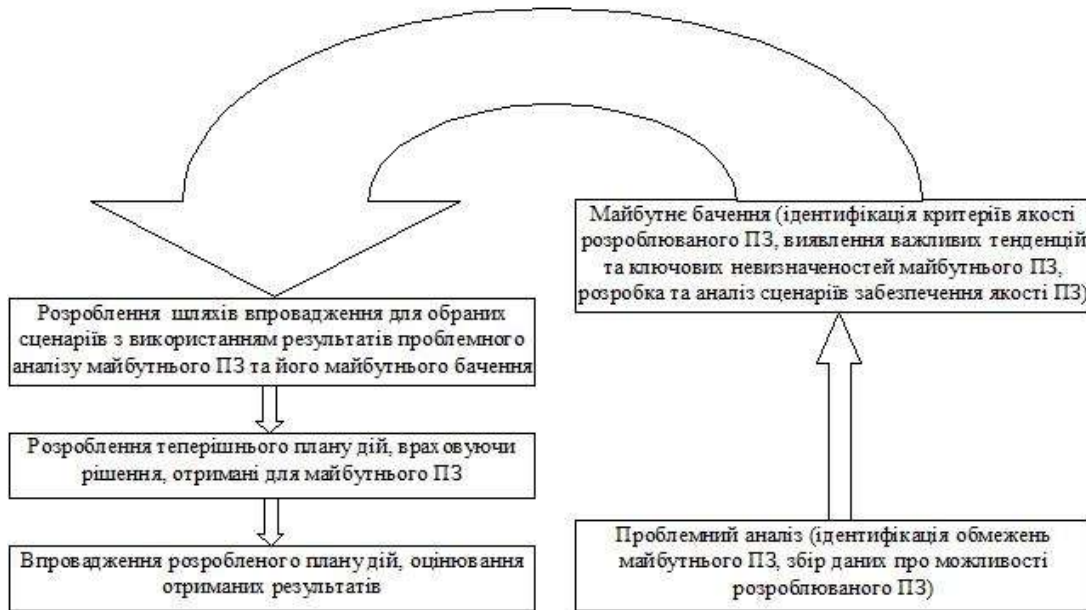


Рис.5. Backcasting-підхід для забезпечення якості ПЗ

Деталізовано етапи backcasting-підходу представлені на рис.6.

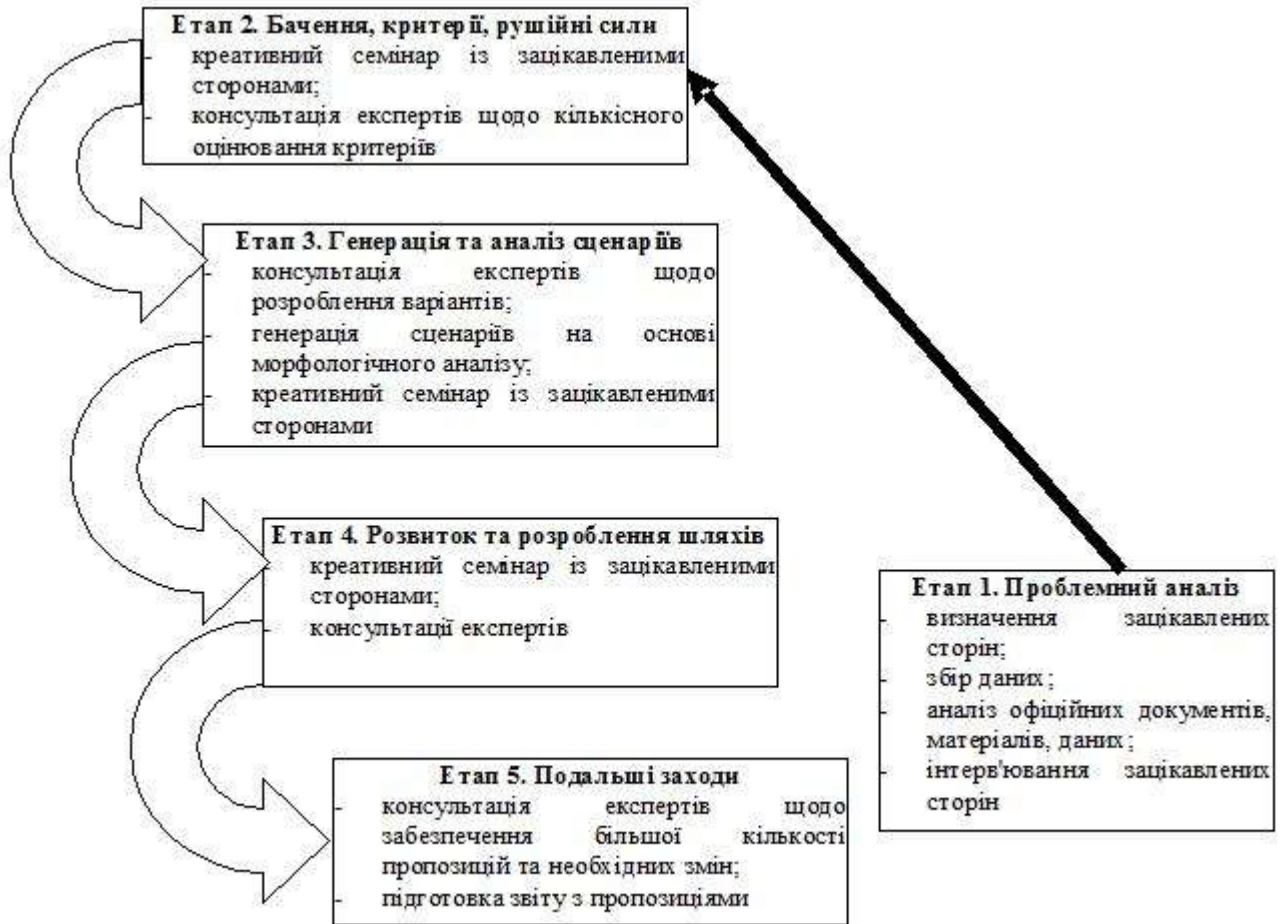


Рис.6. Етапи backcasting-підходу

Роль backcasting-підходу на етапі проектування для забезпечення якості розроблюваного ПЗ представлена на рис.7.

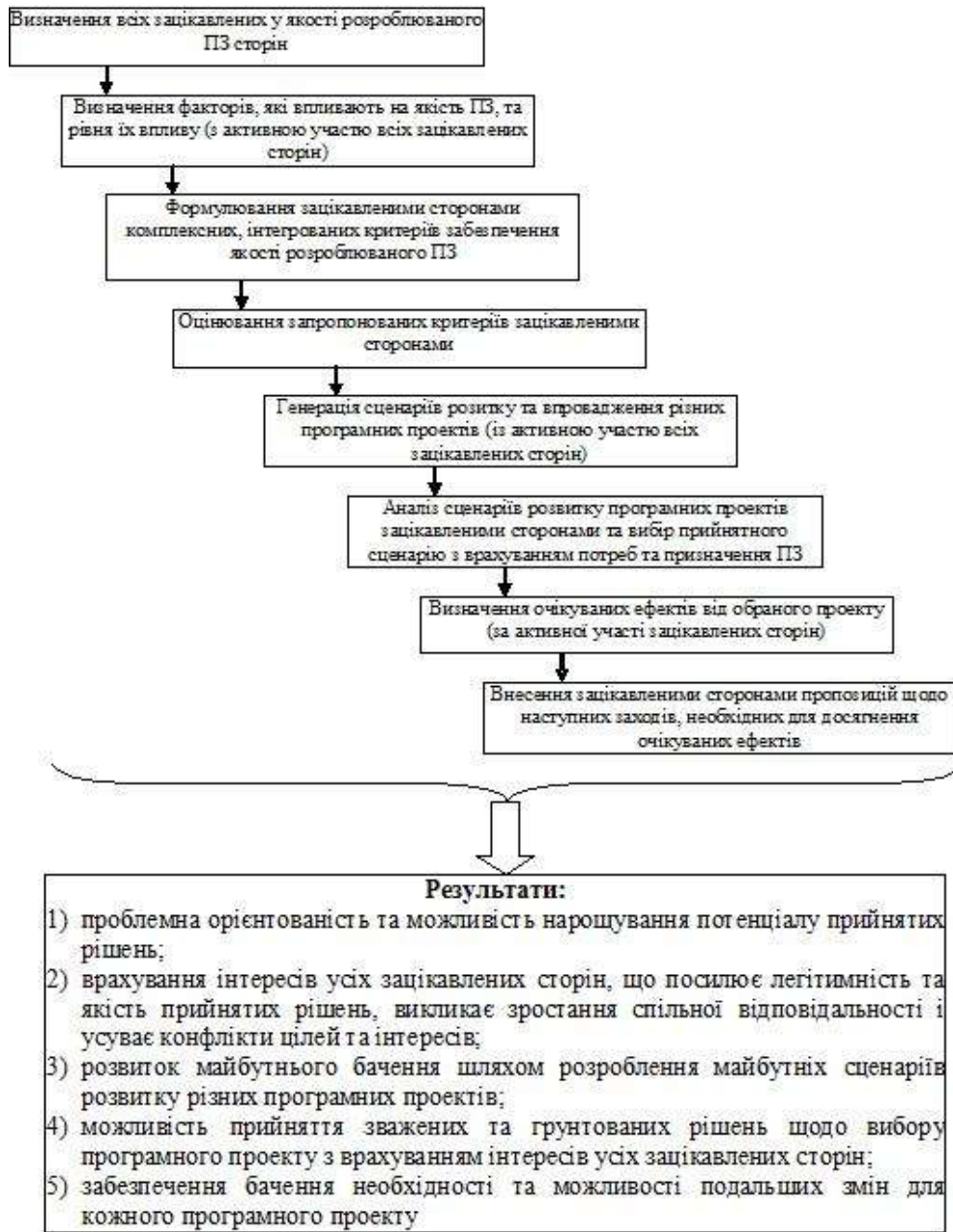


Рис.7. Роль backcasting-підходу для забезпечення якості ПЗ

Етап 1 - проблемний аналіз. Етап проблемного аналізу при забезпеченні якості ПЗ на етапі проектування було розпочато із визначення зацікавлених сторін. Зацікавлені сторони - фізичні та юридичні особи, які мають легітимний інтерес в діяльності організації, тобто залежать від неї або впливають на її діяльність, які можуть впливати на прийняття рішень або на яких можуть впливати прийняті рішення [5, 6]. Участь всіх зацікавлених сторін при прогнозуванні якості ПЗ є важливою, оскільки така спільна робота посилює легітимність прийнятих рішень, надає контекст-залежні рішення та знання, впливає на зростання якості одержаних результатів, а також викликає зростання спільної відповідальності за всі прийняті рішення. Узагальнено зацікавлені сторони, з точки зору проблеми досягнення необхідного рівня якості ПЗ, представлені на рис.8.

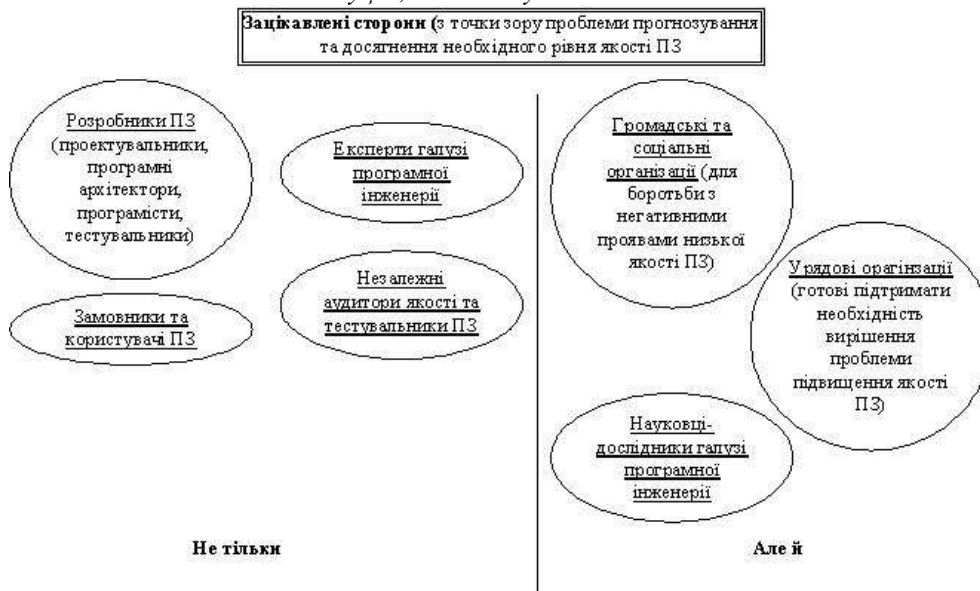


Рис.8. Зацікавлені сторони для проблеми забезпечення якості ПЗ

Далі будувалась сітка зацікавлених сторін, де всі зацікавлені сторони розподілялися згідно рівня своєї зацікавленості та рівня можливого впливу на вирішення проблеми (рис.9).

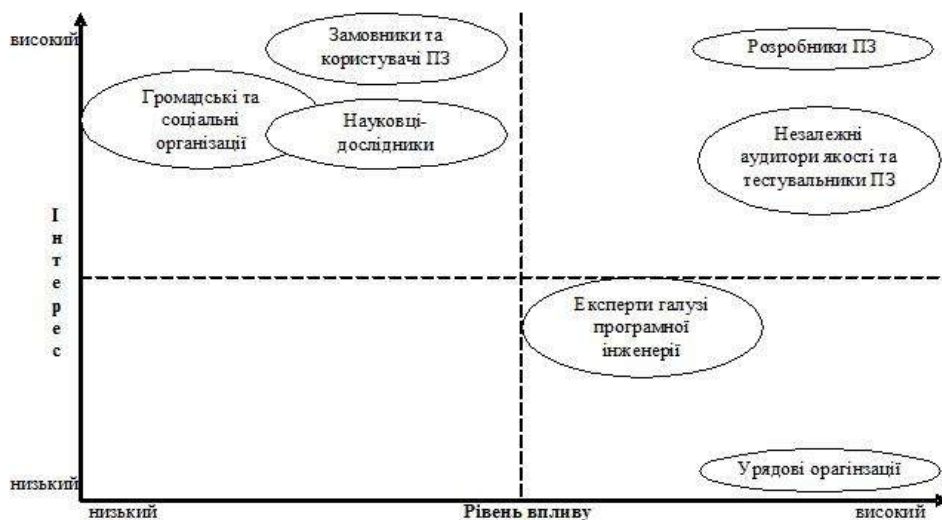


Рис.9. Сітка зацікавлених сторін для проблеми забезпечення якості ПЗ

Після цього зацікавленими сторонами визначались фактори, які впливають на проблему якості ПЗ. Для цього зацікавлені сторони проаналізували всі офіційні та неофіційні документи. Особливу увагу при цьому було приділено галузевим стандартам, тому що при розробленні ПЗ софтверні організації зобов'язані керуватись стандартами як щодо процесів розроблення, так і щодо процесів оцінювання та забезпечення якості. Крім стандартів, зацікавленими сторонами було розглянуто документацію організації-замовника, яка допомогла виявити особливості використання розроблюваного ПЗ, а також специфікацію вимог до ПЗ.

Після визначення множини впливових факторів зацікавлені сторони вказали рівень впливу кожного з факторів, а також розбили множину факторів на підмножини наступним чином:

1) економічні фактори - ефективність (значний рівень впливу), кількість виконавців (середній), розмір і вартість документації для користувачів (незначний), вартість помилки (значний), прогнозовані розмір і вартість розроблюваного ПЗ (значний);

2) технологічні фактори - застосовувані при проектуванні та розробленні ПЗ процеси та технології (значний рівень впливу), функційна придатність (значний), можливість переносу (середній), функційні вимоги (значний), нефункційні вимоги (значний);

3) соціальні фактори - зручність використання (значний), надійність (значний), розмір та склад цільової аудиторії (середній), рівень безпеки програмного забезпечення (значний);

4) організаційні фактори - середовище організації-розробника та зміни в ньому (середній), цілі розробника (середній), розмір та структура організації розробника (незначний), сумісність (середній), супроводжуваність (незначний), термін виконання (значний), залежності та суперечливості вимог (значний).

Етап 2 – формування майбутнього бачення, визначення критеріїв та рушійних сил.

Формування цілісного бачення щодо забезпечення належного рівня якості ПЗ було виконане зацікавленими сторонами після ретельного вивчення ними всіх фактів, які містяться в документації, архівних записах та статистичних вибірках, а також виявлялися під час ретельних спостережень, оглядів, інтерв'ю та опитувань.

Аналіз множини факторів, які впливають на проблему виявлення можливого рівня якості та досягнення необхідного рівня якості ПЗ, а також множини одержаних фактів дав можливість зацікавленим сторонам сформулювати основні критерії забезпечення якості ПЗ. При формуванні критеріїв кожна зацікавлена сторона висувала та захищала саме важливі для неї критерії. Наступним кроком стало визначення в даному випадку п'яти найважливіших інтегрованих критеріїв. З точки зору зацікавлених сторін для забезпечення якості ПЗ такими критеріями є:

1) безпека та надійність ПЗ - даний критерій відстоюють замовники та користувачі ПЗ, громадські та соціальні організації, незалежні аудитори якості та тестувальники, експерти галузі програмної інженерії, науковці-дослідники; 2) функційна придатність ПЗ - відстоюють розробники ПЗ, замовники та користувачі ПЗ, науковці-дослідники; 3) зручність використання ПЗ - відстоюють замовники та користувачі ПЗ; 4) економічна ефективність ПЗ - відстоюють розробники ПЗ, замовники та користувачі ПЗ, урядові організації; 5) сумісність ПЗ - відстоюють замовники та користувачі ПЗ.

Після формулювання основних критеріїв зацікавлені сторони виконали їх узгоджене оцінювання за 5-бальною шкалою, при якому кожна зацікавлена сторона відстоювала свою думку та свою оцінку і намагалась переконати інші сторони в правильності своїх міркувань. В жодному разі оцінка критерію не є середнім арифметичним оцінок даного критерію, наданих різними зацікавленими сторонами, - вона відображає досягнутий компроміс між зацікавленими сторонами.

Наступним кроком було розташування визначених комплексних критеріїв в сітці, яка відображає величину рушійної сили та рівень впливу даного критерію на проблему (рис.10).

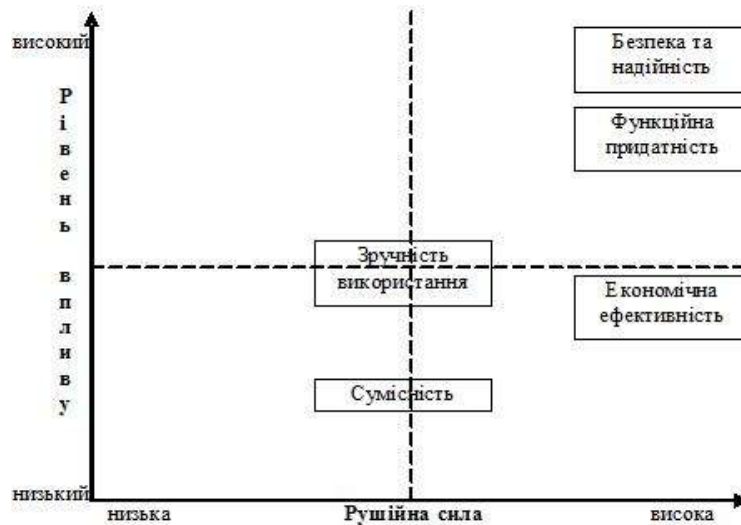


Рис.10. Сітка основних критеріїв для проблеми забезпечення якості ПЗ

Етап 3 – генерація та аналіз сценаріїв. На третьому етапі відбулась генерація сценаріїв розвитку та впровадження різних програмних проектів. Розробниками було запропоновано наступні програмні проекти та сценарії їх розвитку:

1) більша частина проектного часу та коштів (80%) витрачається на забезпечення можливості програми виконувати задані функції в заданих умовах роботи без збоїв та відмов, а решта часу та коштів - на естетичність, доступність, функційну повноту, можливість співіснування з іншим програмним забезпеченням на різних платформах, продуктивність та повернення інвестицій;

2) більша частина проектного часу та коштів витрачається на функційну повноту та доцільність (приблизно 50%) та на можливість вивчення, естетичність та доступність (приблизно 30%), а на забезпечення безперебійної роботи ПЗ, можливість співіснування з іншим програмним забезпеченням на різних платформах, продуктивність та окупність доводиться решта часу та коштів;

3) більша частина проектного часу та коштів доводиться на можливість співіснування та взаємодії з іншим програмним забезпеченням на різних платформах (40%) та продуктивність, окупність і повернення інвестицій (40%), а решта проектного часу та коштів витрачається на функційну повноту, естетичність, доступність і забезпечення безперебійної роботи ПЗ.

Для здійснення обґрунтованого та мотивованого вибору програмного проекту зацікавленими сторонами було виконано аналіз сценаріїв розвитку запропонованих проектів - всі зацікавлені сторони оцінили за п'ятибальною шкалою розроблені сценарії за кожним з обраних критеріїв, виходячи з потреб та призначення ПЗ, та знайшли середні оцінки. Після чого було пораховано сумарні бали для кожного сценарію. Так, наприклад, для критичного ПЗ сценарій (проект) №1 одержав 17 балів, сценарій (проект) №2 - 14 балів, сценарій (проект) №3 - 13 балів. Отже, для забезпечення необхідного рівня якості критично важливого ПЗ слід обирати сценарій (проект) №1.

Етапи 4 і 5 – розроблення шляхів та подальших заходів для обраного сценарію. На четвертому етапі зацікавлені сторони для обраного сценарію надавали відповіді на наступні запитання: 1) Які зміни (методологічні - М, структурні - С, технологічні - Т) необхідні?; 2) Які зацікавлені сторони необхідні для виконання даних змін?; 3) Терміни, доречні для запропонованих змін? - при цьому процес забезпечення якості ПЗ розглядався в трьох часових перспективах ЖЦ: етапи реалізації, тестування та супроводу. Відповіді на ці питання представлено на графіку - рис. 11.

↑ Т С М ↓	Підвищення зрілості технологічних процесів в софтверній організації <u>Зацікавлені сторони:</u> розробники ПЗ, незалежні аудитори, науковці, замовники та користувачі	Зосередженість на функціях самого ПЗ, які допомагають виявляти помилки <u>Зацікавлені сторони:</u> розробники ПЗ, замовники та користувачі, аудитори, науковці, експерти	Придільення значної уваги питанням супроводу після передачі ПЗ в експлуатацію <u>Зацікавлені сторони:</u> замовники та користувачі, громадські та урядові організації, експерти
	Забезпечення можливості кількісного вимірювання результатів етапу реалізації - аудит коду <u>Зацікавлені сторони:</u> замовники та користувачі, аудитори, експерти, науковці	Тестування ПЗ з врахуванням результатів аудиту коду та структурних особливостей ПЗ <u>Зацікавлені сторони:</u> науковці, замовники та користувачі, аудитори, експерти	Забезпечення задоволення вимог користувачів програмним забезпеченням протягом всієї експлуатації <u>Зацікавлені сторони:</u> замовники та користувачі, аудитори, експерти, науковці, громадські організації
	Розроблення методологій розроблення ПЗ з метою концентрування уваги на програмному коді та на детальних моделях ПЗ <u>Зацікавлені сторони:</u> науковці, експерти, громадські організації, урядові організації	Розроблення методологій тестування ПЗ на основі аналізу специфікації вимог та з орієнтацією на код і дефекти <u>Зацікавлені сторони:</u> науковці, експерти, аудитори, громадські організації, урядові організації	Розроблення методологій супроводу ПЗ з метою усунення збоїв та покращення показників продуктивності <u>Зацікавлені сторони:</u> науковці, експерти, аудитори, громадські організації, урядові організації
Етап проектування ↓	Етап реалізації	Етап тестування	Етап супроводу

Рис. 11. Необхідні методологічні, структурні та технологічні зміни для ПЗ

На п'ятому етапі зацікавленими сторонами було визначено очікувані ефекти від обраного сценарію. Так, для обраного сценарію №1 очікуваними ефектами зацікавлені сторони назвали: підвищення надійності та безпеки розроблюваного ПЗ; підвищення зрілості технологічних процесів в організації; забезпечення можливості кількісного вимірювання програмного коду; розроблення відповідних методологій для етапів ЖЦ, які можуть бути використані у наступних проєктах.

Після визначення очікуваних ефектів зацікавлені сторони сформулювали наступні заходи, необхідні для досягнення ефектів: 1) науковці, експерти та аудиторі вимагають стандартизувати метрики для аудиту програмного коду, а також вирішити проблему складності інтерпретації величин метрик; 2) користувачі вимагають розробити програмні засоби, які б були орієнтовані не тільки на розрахунок метрик на етапах проєктування та реалізації, але й на трактування одержаних величин метрик для більш ґрунтовного та мотивованого вибору програмного проєкту; 3) замовники ПЗ та урядові організації вимагають розробити методологію оцінювання якості проєкту та прогнозування якості розроблюваного за проєктом програмного забезпечення на етапі проєктування, яка надаватиме можливість раннього виявлення помилок, що забезпечить підвищення якості ПЗ та зменшення витрат на його розроблення; 4) розробники ПЗ вимагають розробити теорію та методологію галузі програмної інженерії, яка б при використанні однакових технологій розроблення із застосуванням однакових стандартів гарантувала створення однаково якісного ПЗ з відповідними витратами та в межах заданого часу; 5) громадські організації вимагають розробити фундаментальну теорію та методологію оцінювання і забезпечення (гарантування) якості ПЗ, яка дозволить зменшити кількість та вплив помилок і відмов ПЗ, що загрожують катастрофами, людськими жертвами, екологічними катаклізмами, значними економічними та часовими втратами.

Висновки. Автором доведено можливість застосування backcasting-підходу на етапі проєктування для забезпечення необхідного (з точки зору призначення ПЗ) рівня якості ПЗ.

Перевагами використання такого підходу при забезпеченні якості ПЗ є:

- 1) проблемна орієнтованість підходу;
- 2) можливість розвитку майбутнього бачення, розроблення майбутніх альтернатив та сценаріїв;
- 3) активна участь та врахування інтересів усіх зацікавлених сторін на всіх етапах підходу (при бажанні зацікавлених сторін активно співпрацювати та йти на компроміс), що посилює легітимність прийнятих рішень, впливає на зростання якості одержаних результатів, а також викликає зростання спільної відповідальності та дозволяє усунути конфлікти цілей та інтересів;
- 4) можливість прийняття зважених та ґрунтовних рішень щодо вибору програмних проєктів;
- 5) можливість бачення необхідності та здійсненності подальших змін для різних проєктів.

Подальшими перспективами використання backcasting- підходу для забезпечення якості ПЗ є:

- 1) розроблення математичного підґрунтя для використання даного підходу в галузі;
- 2) розроблення фундаментальної теорії та методології забезпечення якості ПЗ;
- 3) забезпечення необхідного ступеня довіри до даного підходу та одержаних рішень.

Література

1. Сборник действующих международных стандартов ИСО серии 9000. Т.1, 2, 3. - М.: ВНИИКИ, 1998.
2. Липаев В.В. Программная инженерия. Методологические основы. - М.: ТЕИС, 2006. - 608 с.
3. Макконнелл С. Совершенный код - М.: Издательство "Русская редакция", 2013 - 896 с.
4. Мищенко В.О., Поморова О.В., Говорущенко Т.А. CASE-оценка критических программных систем. В 3-х томах. Том 1. Качество / Под ред. Харченко В.С. - Харьков: НАУ "ХАИ", 2012. - 201 с.
5. Ebert J. E. J., Gilbert D. T., Wilson T. D. Forecasting and back casting: Predicting the impact of events on the future // Journal of Consumer Research, 2009 - Volume: 36, Issue: 3, pp.353-366
6. Höjer M., Mattsson L-G. Determinism and backcasting in future studies // Futures, 2010 - Volume 32, pp. 613-634

УДК 004.05

С.В. Гринюк, К.Е.Топчевська, П.С. Шолом
Луцький Національний Технічний Університет

КОДУВАННЯ ІНФОРМАЦІЇ ЗА ДОПОМОГОЮ САМОКОРЕГУЮЧОГО ШТРИХОВОГО КОДУ НЕМ-8.

С.В. Гринюк, К.Е.Топчевська, П.С. Шолом. Кодування інформації за допомогою самокорегуючого штрихового коду НЕМ-8. В статті розглядається кодування інформації за допомогою штрих-кодів. Також перераховані основні принципи побудови саморегулюючого штрихового коду НЕМ-8. Розглянуто задачу по максимізації надійності штрихового коду та створено ефективний код, здатний запобігати неправильному зчитуванню закодованої інформації.

Ключові слова: штриховий код, інформація, код, повідомлення, алфавіт, алгоритм Хеммінга.
Форм.3. Табл.5. Рис.3. Літ.7.

С.В. Гринюк, К.Э.Топчевская, П.С. Шолом. Кодирования информации с помощью самокорректирующее штрихового кода НЕМ-8. В статье рассматривается кодирования информации с помощью штрих-кодов. Также перечислены основные принципы построения саморегулюющего штрихового кода НЕМ-8. Рассмотрена задача по максимизации надежности штрихового кода и создан эффективный код, способен предотвращать неправильном считыванию закодированной информации.

Ключевые слова: штриховой код, информация, код, сообщение, алфавит, алгоритм Хемминга.
Форм.3. Табл.5. Рис.3. Лит.7.

S.V. Grynyuk, K.E.Topchevska, P.S. Sholom. Encoding information using samokorehuyuchoho barcode NEM-8.. In the article the encoding information using bar codes. There are the main principles of the samoruhulyuyuchoho barcode Nam-8. The problem to maximize reliability barcode and create efficient code, is able to prevent the wrong read-encoded information.

Keywords: barcode, information, code, message, alphabet, Hamming algorithm.
Form.3. Tab.5. Fig.3. Lit.7.

Вступ. У роботі проведено дослідження про кодування інформації за допомогою штрих-кодів. Сфера застосування штрихових кодів надзвичайно широка і вона весь час розширюється, але не зважаючи на це для більшості пересічних громадян ці чорні та білі смужки залишаються незрозумілими. Широке використання штрихових кодів було зумовлене необхідністю забезпечити автоматизоване введення інформації в комп'ютерні системи управління, що відрізнялося б високою надійністю, простотою і економічністю.

Штриховий код — це не щось особливе, існуюче саме по собі, а передусім елемент системи управління. В відриві від комп'ютерної системи управління, поза зв'язком з її інформаційною базою він не має жодного сенсу. Технологія штрихового кодування застосовується в багатьох сферах людської діяльності, але найбільш широко і ефективно вона використовується в оптовій і роздрібній торгівлі, управлінні матеріальними запасами, управлінні перевезеннями. Штрихові коди характеризуються високою надійністю.

Розглянемо основні принципи та правила, що використовуються при створенні штрихових кодів і які є обов'язковими для будь-якого їх типу. Одразу потрібно зазначити, що інформація яку ми кодуємо представлена в двійковому виді, тобто кодується двома значеннями: '0' та '1'. В штриховому кодуванні існує два способи задання цих значень, першим є спосіб, коли значення '0' та '1' кодуються відповідно двома кольорами - білим та чорним.

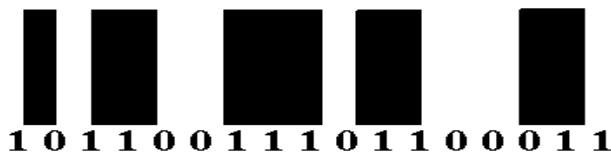


Рис. 1. Штрих-код

В цьому способі штрихи що відповідають '0' та '1' мають однакову ширину. В разі якщо в бінарній послідовності йдуть одне за одним кілька однакових n значень '0' чи '1' їм буде відповідати білий чи чорний штрих n-кратної ширини.

©.В. Гринюк, К.Е.Топчевська, П.С. Шолом

Другим способом представлення бітової послідовності в вигляді штрихового коду є спосіб коли '0' та '1' задані не різними кольорами, а різними значеннями ширини штрихів. Тобто маємо чотири атомарні графічні символи два вузькі штриха та два широкі білого та чорного кольорів. В такому штриховому коді білі та чорні штрихи весь час йдуть почергово, а значенням '0' та '1' відповідають відповідно широкі та вузькі штрихи. В цьому разі наведена вище бінарна послідовність буде мати вигляд:



Рис. 2. Бітова послідовність в вигляді штрихового коду

В кожному з цих варіантів є як переваги так і недоліки. Так в першому варіанті штриховий код буде коротшим в наслідок того, що всі біти кодуються однаковими по ширині штрихами. По цій самій причині в першому варіанті штриховий код бінарної послідовності зі сталим числом бітів буде мати сталий розмір, в той час як в другому варіанті розмір штрихового коду буде залежати від співвідношення нулів та одиниць. Але недоліком першого варіанту є те, що при великій кількості йдучих один за одним однакових бітів їх графічне представлення може неправильно тлумачитися. Так, наприклад буде важко розрізнити штрихкоди для 100001 та для 1000001.

Розглянемо інші особливості побудови штрихових кодів, які також використовуються для класифікації штрихових кодів. Однією з таких особливостей є наявність чи відсутність контрольних штрихів(бітів). Вони використовуються в разі потреби стабілізації швидкості зчитування нашого коду від початку до кінця. В випадку відсутності контрольних штрихів, при нерівномірній швидкості зчитування штрих-коду, цей код можливо буде інтерпретовано неправильно. Щоб цьому запобігти, на початку та в кінці нашого коду розміщується набір з принаймні двох контрольних штрихів. Після зчитування ЕОМ цього коду, обчислювальна машина може судити про зміну швидкості сканування штрихового коду і відповідно корегувати процес декодування. Прикладом застосування контрольних штрихів може бути штриховий код типу EAN-13. В ньому контрольні штрихи наявні не тільки на початку та в кінці, а і в середині коду.

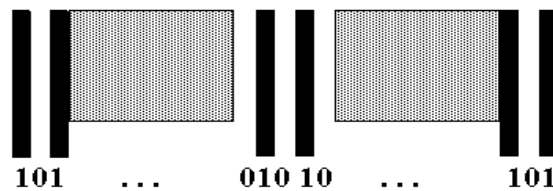


Рис. 3. Штриховий код типу EAN-13

Ще однією особливістю при побудові штрихового коду є наявність чи відсутність контрольної суми. Для гарантування правильності декодування штрихового коду в деяких типах штрихового коду до інформації, що кодється додається деяка контрольна сума яка функціонально залежить від кодової інформації. Ця контрольна сума кодується в штриховий код разом з основною інформацією, а при декодуванні ЕОМ знову вираховує контрольну суму цього коду і порівнює з заданим. Зрозуміло що в разі неспівпадання цих двох контрольних сум штриховий код був не правильно зіскановано.

Кожне повідомлення, кожна інформація про той або інший факті має немов би дві сторони: конкретний зміст даного повідомлення, даного факту, і статистичні (ймовірносні) властивості, що дозволяють порівнювати цілком разнорідні повідомлення по тій різноманітності станів, з якими ці повідомлення зв'язані.

Аналіз останніх досліджень і публікацій. Задача, яка буде описана тут, була сформульована під час введення на деякому великому підприємстві комп'ютеризованої системи пропусків [1]. Кожен ©.В. Гринюк, К.Е.Топчевська, П.С. Шолом

працівник мав свій 6-ти значний ідентифікаційний код, котрий разом з фотографією та іменем працівника був нанесений на пластикову картку-перепустку. Але для автоматизації контролю за переміщенням працівників необхідно було забезпечити можливість зчитувати та аналізувати інформацію з цих карток-перепусток і для ЕОМ. Таку можливість можна забезпечити наявністю на перепустці магнітної стрічки або ж штрихового коду. Магнітна стрічка в даній ситуації не є оптимальним вибором, в зв'язку з більшою вартістю як виготовлення такої картки так і її обслуговування. Головною перевагою магнітної картки є можливість збереження більшої інформації про власника, ніж при застосуванні штрихового коду, але в нашому випадку такої потреби немає - достатньо зберігати інформацію про ідентифікаційний код робітника, та можливо ще яку-небудь (рівень доступу, час роботи, і т.п.) [5].

Постановка проблеми. Таким чином оптимальним рішенням цієї задачі є застосування штрихового коду, який здатний кодувати 6-8 числових символів. Але використання якого-небудь із загальноживаних типів штрихових кодів виявилось ускладненим специфікою використання карток-перепусток - через необхідність весь час носити їх на одязі, картки поступово забруднювалися, на їх поверхні з'являлися подряпини. Все це не впливало на можливість читання інформації людиною, але при використанні сканерів штрихових кодів кількість помилок з часом дуже швидко збільшувалась, виникала потреба в заміні картки-перепустки.

Всі ці проблеми в кінці-кінців привели до формулювання слідуєчої постановки задачі: необхідно створити новий тип штрихового коду, основними вимогами до якого є простота, не вибагливість до скануючого пристрою, можливість кодування 6-8 числових символів та висока стійкість до помилок.

Самокорегуючий штриховий код НЕМ-8. Алфавітом штрих-коду назвемо множину символів з яких складається інформація закодована в цьому штриховому коді.

Бітовий ряд - це послідовність скінченого числа "0" та "1" (в графічному представленні - білих та чорних штрихів), яка задає один символ алфавіту штрихкоду.

Основні вимоги до штрихового коду:

Вимога 1. Мінімальна кількість символів, яку повинен містити штрихкод - 6, але можливо це число варто збільшити до 7 або 8 символів вводячи резервне або контрольне число.

Вимога 2. Алфавіт нашого штрихового коду складається тільки з 10 цифр 0,1, ...,9. Одже на кодування одного символу нам знадобиться $\min 4$ біти $2^4=16$ ($2^3<10$). Насправді ця кількість в нас буде дещо більшою, за рахунок введення контрольних бітів та накладення деяких умов. Однією з таких умов буде наступна.

Вимога 3. Для запобігання злиття двох сусідніх бітових рядів, необхідно щоб кожен з таких рядів розпочинався з 0 та закінчувався 1-цею.

Вимога 4. Велика кількість підряд йдучих однакових символів збільшує імовірність помилки. Тому слід накласти ще одну умову: в бітовому ряді не повинно бути підряд біль ніж 3 нулів чи одиничок.

Вимога 5. Як зазначалося в вимозі 2, в наслідок виимог 3-4, на кодування одного символу потрібно 6 біт.

Наш штриховий код буде забезпечено властивістю самовиправлення помилок, яка буде базуватися на методі Хеммінга. Для цього нам необхідно вияснити яку кількість бітів ми відведемо під перевірочні. Виходячи з відомої формули $2^k > n+1$, де n - загальна кількість бітів, k - кількість провірочних бітів, $m=n-k$ - кількість інформаційних бітів, отримуємо наступну таблицю:

Таблиця 1. Кількість інформаційних бітів

k	n	m=k-n
1	1	0
2	3	1
3	7	4
4	15	11
5	31	26
6	63	57
7	127	120
8	255	247

Мінімальне число бітів в бітовому ряді 6, якщо під інформаційні ми відведемо 26 біт, то ми зможемо кодувати $26/6=4$ символи. Така кількість є недостатньою для нашої задачі, тому виділимо під інформаційні 57 бітів, так ми зможемо кодувати до $57/6=9$ символів, що є навіть надлишково для нас.

В наступній таблиці записано всі можливі 6-ти значні бітові ряди, що задовольняють нашим умовам.

Таблиця 2.

1	000101
2	000111
3	001001
4	001011
5	001101
6	010001
7	010011
8	010101
9	010111
10	011001
11	011011
12	011101

Цієї кількості бітових рядів достатньо, щоб кожному символі нашого алфавіту штрихового коду поставити у відповідність один з них.

В разі виникнення однієї помилки в нашому штриховому коді, вона буде виправлена за допомогою алгоритму Хеммінга. В разі виникнення двох помилок метод Хеммінга перестає бути ефективним, в такій ситуації принаймні повинно бути виявлено наявність помилки. Ми забезпечимо перевірку на наявність подвійних помилок, ввівши восьмий контрольний символ. Він функціонально залежить від інших символів, і ця залежність вибирається так щоб при зміні одного з перших семи символів, восьмий контрольний також обов'язково змінився. Якщо при виникненні двох помилок і вони знаходяться в одному бітовому ряді, то наявність такої помилки буде виявлено контрольним символом, але якщо ці помилки з'являться в різних бітових рядах, тобто будуть змінені два числа в нашому коді, то можливі такі комбінації символів коли помилка не буде виявлена взагалі. Для запобігання таких помилок набір бітових рядів потрібно підібрати так, щоб матриця кодових відстаней цього набору не містила одиниць. Тоді при виникненні двох помилок в різних бітових рядах, ці новоутворені бітові ряди не будуть належати нашому наборові, і таким чином буде виявлено наявність помилок.

Але тут виникає проблема: для того щоб вивести з матриці кодових відстаней одиниці, потрібно скоротити набір бінарних рядів, але тоді цього набору буде замало щоб поставити один бітовий ряд у відповідність одному символу нашого алфавіту. Ми приходимо до потреби збільшення довжини наших бітових рядів з 6-ти до 7-ми символів.

$57/7=8$. Це також задовольняє нашим вимогам. Ми зможемо кодувати до восьми символів. В наступній таблиці записано всі можливі бітові ряди, що задовольняють нашим умовам. Нагадаємо, що алфавіт нашого штрихового коду складається з 10 символів, а оскільки допустимих бінарних рядів 22 то в нас тепер є змога розділивши їх на дві групи, кодувати один з наших символів не бітовим рядом, а належністю бітових рядів інших символів до тієї чи іншої групи.

Таблиця 3.

1	0001001		12	0100101
2	0001011		13	0100111
3	0001101		14	0101001
4	0010001		15	0101011
5	0010011		16	0101101
6	0010101		17	0110001
7	0010111		18	0110011
8	0011001		19	0110101
9	0011011		20	0110111
10	0011101		21	0111001
11	0100011		22	0111011

За рахунок цього можна або додати до нашого коду дев'ятий символ без збільшення кількості бітів в штриховому коді(тим самим зменшивши надлишковість нашого коду), або, залишивши наш код 8-ми символьним, зменшити кількість біт, що будуть кодувати цей код (тим самим також зменшивши надлишковість).

Але в нашій задачі основною вимогою до штрихового коду є не мінімізація надлишковості, а збільшення стійкості коду до помилок. Тому нам доведеться відмовитися від запропонованого вище варіанту розподілу бітових послідовностей на дві групи, з метою запобігання тих самих помилок, що були описані вище.

Виберемо з 22-х бітових рядів (таб. 3) десять таких, щоб їх матриця кодових відстаней не містила одиниць.

Таблиця 4.

0	0001001
1	0010001
2	0010111
3	0011011
4	0011101
5	0100111
6	0101011
7	0101101
8	0110011
9	0110101

Тоді матриця кодових відстаней буде наступна :

Таблиця 5. Матриця кодових відстаней

	0	1	2	3	4	5	6	7	8	9
0	0	2	4	2	2	4	2	2	4	4
1	2	0	2	2	2	4	4	4	2	2
2	4	2	0	2	2	2	4	4	2	2
3	2	2	2	0	2	4	2	4	2	4
4	2	2	2	2	0	4	4	2	4	2
5	4	4	2	4	4	0	2	2	2	2
6	2	4	4	2	4	2	0	2	2	4
7	2	4	4	4	2	2	2	0	4	2
8	4	2	2	2	4	2	2	4	0	2
9	4	2	2	4	2	2	4	2	2	0

Наш штриховий код буде мати наступну структуру:

0xxxxx10xxxxx10xxxxx10xxxxx10xxxxx10xxxxx10xxxxx10xxxxx1nnnnnn

Вона містить один невеликий недолік, останні шість бітів можуть неправильно зчитуватися. Наприклад: а) вище ми накладали обмеження на кількість однакових підряд йдучих символів "0" чи "1", на комбінацію pppppp в данному випадку таке обмеження не поширюється; б) в разі, якщо останній байт чи байти будуть рівні "1", то при графічному представленні цього коду вони будуть втрачені, оскільки зіллються з "тихою зоною". Щоб запобігти цим небажанним явищам, введемо додаткові нульові біти в кінець нашого коду. Отримаємо код наступної структури:

0xxxxx10xxxxx10xxxxx10xxxxx10xxxxx10xxxxx10xxxxx101nn01nn01nn010

В цьому коді всі вище перераховані недоліки усунуто.

Потрібно також для нашого коду вибрати алгоритм знаходження контрольного числа. Запозичимо алгоритм з коду типу EAN. Тоді вагами будуть "3 1 3 1 3 1 3", а контрольне число буде рівне mod10 від зваженої суми.

Таким чином створивши штриховий код(назвемо його НЕМ-8), було розв'язано поставлену перед нами задачу. Нам залишилося тільки перевірити ефективність цього коду. Для початку підрахуємо ентропію нашого коду. Не важко пересвідчитися що НЕМ-8 складається з 71 біта, кількість комбінацій чисел які можна утворити 10 мільйонів (10^7), оскільки восьме число залежить тільки від перших семи, ми його не враховуємо. Кількість інформації що несе в собі штрихкод НЕМ-8 рівна $I(a)=\log(10^7)$. Вище була наведена формула для знаходження величини надлишковості коду, використаємо її тепер.

$$r = 1 - \frac{I(a)}{n \cdot \log L} \quad (1)$$

в нашому випадку алфавіт двозначний $L=2$, кількість бітів $n=71$ тому

$$r = 1 - \frac{\log 10^7}{71 \cdot \log 2} = 0.67 \quad (2)$$

Як бачимо коефіцієнт надлишковості досить великий, для порівняння надлишковість коду EAN-13 рівна

$$r = 1 - \frac{\log 10^{12}}{95 \cdot \log 2} = 0.58 \quad (3)$$

що також не мало. Але завдяки цій надлишковості ці коди стають більш стійкими до помилок, що для нас більш важливо ніж надмірність інформації. Стійкість до помилок нам ще необхідно буде перевірити. Для цього необхідно провести статистичні дослідження по виправленню та виявленню помилок. Для цих досліджень необхідно створити прикладну програму, яка б генерувала для заданого коду певну кількість помилок і намагалася б їх виявити та виправити.

1. Жураковський Ю.П., Полторак В.П. Теорія інформації та кодування: Підручник. — К.: Вища школа, 2011. — 255 с.
2. Берлекемп Э. Алгебраическая теория кодирования. — М., Мир, 1971 — 480 с.
3. Добрушина Р.Л., Лупанова О.Б. -М.: Издательство иностранной литературы, 1963. 830 с.
4. Котов П.А. Повышение достоверности передачи цифровой информации. -М.: Связь, 1966. 184 с.
5. Словарь по кибернетике / Справочное издание под редакцией Михалеви́ча В.С. -К.: Главная редакция Украинской Советской Энциклопедии имени М.П.Бажана, 1989. 752 с.
6. Шеннон К. Работы по теории информации и кибернетике / Пер. с англ. под ред. Котов П.А. Повышение достоверности передачи цифровой информации. -М.: Связь, 1966. 184 с.
7. Шульгин В.И. Основы теории передачи информации, Харьков, 2002. — 160 с.

УДК 681.322

Г. М. Губаль

Луцький національний технічний університет

БІБЛІОГРАФІЧНІ ІНСТРУМЕНТИ ТА ДЕЯКІ ПРАВИЛА ДЛЯ СТВОРЕННЯ МАТЕМАТИЧНИХ ТЕКСТІВ У ВИДАВНИЧІЙ СИСТЕМІ L^AT_EX

Губаль Г. М. Бібліографічні інструменти та деякі правила для створення математичних текстів у видавничій системі L^AT_EX. У статті розглядаються такі бібліографічні інструменти для створення математичних текстів у видавничій системі L^AT_EX, як BIBT_EX і CrossT_EX. Досліджуються три основні і проміжні стильові пакети бібліографічного інструмента BIBT_EX і наводяться приклади для математичних текстів. Обґрунтовується важливість використання бібліографічного інструмента CrossT_EX. Наводяться деякі правила створення математичних формул.

Ключові слова: видавнича система L^AT_EX, бібліографічний інструмент BIBT_EX, бібліографічний інструмент CrossT_EX, стильовий пакет, \vphantom.

Лит. 17.

Губаль Г. Н. Библиографические инструменты и некоторые правила для создания математических текстов в издательской системе L^AT_EX. В статье рассматриваются такие библиографические инструменты для создания математических текстов в издательской системе L^AT_EX, как BIBT_EX и CrossT_EX. Исследуются три основные и промежуточные стилевые пакеты библиографического инструмента BIBT_EX и приводятся примеры для математических текстов. Обосновывается важность использования библиографического инструмента CrossT_EX. Приводятся некоторые правила создания математических формул.

Ключевые слова: издательская система L^AT_EX, библиографический инструмент BIBT_EX, библиографический инструмент CrossT_EX, стилевой пакет, \vphantom.

Лит. 17.

Hubal H. M. Bibliographic instruments and some rules for creation of the mathematical texts in the publishing system L^AT_EX. In the article the bibliographic instruments for creation of mathematical texts in the publishing system L^AT_EX such as BIBT_EX and CrossT_EX are considered. The paper investigates three main and intermediate style packages of the bibliographic instrument BIBT_EX and makes examples for mathematical texts. Importance of the use of the bibliographic instrument CrossT_EX is justified. The paper makes some rules for creation of the mathematical formulae.

Keywords: publishing system L^AT_EX, bibliographic instrument BIBT_EX, bibliographic instrument CrossT_EX, style package, \vphantom.

Bibl. 17.

З 1985 року ВівT_EX є основним інструментом для створення бібліографії у видавничій системі L^AT_EX [1-8, 10-12, 14], внаслідок тісної інтеграції з цією системою і простоти у використанні.

Розглянемо основні пакети бібліографічних стилів інструмента ВівT_EX, які є актуальними для створення математичних текстів. Стиль у термінах ВівT_EX визначається bst-файлом [9, 13, 15-17].

Основними пакетами бібліографічних стилів є footnote (виноска), brackets (квадратні дужки) і author-year (автор-рік).

Бібліографічний стиль footnote використовується не стільки для математики, скільки для її історії. Тому не будемо зупинятись на розгляді цього стилю.

Бібліографічний стиль brackets найчастіше застосовується користувачами L^AT_EX для створення математичних текстів. Ефективність використання стилю brackets впливає з наступних фактів. Вираз у квадратних дужках розуміють як вступний коментар, початок речення або як пояснення. Наприклад,

‘див. [1]’, ‘[2] – це посилання на ...’ або ‘... це було доведено в [3]’.

Квадратні дужки можна поєднувати з круглими, причому форма посилання не залежить від контексту. Наприклад, речення:

‘Доведення цієї теореми проведено у книзі [1], де ...’

можна замінити реченням:

‘Зрозуміло (доведення цієї теореми проведено у книзі [1]), що ...’.

Стиль brackets має одну команду: \cite.

Якщо в доведенні теореми здійснюється посилання на раніше доведену теорему, то автор і дата публікації статті, яка містить раніше доведену теорему не мають важливого значення,

оскільки інформація про першоджерело міститься у бібліографічному списку. Проте в деяких випадках не прослідковується послідовність досліджень. Тому для здійснення посилань є недостатнім використання бібліографічного стилю brackets. У цьому випадку використовують бібліографічний стиль author-year, який дає важливу інформацію (автор і рік) у посиланні:

Замість '[1]' пишемо '(Боголюбов, 1946)'

Бібліографічний стиль author-year використовує круглі дужки замість квадратних.

Вибір бібліографічного стилю залежить від користувача видавничої системи L^AT_EX.

Пакети видавничої системи L^AT_EX, які підтримують стиль author-year, містять численну кількість команд. Розглянемо пакет natbib, який пропонує різноманітні налаштування. Цей пакет забезпечує конвертування до міток з квадратними дужками, присвоєння "псевдонімів", контроль пунктуації і є сумісним з пакетом hyperref. Замість команди \cite цей пакет використовує команду \citep для посилань у круглих дужках і команду \citet для посилань в тексті. Обидві команди мають два аргументи для записів у дужках.

Приклад використання пакета natbib наведений у табл. 1.

(\citep[зокрема][])	Однією з основних задач математичної фізики є побудова рівноважних і нерівноважних станів нескінченних систем частинок. Задача побудови рівноважних станів була зведена російськими науковцями (зокрема М.М. Боголюбовим; Б.І. Хацетом 1949) до задачі функціонального аналізу про існування розв'язку системи рівнянь для функцій розподілу і повністю розв'язана в роботах М.М. Боголюбова, Д.Я. Петрини, Б.І. Хацета і Д. Рюеля (1969, 1971).
\citeyearpar	Стани нескінченних нерівноважних систем частинок описуються нескінченними послідовностями функцій розподілу, визначених на фазовому просторі системи, які задовольняють нескінченній системі інтегро-диференціальних рівнянь, відомих як ланцюжок рівнянь Боголюбова (БГКІ ієрархія) (Боголюбов 1946, Петрина, Герасименко, Малишев 2002).
\citep	БГКІ ієрархія була виведена в роботах Боголюбова (1946), Борна, Гріна (1949), Кірквуда (1946), Івона (1935) для систем частинок із гладким потенціалом взаємодії.
\citet	Рівняння Боголюбова як абстрактне еволюційне рівняння розглядаються з 1972 року (Д.Я. Петрина).
(\citeauthor...)	

Табл.1. Приклад використання пакета natbib

Зауважимо, що існують проміжні пакети бібліографічних стилів, зокрема cite, footbib і alpha. Пакет cite генерує мітки з квадратними дужками у вигляді верхніх індексів, схожих на виноски. Пакет footbib генерує мітки у вигляді верхніх індексів, які вказують на виноски. Пакет alpha встановлює зв'язок між мітками типу '[1]' і '(Боголюбов, 1946)', внаслідок чого одержимо '[Боголюбов, 1946]'.

Для налаштування стилів у базі даних і посиланнях використовують бібліографічний інструмент CrossT_EX, в основі якого лежить об'єктно-орієнтована модель. CrossT_EX здійснює посилання не тільки присвоєнням довільним чином створених ключів, пов'язаних з кожним записом у базі даних, але і за допомогою семантичної інформації, однозначно ідентифікуючи об'єкт, що дає можливість уникнути неспівпадання в бібліографії. CrossT_EX підтримує нові типи об'єктів: url, покоління html-документів. CrossT_EX дає можливість писати стильові файли сучасними мовами програмування. Зауважимо, що CrossT_EX обернено сумісний з базами даних ВівT_EX, будучи побудованим на основі ВівT_EX.

CrossT_EX дає можливість вводити записи у бази даних не повністю, як у ВівT_EX, а окремо один від одного. За допомогою CrossT_EX створюють url кожного запису у базі даних. ВівT_EX достатньо автоматизований для форматування в базі даних. Однак, CrossT_EX дає можливість, наприклад, автоматично створювати аббревіатуру назв журналів, конференцій у записах, а також контролює наявність великих літер у назвах.

Розглянемо деякі правила написання математичних формул. Математичні оператори, такі як назви функцій, границі, символи диференціалів та ін.:

`\sin`, `\tan`, `\lim`, `\differential`, `\max`, `\inf`, `\mod`,

позначають прямим шрифтом:

$$\sin x, \quad \operatorname{tg} x, \quad \lim_{x \rightarrow 0} \frac{\sin x}{x} = 1, \quad \frac{dy}{dx}, \quad \max_{1 \leq i \leq n} \Delta x_i, \quad \inf_{t \geq 0} |f_1(t) - f_2(t)|, \quad \operatorname{mod} p.$$

Пакет `amsmath` дає можливість створювати свої математичні оператори у преамбулі документа за допомогою команди `\DeclareMathOperator` після `\usepackage{amsmath}` і в тексті за допомогою команди `\operatorname`, які генерують прямий шрифт для назви математичного оператора та формують пробіл між цим оператором і його аргументом. Наприклад, команда

`\DeclareMathOperator{\Im}{\Im}`

у преамбулі документа і команда

`\Im z`

генерує формулу

`\Im z`,

яку генерує і команда

`\operatorname{\Im}z`

у тексті.

У математичних текстах формули всередині текста [1, 10] генеруються командою `$...$` або командою `\(...\)`. L^AT_EX розбиває автоматично довгі формули в тексті на кілька рядків. Довгі виключні формули розбиває автор на кілька рядків за допомогою таких оточень пакета `amsmath`, як `\multline`, `\align`, `\aligned`, `\alignat`, `\split`, `\gather`.

Розглянемо, як правильно створювати дужки у багаторядковій виключній формулі. Для автоматичного формування у L^AT_EX розмірів дужок використовуються оператори `\left` і `\right` перед відповідними дужками. При цьому перед переходом на наступний рядок необхідно закрити оператор `\left` або `\right` відповідним оператором `\right`. або `\left`. Наприклад, код

```
\begin{multline*}
\int\limits_{\square^{\nu} \times \square^{\nu}} \{\mathbb{R}^{\nu}\} \times \{\mathbb{R}^{\nu}\}
\{\differential{d}\{x_2\} \setminus \{H_2\}, \setminus \{F_2(t, \{x_1\}, \{x_2\} | F_1(t))\}\} =
\int\limits_{\square^{\nu} \times \square^{\nu}} \{\mathbb{R}^{\nu}\} \times \{\mathbb{R}^{\nu}\}
\{\differential{d}\{x_2\}
\left\{\left\{\sum\limits_{i=1}^2 \frac{p_i^2}{2}\right\} + \Phi(\{q_1\} - \{q_2\})\right\}, \setminus \right. \setminus
\left. \left\{\{F_2(t, \{x_1\}, \{x_2\} | F_1(t))\}\right\} \right. \setminus \right.
\end{multline*}
```

генерує таку формулу:

$$\int_{\square^{\nu} \times \square^{\nu}} dx_2 \{H_2, F_2(t, x_1, x_2 | F_1(t))\} = \int_{\square^{\nu} \times \square^{\nu}} dx_2 \left\{ \sum_{i=1}^2 \frac{p_i^2}{2} + \Phi(q_1 - q_2), F_2(t, x_1, x_2 | F_1(t)) \right\}.$$

Проте висота дужок для кожного оператора `\left` і `\right` генерується незалежно одна від одної. Для вирішення цієї проблеми слід використати екзотичну команду вертикального фантома `\vphantom`, яка шукає найвищий елемент. У наведеному прикладі найвищим елементом є $\sum_{i=1}^2 \frac{p_i^2}{2}$. Тоді код

```
\begin{multline*}
\int\limits_{\square^{\nu} \times \square^{\nu}} \{\mathbb{R}^{\nu}\} \times \{\mathbb{R}^{\nu}\}
\{\differential{d}\{x_2\} \setminus \{H_2\}, \setminus \{F_2(t, \{x_1\}, \{x_2\} | F_1(t))\}\} =
\int\limits_{\square^{\nu} \times \square^{\nu}} \{\mathbb{R}^{\nu}\} \times \{\mathbb{R}^{\nu}\}
\{\differential{d}\{x_2\}
\left\{\left\{\sum\limits_{i=1}^2 \frac{p_i^2}{2}\right\} + \Phi(\{q_1\} - \{q_2\})\right\}, \setminus \right. \setminus
\left. \left\{\{F_2(t, \{x_1\}, \{x_2\} | F_1(t))\}\right\} \right. \setminus \right.
\end{multline*}
```

```

\left\{\sum_{i=1}^2 \frac{p_i^2}{2} + \Phi(q_1 - q_2)\right\} \\
\left. \left\{F_2(t, x_1, x_2 | F_1(t))\right\} \right\} \\
\phantom{\left. \left\{F_2(t, x_1, x_2 | F_1(t))\right\} \right\}} \\
\end{multline*}

```

генерує таку формулу:

$$\int_{\square^v \times \square^v} dx_2 \{H_2, F_2(t, x_1, x_2 | F_1(t))\} = \int_{\square^v \times \square^v} dx_2 \left\{ \sum_{i=1}^2 \frac{p_i^2}{2} + \Phi(q_1 - q_2), F_2(t, x_1, x_2 | F_1(t)) \right\}.$$

Інший спосіб вирішення цієї проблеми є використання пакету `brqpn`, розробленого М. Доуномсом і М. Хогольмом.

Таким чином, у цій статті розглянуто три основні і проміжні стильові пакети бібліографічного інструмента `ВіВТЕХ`, які є актуальними при створенні математичних текстів. Для налаштування стилів у базі даних і посиланнях наведено бібліографічний інструмент `CrossTЕХ`. Наведено деякі правила написання математичних формул.

1. Балдин Е. М. Компьютерная типография L^AT_EX / Е. М. Балдин. – СПб.: БХВ-Петербург, 2008.
2. Беляков Н. С. T_EX для всех / Н. С. Беляков, В. Е. Палощ, П. А. Садовский. – М.: Книжный дом «Либроком», 2009.
3. Губаль Г. М. Анімація в математичних текстах на мові L^AT_EX / Г. М. Губаль // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2013. – № 11.
4. Губаль Г. М. L^AT_EX як видавничча система для створення математичних текстів і для програмування / Г. М. Губаль // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2013. – № 12.
5. Губаль Г. М. Стратегії для створення математичної статті у видавниччій системі L^AT_EX / Г. М. Губаль // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2013. – № 13.
6. Дубинич В. Н. Использование системы L^AT_EX для подготовки научных изданий / В. Н. Дубинич, М. В. Дубинич // Перспективы развития высшей школы: материалы V Международной науч.-метод. конф., Гродно: ГГАУ, 2012.
7. Жуков М. Ю. Оформление математических текстов при помощи пакета L^AT_EX2_ε / М. Ю. Жуков, Е. В. Ширяева. – Ростов н/Д: Изд-во ООО «ЦВВР», 2003.
8. Жуков М. Ю. L^AT_EX2_ε: искусство набора и вёрстки текстов с формулами / М. Ю. Жуков, Е. В. Ширяева. – Ростов н/Д: Изд-во ЮФУ, 2009.
9. Каров П. Шрифтовые технологии. Описание и инструментарий / П. Каров. Перевод с английского Карпинского О. С. и Куликова И. И. под редакцией, с предисловием и дополнением Ефимова В. В. – М.: Мир, 2001.
10. Кнут Д. Е. Всё про T_EX / Д. Е. Кнут. Пер. с англ. М. В. Лисиной. – Протвино: АО RDT_EX, 1993.
11. Котельников И. А. L^AT_EX по-русски / И. А. Котельников, П. З. Чеботаев. – Новосибирск: Сибирский хронограф, 2004.
12. Львовский С. М. Набор и вёрстка в системе L^AT_EX / С. М. Львовский. – 3-е изд., испр. и доп. – М.: МЦНМО, 2003.
13. Ширяева Е. В. Введение в T_EX–программирование / Е. В. Ширяева, И. В. Ширяева. – Ростов н/Д: Изд-во ЮФУ, 2010.
14. Goossens M. The L^AT_EX companion / M. Goossens, F. Mittelbach, A. Samarin. – Addison-Wesley, 1994. Русский перевод: Гуссенс М. Путеводитель по пакету L^AT_EX и его расширению L^AT_EX2_ε / М. Гуссенс, Ф. Миттельбах, А. Самарин. Перевод с английского Маховой О. А., Третьякова Н. В., Тюменцева Ю. В. и Чистякова В. В. под редакцией Маховой И. А. – М.: Мир, 1999.
15. Копка Н. Guide to L^AT_EX / Н. Копка, Р. Daly. – Addison-Wesley, 2004.
16. L^AT_EX. A document preparation system, user's guide and reference manual / L. Lamport. – Addison-Wesley, 1994.
17. Spivak M. The joy of T_EX. A gourmet guide to typesetting with the A_MS-T_EX macro package. – American mathematical society, Providence, RI, 1990. Русский перевод: Спивак М. Восхитительный T_EX: руководство по комфортному изготовлению научных публикаций в пакете A_MS-T_EX / М. Спивак. – М.: Мир, 1993.

УДК 681.322
Г. М. Губаль
Луцький національний технічний університет

ВИШУКАНІ КОМАНДИ ДЛЯ СТВОРЕННЯ МАТЕМАТИЧНИХ ТЕКСТІВ У ВИДАВНИЧІЙ СИСТЕМІ L^AT_EX

Губаль Г. М. Вишукані команди для створення математичних текстів у видавничій системі L^AT_EX. У статті розглядаються вишукані команди для створення математичних текстів у видавничій системі L^AT_EX. Досліджуються проблеми конвертації T_EX-скриптів у html-формат.

Ключові слова: видавнича система L^AT_EX, \cancel, стильовий пакет, L^AT_EX2html, T_EX4ht.

Літ. 16.

Губаль Г. Н. Изысканные команды для создания математических текстов в издательской системе L^AT_EX. В статье рассматриваются изысканные команды для создания математических текстов в издательской системе L^AT_EX. Исследуются проблемы конвертации T_EX-скриптов в html-формат.

Ключевые слова: издательская система L^AT_EX, \cancel, стилевой пакет, L^AT_EX2html, T_EX4ht.

Лит. 16.

Hubal H. M. Refined commands for creation of the mathematical texts in the publishing system L^AT_EX. In the article refined commands for creation of the mathematical texts in the publishing system L^AT_EX are considered. The paper investigates problems of converting of T_EX-scripts into html-format.

Keywords: publishing system L^AT_EX, \cancel, style package, L^AT_EX2html, T_EX4ht.

Bibl. 16.

L^AT_EX – видавнича система для створення математичних текстів [1-5, 8-11, 13]. Фундаментальна концепція цієї системи – поняття стилю документа [5-10, 13-16]. Робота з видавничою системою L^AT_EX подібна до програмування, тим більше, що ця система розрахована на використання текстового режиму, командного рядка і багатьох конфігураційних файлів [4, 12].

Видавнича система L^AT_EX створена так, що навіть досвідчені T_EXніки іноді вивчають нові вишукані команди для втілення їхніх ідей. Розглянемо деякі тонкі питання, пов'язані із створенням математичних текстів у видавничій системі L^AT_EX при підключеному стильовому пакеті amsmath:

```
\usepackage{amsmath}
```

у преамбулі документа.

Для створення спеціальних матриць, в яких елементи одного стовпця можуть бути переміщені в інший стовпець, оскільки інші елементи можуть переміститись на їхнє місце, тобто матриць вигляду

$$A = \begin{array}{ccc} \cancel{1} & 2 & 3 \\ \cancel{4} & 5 & 6 \\ \cancel{7} & 8 & \cancel{9} \end{array}$$

вихідний код запишеться так:

```
\mathcal{A}=\left(
\begin{array}{ccc}
1 & 2 & \&\& 3 \\
4 & 5 & \&\& 6 \\
7 & 8 & \makebox[2\arraycolsep]{\smash{\vdots}} \\
& & \& 9
\end{array}
\right)
```

Для створення матриць із границями вигляду

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{matrix} 1 & 3 & 4 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix} \end{matrix}$$

вихідний код запишеться так:

```
\mathcal A=
\bordermatrix{
& 1 & 2 & 3 \cr
1 & 1 & 3 & 4 \cr
2 & 4 & 5 & 6 \cr
3 & 7 & 8 & 9 \cr
}
```

Для позначення скорочення членів у математичних виразах можна використовувати команду `\not`. Якщо цю команду поставити перед командою, яка генерує символ, то дістанемо зображення перекресленого символу (якщо підключити пакет `amssymb`, то можна замість команди `\not` використовувати готові команди цього пакета для перекреслених символів). Оскільки не дуже зручно використовувати команду `\not`, то можна підключити пакет `cancel`:

```
\usepackage{cancel}
```

у преамбулі документа. Тоді для створення виразу

$$\frac{(a \cancel{+} b)(a - b)}{(a \cancel{+} b)(a^2 - ab + b^2)}$$

у тілі документа код набуде вигляду:

```
\dfrac{\cancel{(a+b)}(a-b)}{\cancel{(a+b)}(a^2-ab+b^2)}
```

Багаторядкові виключні формули з правою фігурною дужкою, в яких вираз у лівій частині виглядає по-різному в різних випадках, можна оформити за допомогою оточення `array`. Наприклад, багаторядкова виключна формула

$$\begin{matrix} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{matrix} \stackrel{\text{def}}{=} \text{sgn } x$$

запишеться одним із способів:

1) за допомогою оточення `array`:

```
\left.
\begin{array}
... \\
... \\
...
\end{array}
\right\}
```

2) за допомогою оточень `aligned`, `gathered`, `alignedat`;

3) за допомогою оточення "revert cases", яке можна назвати `revcas` у преамбулі документа:

```
\usepackage{amsmath}
\makeatletter
\newenvironment{revcas}{%
\let@ifnextchar\new@ifnextchar
```

```
\left.%
\def\arraystretch{1.2}%
% можна здійснити вирівнювання відмінне від вирівнювання по лівому краю
\array{@{}l@{\quad}l@{}}%
}{\endarray\right\}}
\makeatother
```

Тоді нове оточення `revcas` можна використати у тілі документа:

```
\begin{equation}
\begin{revcas}
-1, & x < 0 \\
0, & x = 0 \\
1, & x > 0
\end{revcas}=\operatorname{sgn} x
\end{equation}
```

Системи рівнянь можна набирати за допомогою оточення `agga`. Це оточення можна використовувати не тільки у виключних формулах, але й у формулах всередині тексту, хоча результат при цьому, як правило, виглядає негарно. Якщо необхідно, щоб окремі рівняння в системі були занумеровані, можна використати оточення `eqnagga` (воно створює лише виключні формули). Воно працює так само, як і оточення `agga`, але при цьому в кожного рівняння автоматично друкується його номер (аналогічно до того, як автоматично друкується номер виключної формули, створеної за допомогою оточення `equation`). При цьому оточення `eqnagga` не створює фігурної дужки, яка охоплює систему рівнянь. Щоб вирішити цю проблему, використаємо оточення `subequations` і `empheq`. Наприклад, система рівнянь

$$\begin{aligned} f_1(x) &= b_1 & (1) \\ f_2(x) &= b_2 & (2) \\ f_3(x) &= b_3 & (3) \end{aligned}$$

створюється таким кодом:

```
\begin{subequations}
\begin{empheq}[left=\empheqlbrace]{align}
f_1(x) &= b_1 \\
f_2(x) &= b_2 \\
f_3(x) &= b_3
\end{empheq}
\end{subequations}
```

Для створення нумерованого списку використовується оточення `enumerate`. У таких списках кожен елемент вводиться командою `\item` без аргумента, і при друці він буде відмічений номером (ці номери створюються системою L^AT_EX автоматично; якщо переставити місцями які-небудь елементи списку, щось видалити або додати, то нумерація автоматично зміниться). Цей номер можна, наприклад, виділити, помістивши в коло. Наприклад, такий список:

- ① Елемент 1
- ② Елемент 2
- ③ Елемент 3
- ④ ...
- ⑤ Елемент n

генерується таким кодом у тілі документа:

```
\begin{enumerate}[\textcircled{\arabic{enumi}}]
\item Елемент 1
\item Елемент 2
\item Елемент 3
```

```
\item \ldots  
\item Елемент \(\n)  
\end{enumerate}
```

а в преамбулі документа підключимо пакет enumerate:

```
\usepackage{enumerate}
```

Іноді доцільно нумерувати абзаци в тексті документа. Наприклад, запишемо такий код:

```
\newcounter{vcount}  
\def\Header#1{\medskip%  
  \hbox{\bfseries #1}%  
  \setcounter{vcount}{1}%  
  \everypar{\arabic{vcount}}%  
  \stepcounter{vcount}\}%  
}
```

До цього коду звертаємося так:

```
\Header{Заголовок}  
Перший абзац.
```

Другий абзац.

Третій абзац.

У результаті дістанемо такий текст:

Заголовок

1 Перший абзац.

2 Другий абзац.

3 Третій абзац.

Для світових наукових видань єдиним форматом є L^AT_EX. Проте для розміщення документів у Інтернеті часто використовують html-формат. У цьому випадку застосовують програми для перетворення формату L^AT_EX у html.

Найбільш відомими конверторами T_EX-скриптів у html-формат є L^AT_EX2html, T_EX4ht. Взагалі кажучи, конвертори не можуть конвертувати текст без помилок. Однією з причин є використання власних макросів, яке порушує роботу конверторів. Оскільки існує відмінність між налаштуванням параметрів T_EX і html (вони виконують різні завдання), то вони неконгруентні.

Для успішної конвертації T_EX-скриптів у html-формат необхідно:

- створити вихідний код, доступний для розпізнавання;
- використовувати пробіл над і під усіма окремими контрольними послідовностями;
- не використовувати власних макросів;
- вибирати імена команд і оточень, які існують у L^AT_EX.

1. Балдин Е. М. Компьютерная типография L^AT_EX / Е. М. Балдин. – СПб.: БХВ-Петербург, 2008.

2. Беляков Н. С. T_EX для всех / Н. С. Беляков, В. Е. Палощ, П. А. Садовский. – М.: Книжный дом «Либроком», 2009.

3. Губаль Г. М. Анімація в математичних текстах на мові L^AT_EX / Г. М. Губаль // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2013. – № 11.

4. Губаль Г. М. L^AT_EX як видавнича система для створення математичних текстів і для програмування / Г. М. Губаль // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2013. – № 12.

5. Дубинич В. Н. Использование системы L^AT_EX для подготовки научных изданий / В. Н. Дубинич, М. В. Дубинич // Перспективы развития высшей школы: материалы V Международной науч.-метод. конф., Гродно: ГГАУ, 2012.

6. Жуков М. Ю. Оформление математических текстов при помощи пакета L^AT_EX2_ε / М. Ю. Жуков, Е. В. Ширяева. – Ростов н/Д: Изд-во ООО «ЦВВР», 2003.
7. Жуков М. Ю. L^AT_EX2_ε: искусство набора и вёрстки текстов с формулами / М. Ю. Жуков, Е. В. Ширяева. – Ростов н/Д: Изд-во ЮФУ, 2009.
8. Каров П. Шрифтовые технологии. Описание и инструментарий / П. Каров. Перевод с английского Карпинского О. С. и Куликова И. И. под редакцией, с предисловием и дополнением Ефимова В. В. – М.: Мир, 2001.
9. Кнут Д. Е. Всё про T_EX / Д. Е. Кнут. Пер. с англ. М. В. Лисиной. – Протвино: АО RDT_EX, 1993.
10. Котельников И. А. L^AT_EX по-русски / И. А. Котельников, П. З. Чеботаев. – Новосибирск: Сибирский хронограф, 2004.
11. Львовский С. М. Набор и вёрстка в системе L^AT_EX / С. М. Львовский. – 3-е изд., испр. и доп. – М.: МЦНМО, 2003.
12. Ширяева Е. В. Введение в T_EX–программирование / Е. В. Ширяева, И. В. Ширяева. – Ростов н/Д: Изд-во ЮФУ, 2010.
13. Goossens M. The L^AT_EX companion / M. Goossens, F. Mittelbach, A. Samarin. – Addison-Wesley, 1994. Русский перевод: Гуссенс М. Путеводитель по пакету L^AT_EX и его расширению L^AT_EX2_ε / М. Гуссенс, Ф. Миттельбах, А. Самарин. Перевод с английского Маховой О. А., Третьякова Н. В., Тюменцева Ю. В. и Чистякова В. В. под редакцией Маховой И. А. – М.: Мир, 1999.
14. Kopka H. Guide to L^AT_EX / H. Kopka, P. Daly. – Addison-Wesley, 2004.
15. Lamport L. L^AT_EX. A document preparation system, user's guide and reference manual / L. Lamport. – Addison-Wesley, 1994.
16. Spivak M. The joy of T_EX. A gourmet guide to typesetting with the AMS-T_EX macro package. – American mathematical society, Providence, RI, 1990. Русский перевод: Спивак М. Восхитительный T_EX: руководство по комфортному изготовлению научных публикаций в пакете AMS-T_EX / М. Спивак. – М.: Мир, 1993.

УДК 681.515.8

С.А. Губіш, Н.В. Здолбіцька, А.П. Здолбіцький
Луцький національний технічний університет**ПОШИРЕННЯ СИГНАЛУ НА ВІДКРИТОМУ ПРОСТОРИ**

С.А. Губіш, Н.В. Здолбіцька, А.П. Здолбіцький. Поширення сигналу на відкритому просторі. Наведено відомості про поширення радіосигналу на відкритому просторі, а також розглянуто особливий випадок – відбиття хвилі від поверхні землі. Здійснено практичну реалізацію та вимірювання рівня сигналу на відкритому просторі.

Ключові слова: сигнал, хвиля, антена, радіопередавач, СС2500, рівняння Фрііс.

Форм. 4. Рис. 4. Літ. 10.

С.А. Губиш, Н.В. Здолбицкая, А.П. Здолбицкий. Распространение сигнала на открытом пространстве. Приведены сведения о распространении радиосигнала на открытом пространстве, а также рассмотрен особый случай – отражение волны от поверхности земли. Осуществлена практическая реализация и измерения уровня сигнала на открытом пространстве.

Ключевые слова: сигнал, волна, антенна, радиопередатчик, СС2500, уравнение Фриис.

S.A. Hubish, N.V. Zdobitska, A.P. Zdobitsky. Signal propagation in open space. Information on radio propagation in the open air, and considered a special case – the wave reflection from the surface was described. Practical implementation and measurement of signal level in the open air environment was done.

Keywords: signal, wave, antenna, radio, СС2500, Friis equation.

Постановка проблеми. Радіозв'язок ґрунтується на основі передачі енергії сигналу з одного місця в інше. Отримана енергія сигналу повинна бути достатньо великою, щоб відрізнити корисний сигнал від шуму, який завжди присутній. Ця залежність описується як відношення сигналу до шуму (S/N).

Необхідне відношення для нормального радіозв'язку зазвичай вказується у технічній документації до приймача, хоча також досить часто використовується таке поняття як чутливість – це абсолютний рівень сигналу (S). Якщо ж вказана лише чутливість, то мається на увазі, що пристрій знаходиться при кімнатній температурі і присутній лише тепловий шум.

Далі буде розглянуто теорію, що використовується для визначення радіусу поширення радіосигналу у повітрі.

Метою роботи є визначення радіусу дії радіопередавача, на якому можливе стабільне з'єднання, для створення стабільних бездротових мереж.

Рівняння Фрііс. Відстані у радіозв'язку, як правило, описуються рівнянням Фрііс (1).

$$P_R = \frac{G_T G_R \lambda^2}{(4\pi)^2 d^n}, \quad n = 2. \quad (1)$$

де

P_R – потужність, що надходить з прийомної антени;

P_T – потужність, що подається на передавальну антену;

G_R – підсилення приймальної антени;

G_T – підсилення передавальної антени;

λ – довжина хвилі, де $\lambda = c/f$, c – швидкість світла, і f – частота;

d – відстань;

c – швидкість світла у вакуумі $299.972458 \cdot 10^6$ [м/с]

Це рівняння описує залежність між відстанню, частотою (довжиною хвилі), коефіцієнтом підсилення антени та потужністю сигналу.

Рівень сигналу. Рівняння Фрііс часто називають рівнянням рівня сигналу. Різницю між прийнятою потужністю сигналу (P_R) і чутливістю приймача називають границями сигналу. У реалістичній моделі сигнал піддається додатковим втратам, що повинні бути додані до втрат, що розраховані рівнянням Фрііс.

Діапазон – це відстань, на якій можливий стабільний радіозв'язок з рівнем сигналу, що дорівнює рівню чутливості приймача. У цифрових системах радіозв'язку чутливість часто визначається як рівень вхідного сигналу.

Відбиття від землі. У типовій радіопередачі хвилі відбиваються і затухають при проходженні крізь об'єкти. Обчислення діапазону в реалістичному середовищі є складним

завданням, що вимагає величезних обчислювальних ресурсів. Багато середовищ включають в себе деякі мобільні об'єкти, що додають складності завданню. Більшість вимірювань діапазону виконуються на великих відкритих просторах, що не містять перешкод та інших рухомих об'єктів чи джерел радіовипромінювання. Це слід в першу чергу зробити, щоб отримати стійкі результати. Рівняння Фрііс можна застосовувати лише на вільному просторі. Звичайне обладнання зазвичай працює близько до землі. Це, звичайно, необхідно враховувати, щоб зробити достовірні розрахунки діапазону дії.

Рисунок 1 ілюструє ситуацію з нескінченною, абсолютно плоскою поверхнею землі за відсутності інших об'єктів, що перешкоджали б сигналу. Загальна отримана енергія може бути змодельована як векторна сума безпосередньо направленою сигналу та відбитого сигналу.

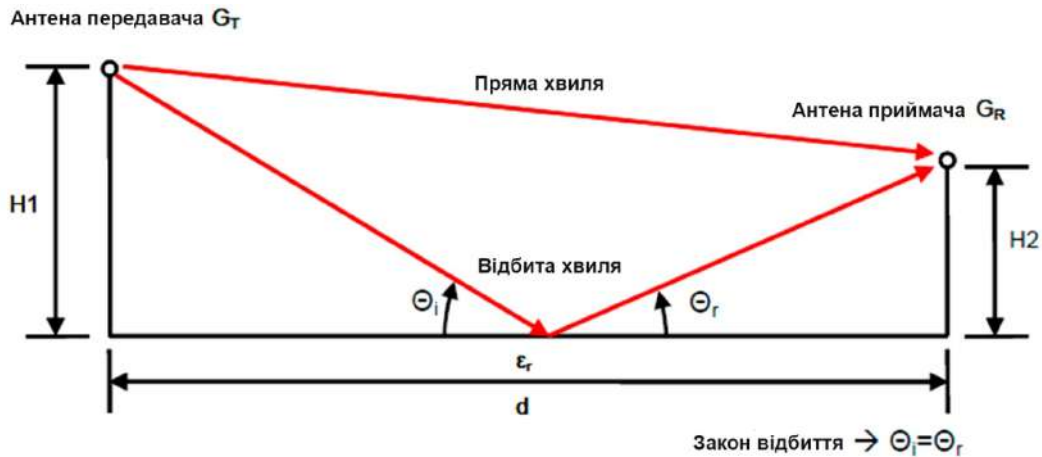


Рис. 1. Відбиття сигналу від землі

Дві хвилі додаються конструктивно або деструктивно в залежності від їх різниці фаз в приймач. Амплітуда і фаза прямих хвиль змінюється залежно від пройденої відстані. Амплітуда відбитої хвилі залежить від загальної пройденої відстані і коефіцієнту відбиття (G), що характеризує хвилю до і після відбиття.

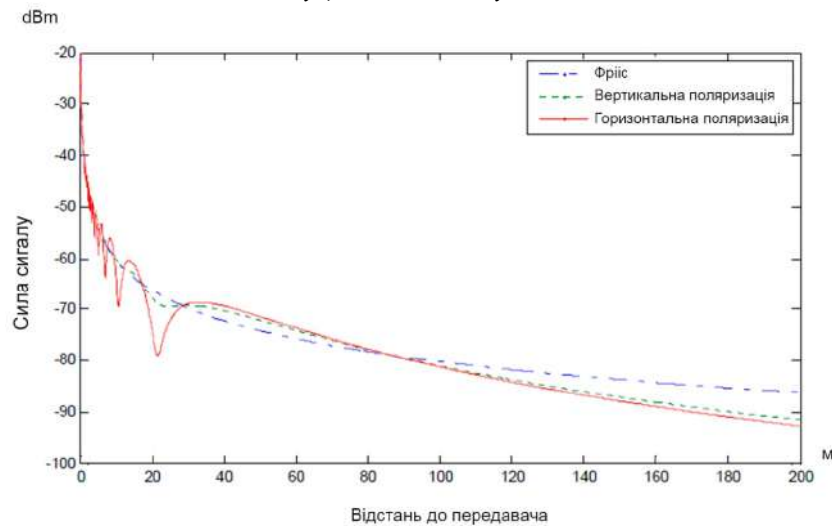
Коефіцієнт відбиття. Кожен раз, коли радіосигнал проходить крізь середовища з різною діелектричної проникністю, частина енергії відбивається, а решта енергії передається далі. Частина, що відбилася, залежить від поляризації сигналу, кута падіння і різної діелектричної проникності середовищ (ϵ_r , μ_r і σ). Якщо припустити, що обидві речовини мають рівну проникність $\mu_r = 1$ і що одне середовище – вільний простір, то рівняння (2) і (3) визначатимуть коефіцієнти відбиття Френеля для вертикальної і горизонтальної поляризації сигналів.

$$G_v = \frac{(\epsilon_r - j60\sigma\lambda)\sin(\theta_i) - \sqrt{\epsilon_r - j60\sigma\lambda - \cos^2(\theta_i)}}{(\epsilon_r - j60\sigma\lambda)\sin(\theta_i) + \sqrt{\epsilon_r - j60\sigma\lambda - \cos^2(\theta_i)}}, \quad (2)$$

$$G_h = \frac{\sin(\theta_i) - \sqrt{\epsilon_r - j60\sigma\lambda - \cos^2(\theta_i)}}{\sin(\theta_i) + \sqrt{\epsilon_r - j60\sigma\lambda - \cos^2(\theta_i)}}. \quad (3)$$

У рівняння необхідно підставити деякі електричні параметри для ґрунту. З таблиць можна взяти параметри для різних типів середовища. Прийmemo $\epsilon_r = 18$ і $\sigma = 0$ і надалі будемо використовувати ці значення для усіх розрахунків.

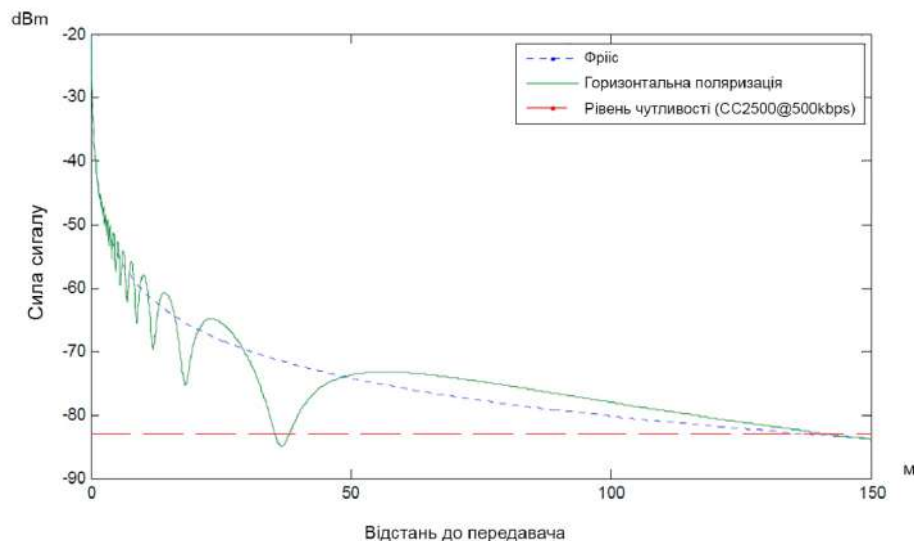
У системах, де $H1$ і $H2$ (висоти приймача і передавача над рівнем землі відповідно) є невеликою порівняно з D , рівняння (2) і (3) може бути спрощено до $G_v = G_h = -1$. Тобто в системах з малим кутом падіння хвилі вся енергія відбивається. Зміна фази відбитої хвилі має важливе значення для рівня сигналу, як показано на рисунку 2.



$$H_1 = H_2 = 1.15 \text{ м, } \epsilon_r = 18, f = 2445 \text{ МГц}$$

Рис. 2. Втрати при передачі відповідно до поляризації сигналу

Рисунок 2 показує вплив поляризації і вплив землі у вимірах на відкритому просторі. На графіку можна побачити велику різницю між рівнем сигналу, визначеним рівнянням Фрііс для відкритого простору і рівнем сигналу, якщо врахувати вплив землі. Графік також показує, що горизонтальна поляризація (Н) є ефективнішою для багатокрокового поширення сигналу ніж вертикальна (V). На далеких відстанях рівень сигналу із врахуванням впливу землі є значно нижчим, ніж розрахований рівнянням Фрііс. Вертикальна поляризація сигналу дозволяє передавати його з більш високою енергією на великі відстані ніж з використанням горизонтальної поляризації.



$$H_1 = H_2 = 1.15 \text{ м, } \epsilon_r = 18, f = 2445 \text{ МГц}$$

Рис. 3. Багатокрокове згасання

На рисунку 3 показані розрахункові значення для горизонтально поляризованого сигналу частотою 2445 МГц. Для порівняння до графіку також включені величини, розраховані рівнянням Фрііс для вільного простору і рівня чутливості приймача CC2500 рівного 500 кбод. Для вимірювання ефективного радіусу дії для CC2500 на цій швидкості передачі даних, як правило, використовується EB PER тест з поступовим збільшенням відстані між пристроями радіозв'язку. На рисунку показано, що зв'язок буде втрачено через близько 35 метрів. Очевидно, що діапазон може бути значно більшим. Для того, щоб використати незадіяний діапазон, пристрої повинні бути розміщені на більш ніж 39 м один від одного, щоб відновити зв'язок.

Розташування цієї сліпої плями буде змінюватися в залежності від частот, наземних електричних характеристик і висоти розміщення антени. Дуже важливо знати про це під час вимірювань, щоб визначити чи це є локальна сліпа пляма, чи досягнутий максимальний радіус дії обладнання.

Різниця між рівнем визначеним рівнянням Фрііс і чутливістю приймача часто позначається як запас згасання.

Шум. Шум є ще одним важливим параметром при визначенні діапазону. Шум можна класифікувати за його джерелами. Тепловий шум є шум, що створюється всіма об'єктами через їх молекулярну теплову діяльність.

Інше радіовипромінювання можна вважати ще одним джерелом шуму. Шум від іншого електричного обладнання досить важко описати математичними/статистичними моделями. Рівняння (4) описує тепловий шум.

$$v_n = \sqrt{\frac{4hfBR}{e^{\frac{hf}{kT}} - 1}} \approx \sqrt{4kTBR} \text{ [Volt}_{rms} \text{]}. \quad (4)$$

Температура, ефективна ширина смуги шуму і опір визначають загальний тепловий шум.

При кімнатній температурі (300 К, 27°C) це рівняння часто апроксимується до $-174 \text{ дБм} + 10\log_{10}(B)$, описуючи ситуацію з досконалим навантаженням.

Тепловий шум не є проблемою під час вимірювань. Слід, однак, перевірити, що територія вільна від інших джерел шуму на тій же смузі частот. Це може бути зроблено за допомогою спектр-аналізатора для пошуку джерел шуму до виконання тесту. Цю перевірку слід регулярно повторювати з однаковими інтервалами під час тесту. Рекомендовано вибирати ділянки з низькою ймовірністю виникнення сторонніх шумів.

Експериментальні дослідження.

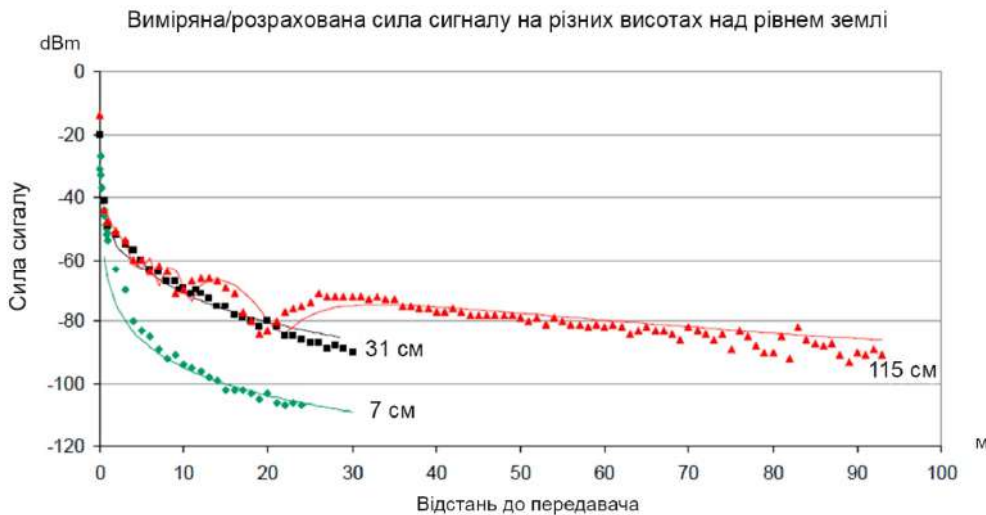


Рис. 4. Рівні сигналу на різних висотах 7см, 31см та 115см над землею

На рисунку 4 показано порівняння між CC2500, що працює в SmartRF04EB і моделлю відображення сигналу від землі у Matlab. Вимірювання були виконані на футбольному полі. Точки – це вимірювання рівня сигналу, а лінії – розрахункові значення.

Фіксований рівень корекції був доданий до розрахункових значень, щоб отримати краще узгодження з виміряними значеннями. Це значення корекції являє собою різницю між ідеальною ізотропною антеною та ефективністю CC2500EM і SmartRF Studio EB.

Виміряна енергія сигналу була вищою для горизонтально поляризованого сигналу. Це пояснюється використанням горизонтально-орієнтованої чвертьхвильової антени.

Висновки

Було розглянуто вплив землі під час вимірювань дальності дії радіопередавача CC2500.

Було показано, що багатокрокове затухання може створювати плутанину під час вимірювань, якщо не враховувати згадане явище. Було виявлено, що земля значно пришвидшує затухання сигналу в порівнянні з розрахованим за рівнянням Фрііс. Отже, земля знижує ефективну дальність.

Було показано, що вертикальна поляризація менш схильна до затухання від відображення від землі і, як наслідок, зменшення діапазону, ніж горизонтальна поляризація. Для просто обладнання поляризація, як правило, не є контрольованим параметром і це не відіграє значної ролі.

Нарешті, було підкреслено, що інші радіовипромінювання в діапазоні вимірювань повинні контролюватися протягом вимірювань.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дымов, А.В. Исследование распространения радиоволн в городских условиях в диапазоне 1800 МГц / А.В. Дымов, В.А. Тимофеев // Труды XIII международной научно-технической конференции «Радиолокация, навигация и связь». – Воронеж, 2007. – Т.1. – С.611-616.
2. Г.А. Ерохин, О.В. Чернышев, Н.Д. Козырев, В.Г. Кочержевский. Антенно-фидерные устройства и распространение радиоволн. Под. ред. Г.А. Ерохина. – 2-е изд. – М.: Горячая линия - Телеком, 2004. – 491 с.
3. Журнал «Беспроводные технологии» №1. – 2007 г.
4. Козлов, Р.Ю. Моделирование распространения радиоволн в городе с учетом рельефа местности / Р.Ю. Козлов, В.А. Тимофеев // Труды X международной научно-технической конференции «Радиолокация, навигация, связь». – Воронеж, 2004. – Т. 3. – С.1845 – 1850.
5. Модели радиоканала. Имитационное моделирование компьютерных сетей. http://student.telum.ru/images/2/28/Network_modeling_lecture_03.pdf
6. Семенов А.И. Распространение радиоволн по естественным трассам. Учебное пособие для вузов. – М., САЙНС-Пресс, – 2005.
7. Электродинамика и распространение радиоволн / В.А. Неганов, О.В. Осипов, С.Б. Раевский, Г.П. Яровой. - М.: Радио и связь, – 2005. – 648 с
8. Яковлев, О.И. Распространение радиоволн / О.И. Яковлев, В.П. Якубов, В.П. Урядов, А.Г. Павелъев. – СПб. Ленанд, – 2009. – 496 с.
9. Radar Technology Encyclopedia, David K. Barton, Sergey A. Leonov, 1997 Artech House Inc. Boston/London
10. http://en.wikiversity.org/wiki/Friis_Transmission_Equation

УДК 621.3.078.4

О.В.Дудік

Луцький національний технічний університет

АСПЕКТИ БЕЗПЕКИ ПРОГРАМУВАННЯ В PHP

О.В.Дудік. Аспекти безпеки програмування на PHP. В статті розглядається ризики при програмуванні на мові PHP на рівнях даних, системи, виконання програм.

Ключові слова: запит, система, мережа, дані, права доступу.

Лит. – 2.

А.В.Дудік. Аспекти безпеки програмування на PHP. В статті розглядається ризики при програмуванні на мові PHP на рівнях даних, системи, виконання програм.

Ключевые слова: запрос, система, сеть, данные, права доступа.

Лит. – 2.

O.V.Dudick. Safety aspects of programming in PHP. This article discusses the risks of programming in PHP on levels of data, systems, program execution.

Keywords: request, system, network, data, permissions.

References – 2.

Актуальність досліджуваної теми полягає в дослідженні можливостей збою роботи веб-систем і ресурсів для ефективного запобігання несанкціонованого втручання в їх роботу.

Проблема полягає в тому, що неефективна побудова веб-ресурсу призводить до його вразливості з боку хакерів, а також невірної роботи при виникненні нестандартних ситуацій.

Для вирішення даної проблеми розглядаються основні аспекти ефективного безпечного програмування на PHP, а також аналізуються різні фактори впливу на дані, систему і т.д.

Ризики на рівні даних

Ризики зберігання даних - це ті ризики, які включають розміщення даних у базі даних або файлової системі. Найбільш широко відомі атаки на цей клас - атаки **SQL injection**. Цей тип атак являє собою атаку в SQL-запит довільних даних, що може дозволити отримати практично будь-яку інформацію з бази даних аж до авторизаційної інформації і кредитних карт.

Найпростіший спосіб уникнути цих атак - це захистити кожну змінну, що використовується, яка використовується для SQL-запиту. На щастя, PHP має декілька вбудованих функцій для цього, наприклад `mysql_escape_string()`. Функція екранує усі спецсимволи в `unescaped_string`, внаслідок чого її можна безпечно використовувати в SQL-запиті.

Приклад екранування змінної:

```
$item = "Zak' s Laptop";  
$escaped_item = mysql_escape_string($item);  
printf("Escaped string: %s\n", $escaped_item);
```

Вищеописаний приклад видасть наступний результат:

Escaped string: Zak\'s Laptop

Деякі програмісти вважають за краще екранувати дані, як тільки вони поступають в прикладення, інші віддають перевагу робити це безпосередньо перед запитом у базу даних. Хорошим стилем вважається ставити екранування перед запитом, тобто можна завжди подивитися код, операції з базою даних і дані, які переходять в екранований запит, і не треба шукати за усім початковим кодом місце екранування.

Другий ризик, - це розміщення паролів в текстовому файлі (маються на увазі в незашифрованому вигляді). Багато прикладень є з відкритим початковим кодом, де паролі знаходяться у відкритому вигляді. Необхідно узяти на замітку, що немає жодної вагомої причини залишати паролі у відкритому вигляді. Не має значення, де зберігаються паролі - в текстовому файлі або у базі даних, завжди необхідно їх розміщувати у вигляді хеша.

Трансформацію паролів досить просто зробити за допомогою PHP - функція `md5()`, слід зашифрувати їх перед збереженням. Оскільки `md5` повторювана функція, можна перевіряти паролі звичайним порівнянням. `MD5`-хеш рядка `str` обчислюється, використовуючи алгоритм **MD5 RSA Data Security, Inc.** Хеш є 32-значним шістнадцятковим числом. Якщо необов'язковий аргумент `raw_output` має значення `TRUE`, то повертається бінарний рядок з 16 символів. (Зауваження: необов'язковий аргумент `raw_output` був доданий в PHP 5.0.0 і за умовчанням рівний `FALSE`.)

Не можна дозволяти змінній з паролем "плавати" по прикладненню. Як тільки пароль був введений, він відразу має конвертуватись. Бажано відразу помістити хеш пароля в змінну і використовувати тільки її.

Слід завжди розділяти про імена користувачів і паролі в різних PHP-файлах з кодом і посилатися на них як на константи або змінні. Це не лише спростить програмування, але і, приміром, якщо необхідно змінити пароль, буде відоме його точне місце знаходження.

Приклад 1. Приклад використання md5():

```
$str = 'apple';  
if (md5($str) === '1f3870be274f6c49b3e31a0c6728957f'){  
echo "Would you like a green or red apple"?;  
exit;  
}
```

Тут результат md5(\$str) є зашифрованим рядком apple.

Також хотілося б торкнутися концепції розподілу прав. Усі користувачі бази даних повинні мати найменші права доступу, необхідні тільки для коректного виконання функцій. Якщо прикладненню потрібне тільки читання з бази даних, воно повинне мати права на виконання SELECT-запиту і ніяких прав доступу до інших баз даних.

Для дотримання цієї концепції рекомендується зробити декілька акаунтів у базі даних. Один акаунт матиме тільки права на запис в потрібні таблиці - INSERT, і повністю відокремлений акаунт матиме права на читання - SELECT. Це дозволить бути упевненим, що ніякі INSERT-запити не будуть випадково виконані, і це зменшить можливі ушкодження, зроблені ін'єкціями SQL.

Звичайно, декілька акаунтів працюють краще, коли є чіткий розподіл між тими, хто може писати у базу даних, і тими, хто може її читати (це використовується в деяких **CMS - Content Management Systems**). У теорії можна використовувати декілька акаунтів у будь-якому прикладненні, але можуть виникнути проблеми з численними підключеннями до бази даних, які мають бути вирішені на стадії дизайну архітектури ПЗ.

Програмуючи великі прикладнення, зручно розбивати код на безліч логічних файлів, але більшість з PHP-програмістів має звичку давати файлам розширення, відмінні від **.php**, наприклад **.inc** або **.config**. Це дуже погана ідея, тому що сервер може бути не налаштований на парсинг цих файлів як PHP, і абсолютно будь-хто може завантажити ці файли як початковий код (що потенційно містить паролі, імена користувачів і іншу приватну інформацію). Як варіант для наочності можна називати такі файли `inc_` або `class_` при необхідності, але розширення все-таки ставити `.php`.

Говорячи про інклудінг (включення, підвантаження) файлів, хотілося б також зазначити особливості безпечної роботи. Якщо наявний PHP-файл необхідно використовувати тільки як частину великого прикладнення, то вставляють його рядки в початок файлу (`__FILE__`, `$_SERVER['PHP_SELF']`).

Це захистить файл від прямого запуску, іншими словами, зупинить його роботу, якщо хтось звернеться до нього без-посередньо. Добре написаний клас або інклуд-файл не повинен робити нічого самостійно, але особливо з цього приводу можна не турбуватися - один раз вставлений рядок на початку файлу знімає усі проблеми.

Інша особливість інклудинга файлів - різниця між функціями `include()` і `readfile()`. Include говорить серверу парсити файл як PHP, тоді як `readfile()` інтерпретує файл як звичайний текст.

Ніколи не слід використовувати `include` з файлами, у яких є публічний доступ на запис, - наприклад, коли у вас є прикладнення, яке додає прийняті від користувача дані в кінець файлу (такі як гостьова книга), або з файлами, які ви не контролюєте (файли на іншому сервері або з можливістю редагування іншими користувачами), — довільний користувач зможе з легкістю впровадити його власний PHP-код у вашу систему.

В той же час ніколи не слід виконувати `readfile` з файлами, які закінчуються на `.php`. При неправильній конфігурації системи виникає ризик розкриття усім вашого початкового коду. Підвівши підсумки, можна сказати, що виконувати `readfile()` можна з `html` -, текстовими і видаленими файлами. `Include()` варто виконувати з локальними `php`-файлами, які необхідно виконати.

Ризики на рівні системи

Системний ризик залежить від того, як саме виконується код. Основний ризик системи у будь-якому прикладненні - невірні дані. Важко абсолютно і достовірно перевірити дані. Як тільки користувач вводить дані в систему, необхідно відразу ці дані перевіряти і фільтрувати. При перевірці даних треба активувати найретельніший фільтр. Приміром, якщо програма має на увазі відсотки, не треба перевіряти, чи було введено що-небудь, а перевіряти чітку відповідність числам між 0 і 100.

Також варто робити перевірки на усіх рівнях коду. Кожного разу як функція отримує дані, перевіряють дані і, якщо вони невірні чи підозрілі, реагують відповідно. Це також дозволить уникнути логічних помилок у розробленому прикладненні.

Ризики на рівні виконання

Небезпечні функції `eval()`, `exec()` і їм подібні (`shell_exec()`, `system()`, `passthru()` і `pcntl_exec()`).

`Eval` виконує будь-який переданий в його змінну php-код. Це досить небезпечно, тому що вже не буде абсолютного контролю над виконуваним кодом. Якщо все-таки необхідно використовувати `eval()`, ніколи не дозволяють йому запустити змінні, в яких дані отримуються від користувача, інакше хакер зможе впровадити свій код. `Exec()` і подібні команди містять аналогічні загрози, що дозволяють скрипту взаємодіяти з командним рядком, і варто використовувати цю можливість якомога рідше.

Існують ризики розкриття інформації. Тому варто прибирати повідомлення про помилки і налагоджувальну інформацію, бо вони можуть дати хакерам інформацію про потенційні помилки в системі. У працюючих системах завжди вимикають повідомлення про помилки і використовують замість цього функцію `PHP errorlog()`.

Останній ризик - це використання сесійних ідентифікаторів. Простіше кажучи, не можна посилати ід-сесії користувачеві. Сесії не є небезпечними, але якщо передається сесійний ід, то хтось інший, відмінний від очікуваного користувача, так званий «man in the middle», може оволодіти сесією користувача. Прикладом може бути перехоплення сесії інтернет-магазину. Зловмисник може отримати дані кредитної карти, змінити адресу доставки або зробити ще дещо гірше залежне від системи.

У статті обговорено основні ризики безпеки при програмуванні в PHP, але усі вони вписуються в декілька простих концепцій:

- ніколи не можна довіряти користувачеві, не дозволяти йому запускати код на сервері і завжди перевіряти дані, що надходять;
- не можна давати користувачеві або програмному забезпеченню рівень доступу більший абсолютного мінімуму, необхідного для успішного виконання завдань;
- не можна розкривати користувачеві більше інформації, чим йому необхідно знати, не дозволяти бачити початковий код, сесійні ID і будь-які повідомлення про помилки, за винятком тих, які створені спеціально для нього.

Реалізація захисту засобами АОП

Термін аспектно-орієнтоване програмування уперше був згаданий в роботі Gregor Kiczales et al. "Aspect - oriented programming" в 1997 році. АОП ставить своєю метою розробити механізм реалізації наскрізної функціональності в ООП системах. АОП не в якій мірі не замінює собою ООП, ця техніка усього лише добудовує концепції ООП. Для реалізації прикладів статті використовувався AspectJ - реалізація АОП для Java. У загальних словах АОП надає новий механізм композиції відмінний від наявних в ООП. Для прикладу розглянемо терміни, в яких працює AspectJ :

- `JoinPoint` - певна точка у виконанні програми, це може бути виконання методу, зміна атрибуту класу, виклик методу, викидання виключення і так далі.
- `Pointcut` - набір (0..N) точок виконання програми, наприклад виконання усіх методів що починаються з «get» класів певного пакету, або виклик методів класів тих, що реалізують деякий інтерфейс.
- `Advice` - java код виконуваний до (`before advice`), потім (`after advice`) або замість (`around advice`) кожної точки виконання що входять в певний `pointcut`.
- `Aspect` - модуль в термінах АОП, аналог класу в Java може містити публічні/приватні `pointcut` і `advice` крім того звичайні методи класу, можуть наслідувати і бути абстрактними.

• Introduction - метод зміни структури спадкоємстві і реалізацій існуючої системи, застосовується наприклад для того, щоб додати додатковий інтерфейс до існуючого класу або змінити ланцюг спадкоємців.

Загалом, аспекти додають додаткову функціональність в точки виконання програми (pointcut) через advice. Важливе те, що pointcut можуть збирати не лише точки виконання, але і контекст в яких ці точки знаходяться, наприклад, якщо pointcut визначає усі виклики методів someMethod усіх об'єктів класу SomeClass, то контекст виконання це:

- Об'єкт, метод якого викликаний.
- Параметри методу.
- Об'єкт, що викликав метод.

Приклад реалізації

```
package aop.example;

import aop.example.model.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.IOException;

/**
 * Авторизаційний аспект
 * @author Zubairov Renat
 */
public aspect AuthorizationAspect {

    /**
     * pointcut метод, що включає
     з ServletRequest для
     * того що б потім можна було
     би отримати його в об'єднанні
     * з іншим pointcut (патерн "червоточина")
     */
    pointcut requestMethod
(ServletRequest request) :
        execution(* aop.example.EntranceFilter.
doFilter(ServletRequest, ServletResponse,
FilterChain))
        && args(request, ServletResponse
FilterChain);

    /**
     * pointcut визначальний метод фільтру
     у якому ми відловлюватимемо
     * виключення
     */
    public pointcut doFilterMethod
(ServletRequest srequest, ServletResponse
sresponse, EntranceFilter filter) :
        execution(void aop.example.
EntranceFilter.doFilter(ServletRequest
ServletResponse, FilterChain))
        && args(srequest
sresponse, FilterChain) && this(filter);

    // усі методи проводять читання інформації
об'єктів моделі
    pointcut readMethods(Object object)
: execution (public * aop.example.model.
*.get*(.) && this(object));

    // усі методи проводять додавання
об'єктів моделі
    pointcut addMethods(Object object)
: execution (public * aop.example.model.
*.add*(.) && this(object));

    // усі методи проводять видалення
об'єктів моделі
    pointcut deleteMethods(Album album)
: execution (public * aop.example.
model.AlbumList.deleteAlbum(Album)) && args(album);

    // методи перевірки на доступність
читання (пред-перевірка)
```

```

        pointcut controlledRead(Object object)
        : execution(public boolean aop.example.model.
Controlled+.isReadable()) && this(object);

        // методи перевірки на доступність
        додавання (пред-перевірка)
        pointcut controlledAdd(Object object)
        : execution(public boolean aop.example.model.Controlled+.
isAddable()) && this(object);

        // методи перевірки на доступність
        видалення (пред-перевірка)
        pointcut controlledDelete(Object object)
        : execution(public boolean aop.example.model.
Controlled+.isDeletable()) && this(object);

        // Виклик методів читання що сталися в потоці
        виконання
        // що йде за викликом методу фільтру
        // ми об'єднали два pointcut - реалізація
        патерну "червоточина"
        pointcut readAccess(ServletRequest request
Object object) :
                cflow(requestMethod(request))
&& readMethods(object);

        // те ж саме тільки для додавання
        pointcut addAccess(ServletRequest request
Object object) :
                cflow(requestMethod(request)) &&
addMethods(object);

        // те ж саме тільки для видалення
        pointcut deleteAccess(ServletRequest request
Album album) :
                cflow(requestMethod(request))&&
deleteMethods(album);

        // пред-перевірка на читання
        pointcut readCheck(ServletRequest request
Object object) :
                cflow(requestMethod(request))&&
controlledRead(object);

        // пред-перевірка на додавання
        pointcut addCheck(ServletRequest request
Object object) :
                cflow(requestMethod(request))&&
controlledAdd(object);

        // пред-перевірка на видалення
        pointcut deleteCheck(ServletRequest request
Object object) :
                cflow(requestMethod(request))&&
controlledDelete(object);

        /**
         * Around advice що відловлює виключення
         * і відправляючий запит на сторінку з помилкою
         */
        void around(ServletRequest srequest
ServletResponse sresponse, EntranceFilter filter)
throws IOException, ServletException{
doFilterMethod(srequest, sresponse, filter){
    try {
                // виконуємо метод фільтру
                proceed(srequest, sresponse, filter);
    } catch (AuthorizationException e){
                // ловимо виключення
                srequest.setAttribute("error_message",
e.getMessage());
                // вперед на сторінку з повідомленням
                // про помилку
                filter.getConfig().getServletContext()
.getRequestDispatcher("error.vm").
forward(srequest, sresponse);
    }
}
    }

```

```
/**
 * Before advice перевірки на читання
 */
before(ServletRequest request, Object object)
: readAccess(request, object){
    if (!AuthHelper.isAbleToRead
(extractUser(request), object)) {
        throw new AuthorizationException
("Read access not allowed");
    }
}

/**
 * Before advice перевірки на додавання
 */
before(ServletRequest request, Object object)
: addAccess(request, object){
    if (!AuthHelper.isAbleToAdd
(extractUser(request), object)) {
        throw new AuthorizationException
("Add access not allowed");
    }
}

/**
 * Before advice перевірки на видалення
 */
before(ServletRequest request, Album album)
: deleteAccess(request, album){
    if (!AuthHelper.isAbleToDelete
(extractUser(request), album)) {
        throw new AuthorizationException
("Delete access not allowed");
    }
}

/**
 * Around advice пред-перевірки, тут
ми ігноруємо повернене методом
* значення, і увесь час повертаємо
те яке задовольняє правилам
* авторизації
* Ми не обробляємо інші пред-перевірки
оскільки за умовчанням будь-хто може
* читати, і усе аутентифіковані
користувачі можуть додавати
*/
boolean around(ServletRequest request
Object object) : deleteCheck(request, object){
    return AuthHelper.isAbleToDelete
(extractUser(request), object);
}

/**
 * Приватна функція, яка виймає
користувача із запиту
*/
private AnonymousUser extractUser
(ServletRequest request) {
    return (AnonymousUser)((HttpServletRequest
request).getSession().getAttribute
(EntranceFilter.USER_KEY);
}
}
```

Висновки

Для ефективної роботи веб-систем необхідно :

1. Оцінити ризики на рівнях даних, систем і виконання.
2. Використовувати прогресивну аспектно-орієнтовану модель програмування.

1. Д.Н. Колисниченко. РНР 5 Самовчитель - Видання 3-і - Спб.: Наука і Техніка, 2006. - 576 с.
2. Ед Леки-Томпсон. РНР 5 для професіоналів. - М.: Вільямс, 2006.- 608с.

УДК 004.94

А.П. Здолбіцький

Луцький національний технічний університет

ОБЪКТНО-ОРИЕНТОВАНА ПРОГРАМНА РЕАЛІЗАЦІЯ МАТЕМАТИЧНОЇ МОДЕЛІ ДЕФОРМУВАННЯ ДЕРЕВИНИ У ПРОЦЕСІ НЕІЗОТЕРМІЧНОГО ВОЛОГОПЕРЕНЕСЕННЯ

А.П. Здолбіцький. Объектно-ориентированная программная реализация математической модели деформирования древесины в процессе неизо термического влагопереноса. В рамках объектно-ориентированного подхода разработано прикладное программное обеспечение для численной реализации математических моделей теплопереноса и пружновязкопластического деформирования древесины при сушке. Разработанный программный комплекс на языке программирования Java содержит информационную модель и интерфейс программной системы в виде пакетов классов и отношений между ними с использованием графических диаграмм UML, компоненты программного кода, вычислительные схемы реализации МСЕ. Разработанные классы отражают сущность объектно-ориентированной реализации метода конечных элементов. Это создает возможность интеграции разработанных программ существующих систем автоматизированного моделирования с целью расширения их функциональных возможностей.

Ключевые слова: диаграмма UML, влагоперенос, деформирования, моделирование.

Форм. 2. Рис. 11. Лит. 17.

А.П. Здолбіцький. Объектно-ориентированная программная реализация математической модели деформирования древесины в процессе неизо термического влагопереноса. В рамках объектно-ориентированного подхода разработано прикладное программное обеспечение для численной реализации математических моделей теплопереноса и пружновязкопластического деформирования древесины при сушке. Разработанный программный комплекс на языке программирования Java содержит информационную модель и интерфейс программной системы в виде пакетов классов и отношений между ними с использованием графических диаграмм UML, компоненты программного кода, вычислительные схемы реализации МСЕ. Разработанные классы отражают сущность объектно-ориентированной реализации метода конечных элементов. Это создает возможность интеграции разработанных программ существующих систем автоматизированного моделирования с целью расширения их функциональных возможностей.

Ключевые слова: диаграмма UML, влагоперенос, деформирования, моделирование.

A.P. Zdobitsky. Object-oriented software implementation of mathematical models of deformation of wood in non-isothermal moisture transfer process. Within an object-oriented approach developed application software for the numerical implementation of mathematical models and moisture transfer elastoviscoplastic deformation of wood during drying. The developed software package contains Java programming language and interface information model of software system in a package of classes and the relationships between them using graphical diagrams UML, component code, computational schemes of the Union. Developed classes reflect the essence of object-oriented implementation of the finite element method. This creates the possibility of integration with existing applications developed computer-aided modeling in order to extend their functionality.

Keywords: diagram UML, moisture transfer, deformation, modeling.

Аналіз досліджень. На теперішній час існує порівняно небагато публікацій застосування об'єктно-орієнтованого підходу для програмної реалізації МСЕ [1-4], в яких запроєктовані класи базових компонентів МСЕ (елементів, вузлів, граничних умов та навантажень). Їх аналіз дозволяє стверджувати про те, що безпосереднє створення об'єктів в програмному коді породжує необхідність перекомпілювання програми для нових даних. Потребує подальшого вдосконалення задача візуального створення та редагування об'єктів шляхом їх інтегрування з генератором розбиття геометричної області. У даній статті в рамках об'єктно-орієнтованого підходу розроблено алгоритмічне та програмне забезпечення для чисельної реалізації наведених у попередніх публікаціях [7,8] математичних моделей деформування деревини у процесі неізотермічного вологоперенесення.

Об'єктно-орієнтований підхід для програмної реалізації. Для побудови програмної моделі методу скінченних елементів необхідно здійснити об'єктно-орієнтований аналіз методу, виділити програмні моделі, які є самостійною сутністю об'єктно-орієнтованої реалізації алгоритмів МСЕ у контексті основних питань і термінів методу. З одного боку, програма методу скінченних елементів має бути визначена у термінах ООП – алгоритми, методи і дані (класи), а з іншого – основні поняття МСЕ мають бути формалізовані у контексті ООП.

Такий підхід дозволяє розкласти програмну систему у вигляді класів та відношень між ними. Пояснення зв'язків між спроектованими класами та взаємодію об'єктів цих класів здійснено за допомогою діаграм UML [5]. В окремі пакети виділено класи, які реалізують: геометричні та фізико-механічні характеристики об'єкта досліджень; розбиття області на скінченні елементи за допомогою сітки вузлів; визначення базисних функцій в межах скінченних елементів; обчислювальні класи (квadrатури для чисельного інтегрування); інтерполяційні функції; розв'язання систем лінійних алгебраїчних рівнянь (СЛАР); класи, орієнтовані на конкретні обчислення матричної та векторної алгебри; класи збереження, введення та виведення даних; інтерфейс користувача.

Структура простої об'єктно-орієнтованої програми методу скінченних елементів наведена на рисунку 1.

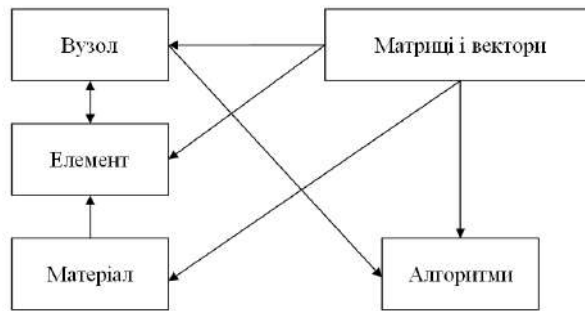


Рис. 1. Структура об'єктно-орієнтованої програми

Введемо наступні основні класи МСЕ: (*TUNode*) (Вузол), *Element* (Елемент), *Material* (Матеріал) і *Property* (Властивості). Нижче буде приведено опис їх функціонування у шаблоні ООП, їх основні атрибути і методи. Будуть визначені класи *List* і алгебраїчні класи (*Matrix*) Матриця і (*Vector*) Вектор.

Клас (*TUNode*) відіграє подвійну роль: з одного боку це точка спряження елементів (геометричний вузол), з іншого це інтерполяційний вузол, що відповідає за роботу з невідомими, розв'язуваної задачі, які називаються степенями свободи – *dof* і навантаженнями – *load*. Для того, щоб в глобальній системі число рівнянь дорівнювало числу невідомих необхідно в кожному вузлі задати один елемент з дуальної пари (*dof*, *load*). Ця дуальність впливає на вирішення глобальної системи рівнянь, тому що задане *dof* вводить невідоме початкове навантаження, тому глобальна змінна *dof* і глобальний вектор навантаження одночасно можуть містити невідомі значення. Пропонується зазначену дуальність враховувати на рівні методів вирішення так, щоб метод *Matrix_solve* одночасно обчислював невідомі складові степенів свободи і реакцій. *Node* повинен передавати задані *dof* і *load* у глобальну систему рівнянь і довізначати невідомі значення, отримані в результаті її вирішення.

Клас *Element* містить інформацію про розподілені параметри задачі. Його функції змінюються залежно від використання на попередній або завершальній фазі опрацювання інформації. В процесі попереднього опрацювання *Element* акумулює властивості, такі як геометрія та фізична поведінка, обираючи їх з поміж інших класів для подальшого опрацювання дискретних форм зображення вихідної неперервної задачі. Зазвичай, для стаціонарних задач – це матриці жорсткості елементів і вектори узагальненого навантаження, які повинні бути зібрані в глобальну систему рівнянь. Клас *Material* описує фізичні параметри моделі, що використовуються класом *Element*. Його задача полягає в керуванні параметрами таким чином, щоб *Element* в будь-який час був здатний обчислити елементи матриці жорсткості, тензор напруження, вектор зміщення і т. д. Клас *Property* вводить для визначення факторів, котрі формально не належать до опису матеріальних властивостей середовища (матеріалу). Зазвичай, це різноманітні параметри, що належать до виду об'єкта, що моделюється. До них можуть належати, наприклад, товщина, площа поперечного перерізу, моменти інерції, і т. д. На запит ці параметри передаються класу *Element*.

Принципова структура об'єктно-орієнтованої програми методу скінченних елементів наведена на рисунку 2.



Рис. 2. Структура об'єктно-орієнтованої програми методу скінченних елементів

МСЕ завжди призводить до розрідженої матриці – з великою кількістю нулів поза головною діагоналлю. Зберігання нульових елементів різко знизило б ефективність методу. Тому конкретна реалізація МСЕ зазвичай використовує спеціальну структуру зберігання матричних елементів, мінімізуючи або зовсім виключаючи з неї нульові елементи [6]. В даній програмі для зберігання індексів ненульових елементів використовується уявлення розрідженої матриці у вигляді графа.

Слід зазначити, що доцільно, щоб вузли і елементи сітки мали «близьку» нумерацію. При цьому значно прискорюється опрацювання масивів, пошук потрібних вузлів та елементів, суттєво підвищується ефективність роботи постпроцесора, особливо в разі нестачі оперативної пам'яті і використання файлів підкачки. Для цього вихідна звичайно-елементна сітка має бути піддана процедурі перенумерації (перевпорядкування) вузлів та елементів.

Спосіб зберігання зв'язаних даних. Алгоритм методу скінченних елементів передбачає, що скінченні елементи та базисні функції взаємопов'язані. При програмуванні цих двох сутностей як класів між ними можна поставити зв'язок узагальнення: скінченний елемент (TUElem) наслідуює порядок, що зберігається у класі базисних функцій (TUInterpolation) (рис. 3).

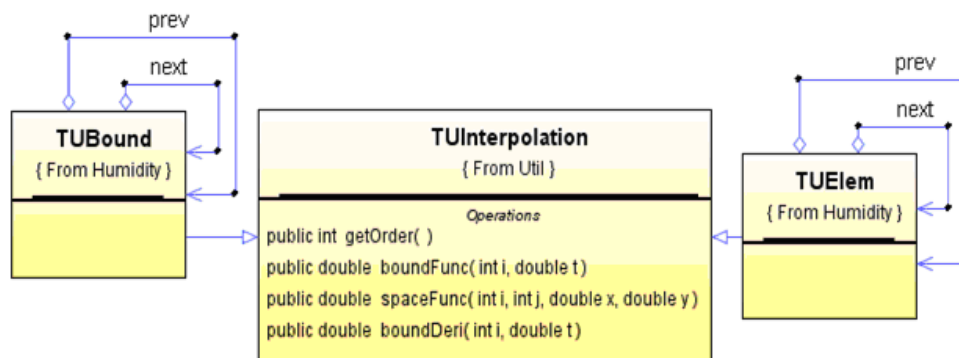


Рис. 3. Зв'язок узагальнення між базисною функцією та скінченним елементом

Скінченні елементи – це основа для обчислення подвійних інтегралів, що входять у варіаційні формулювання задачі тепломасоперенесення та задачі визначення напружено-деформівного стану [7,8]. Для обчислення одинарних інтегралів по границі області, що входять у співвідношення, визначений клас граничний елемент (TUBound), який також наслідуює клас базисних функцій (TUInterpolation) (рис. 3).

Скінченні і граничні елементи будуються на певній сітці вузлів, якою покривається область задачі. Постає питання, як реалізувати зв'язок між елементом та вузлами, на яких він побудований. Відомі програмні реалізації методу скінченних елементів [4] передбачають використання масивів різних розмірностей. Зокрема перший масив зберігав усі вузли (їхні номери, координати, тощо), другий встановлював відповідність номерів скінченних елементів номерам вузлів, що їм належать, третій зберігав номери вузлів, що знаходяться на границі. Це вимагало значних зусиль при написанні та відлагодженні програмного коду.

В даній програмній реалізації МСЕ для збереження усієї множини вузлів, які накладаються на область задачі, використано двозв'язний замкнений список. Кожен вузол (*TUNode*) містить вказівник на попередній та на наступний вузли. Крім того шукані вузлові значення виділено в окрему структуру (*TUValues*). Для задачі тепломасоперенесення у неї входять температура та вологість у певний момент часу. Таким чином, кожен вузол (*TUNode*) – це точка (*Point*) з певними координатами, який містить вказівники на попередній та наступний вузли та масив структур вузлових значень (*TUValue*) для різних моментів часу (рис.4).

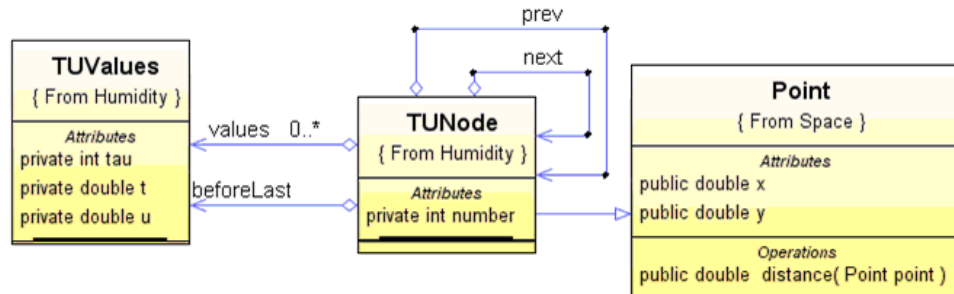


Рис. 4. Відношення між класами, що зберігають інформацію про вузли і вузлові значення

Всі об'єкти МСЕ-моделі зберігаються у зв'язаному списку, котрий визначається шаблоном, класом *List*. Саме там задаються підписки *Node_List*, *Element_List*, *Material_List* і *Propert_List*. В них даними є адреси об'єктів визначеного типу, а методи, що наприклад, належать до класу *Element*, застосовуються до записів підписку *Element_list*, що дає можливість у зв'язаному списку *List* імітувати традиційні методи циклічної обробки типу *for – do*, *while – do*, і *for – in*, що є стилем, який покращує зручність читання програми, наприклад, в модулі відповідальному за обчислення і формування глобальної матриці жорсткості. Її асемблування здійснюється циклом по всіх об'єктах (*elem*) класу *Element*, що перебуває в списку *Element_List* – (*ellist*), що може бути представлено на псевдокоді наступним способом:

```

For elem from ellist.start() to ellist.end do
    elem.assm_stiffness(K)
od,

```

де *start* повертає перший запис в списку. Символ *ellist.end* не є в дійсності методом, він вводиться лише для формалізації запису.

Розроблення структури класів та діаграми класів. Приступаючи до побудови архітектури програмної системи на мові *Java* з використанням середовища *IDE Eclipse Helios* необхідно визначити основні абстракції.

Точка входу в програму знаходиться в класі *Main*, в якому знаходиться функція *main()*. Звідси запускається вікно для введення даних – фрейм *InputFrame*, який успадковується від системного класу-вікна *JFrame* (рис. 5). На формі знаходяться поля для введення даних, які представлені спеціально розробленим класом *DoubleTextField*, для введення числових даних. Клас успадкований від *JTextField*. Основним методом в цьому класі є *getValue()*, який повертає введені значення типу *double*. Для того щоб компонент працював тільки з числовими значеннями переважано метод *createDefaultModel()*, який повертає об'єкт типу *DoubleTextDocument* [101, 124].

Після завершення введення даних відбувається ініціалізація класу для їх зберігання – *InputDataHolder* – це синглтон, який створюється тільки один раз протягом одного сеансу використання програми і забезпечує доступ до вхідних даних в будь-якій точці. За створення цього класу відповідає метод *getInstance()*:

```

private static InputDataHolder instance = null;
public static InputDataHolder getInstance()
{
    if(instance == null){
        synchronized (InputDataHolder.class)
        {
            if(instance == null){
                instance = new InputDataHolder();
            }
        }
    }
}

```



```

    }
  }
  return instance;
}

```

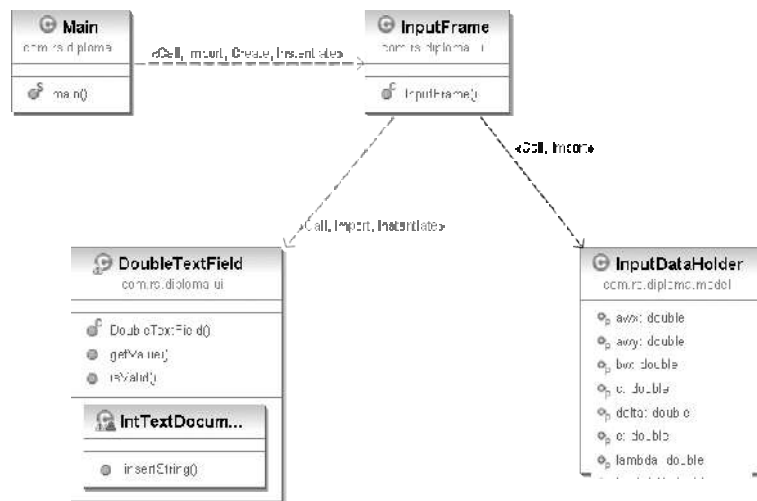


Рис. 5. Діаграма класів "Введення даних"

Розпочнемо декомпозицію системи з виділення абстракцій, які відповідатимуть за паралельні обчислення, це – об'єкт, який відповідатиме за триангуляцію (*Triangulator*), потік для обчислень значень температури (*TFECalculator*), потік для обчислень вологості (*WFECalculator*).

Окремий потік (*Thread*) представляє собою клас для здійснення триангуляції – розбиття на трикутники – *Triangulator* (рис. 6). Він успадковується від системного класу *Thread* і реалізовує інтерфейс *Runnable*. Відповідно до концепції багатопотоковості в *Java* основні операції виконуються в методі *run()*. Для доступу до початкових даних тут використовується клас-сінглетон *InputDataHolder*, в якому зберігаються вхідні дані програми. Для сповіщення інших об'єктів в програмі про закінчення процесу розбиття використовується інтерфейс *OnTriangulationListener*. В ньому викликається метод *onTriangulationFinished()* відразу після закінчення процесу.

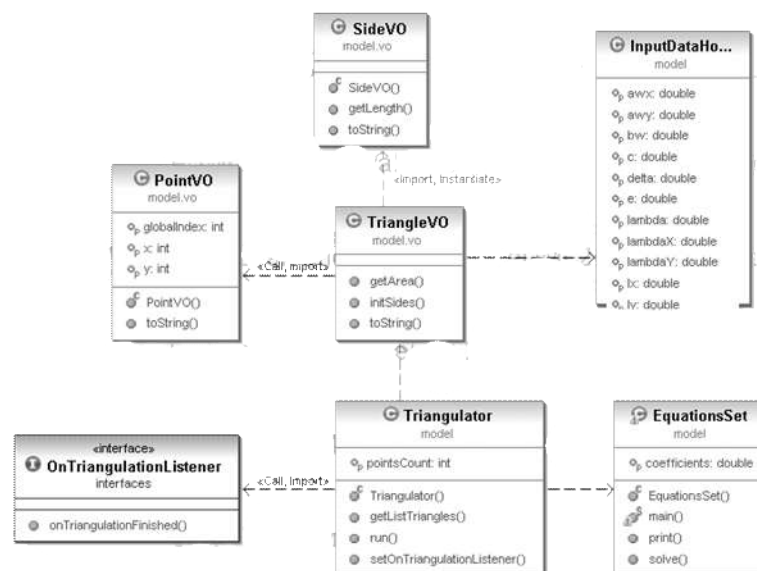


Рис. 6. Діаграма класів "Триангуляція"

Для представлення простих абстракцій, таких як трикутник, точка, сторона та інших використано поняття об'єкта-значення (*value object*). Найпростішим з цих об'єктів є точка – *PointVO*, яка має дві координати (*x*, *y*) та глобальний індекс. Об'єкт *SideVO* описує одну сторону трикутника і будується на основі двох точок і має метод для обчислення довжини сторони.

Трикутник (*TriangleVO*) використовується в моделі для логічного представлення трикутника в програмі. Він, відповідно, складається з трьох точок і трьох сторін, також має метод для обчислення площі. Тут сторони обчислюються при початковій ініціалізації самого трикутника і ніде в програмі не перераховуються для збільшення швидкодії системи.

Клас *EquationSet* використовується для розв'язання системи рівнянь отриманих комбінуванням результатів триангуляції і вхідних параметрів. Основним методом тут є *solve()*, який фактично є проксі-методом, що переадресовує запит до рекурсивної функції розв'язання системи, а потім повертає результат. Так зроблено для зручності введення рекурсії. Метод *solve()* звертається до методу *simplify()*, який спочатку спрощує систему, а потім виконує основні рекурсивні обчислення.

Обчислення температури і вологості матеріалу відбувається в двох основних класах – *WFETriangulator* (вологість) і *TFETriangulator* (температура) (рис. 7). Ці класи є потоками, які виконуються паралельно і взаємодіють між собою після кожної ітерації. Всі операції виконуються в методі *run()*, який включений в інтерфейс *Runnable*. Як глобальні члени класу винесені матриці *C*, *K*, і *F*. Результуючу матрицю можна дістати викликавши метод *getResultMatrix()*, який повертає масив результатів. Оскільки класи виконують велику кількість операцій з матрицями, то найпоширеніші з них доцільно було винести у клас-утиліту *MatrixUtils*, який і використовується при обчисленнях. Вхідні дані для обчислень потоки беруть в класі *InputDataHolder*.

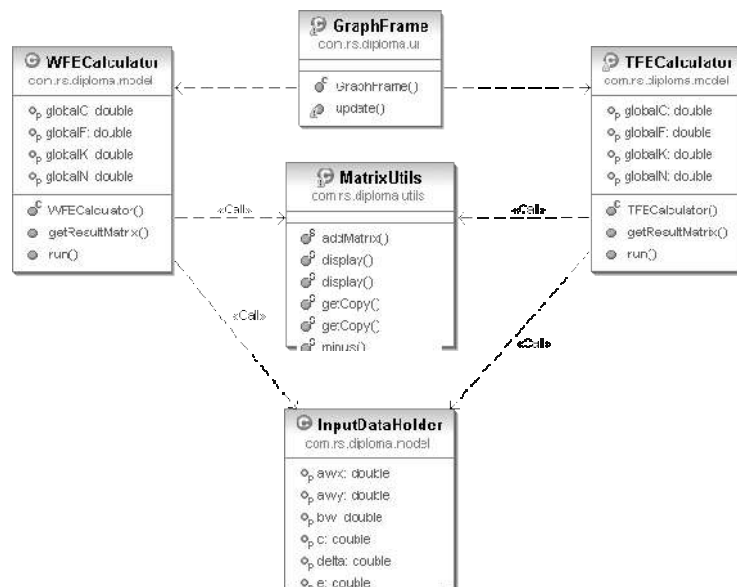


Рис. 7. Діаграма класів "Обчислення температури і вологості"

Спілкування між двома потоками забезпечує механізм синхронізації в *Java* заснований на методах *wait*, *notify* і *notifyAll*. Ці методи реалізовані, як *final*-методи класу *Object*, так що вони є в будь-якому *Java*-класі. Всі ці методи повинні викликатися тільки з синхронізованих методів.

Потік, який чекає виконання будь-яких умов, викликає у цього об'єкта метод *wait*, попередньо захопивши його монітор. На цьому його робота припиняється. Інший потік може викликати на цьому ж самому об'єкті метод *notify* (знову ж, попередньо захопивши монітор об'єкта), в результаті чого, чекає на об'єкті потік "прокидається" і продовжує своє виконання.

Об'єктом для відображення результатів розрахунків служить вікно *GraphFrame*, яке успадковане від системного класу *JFrame* (рис. 8). На його панелі можна побачити відображений зразок розбитий на трикутники.

В програмі можна виводити графіки двох типів: трьохвимірні і двохвимірні. Ця можливість реалізована з допомогою *Java* бібліотек: *jzy3D* і *JFreeChart* з відкритим кодом. Всі вікна для відображення графіків викликаються з класу *GraphFrame*.

JFreeChart є бібліотекою з відкритим кодом і розповсюджується під ліцензією *GNU Lesser General Public Licence (LGPL)*, що дозволяє використовувати продукт в усіх додатках і включає такі функції: добре документований *API*, який підтримує велику кількість типів графіків; гнучкий дизайн, який просто впровадити; підтримка багатьох вихідних форматів, включаючи *Swing*-

компоненти, графічні файли (включаючи PNG і JPEG) формати векторної графіки (включаючи PDF, EPS і SVG).

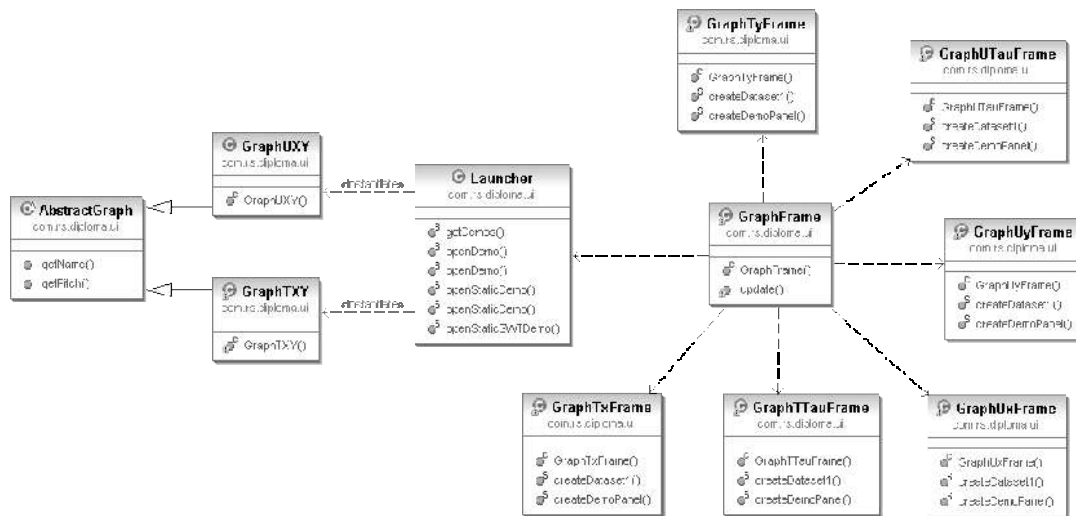


Рис. 8. Діаграма класів "Відображення графіків"

Для запуску вікон з трьохвимірних графіків в програмі використовується клас *Launcher*. Запуск нового вікна здійснюється зі статичного методу *openDemo()* в який потрібно передати об'єкт, що реалізує інтерфейс *IFrame*. З його допомогою можна запустити вікна *GraphUXY* і *GraphTXY*, які попередньо мають бути визначені в методі *getDemos()*:

```
public static List<IFrame> getDemos() throws IOException
{
    List<IFrame> demos = new ArrayList<IFrame>();
    demos.add(new GraphUXY());
    demos.add(new GraphTXY());
    return demos;
}
```

З метою узагальнення, класи *GraphUXY* і *GraphTXY* успадковуються від класу *AbstractGraph*, який реалізує інтерфейс *IFrame*. В цих класах всі дані для побудови графіків формуються в класі *getCoordinates()*, який повертає список (*List*).

Для формування двовимірних графіків використовується базовий клас *ApplicationFrame*, який наслідують всі класи, що відповідають за відображення графіків. Набір даних для відображення тут формується в методі *createDataset()*.

Дослідження адекватності та верифікації математичних моделей. Для дослідження адекватності математичної моделі тепломасоперенесення [7,8] проведені порівняння результатів чисельного моделювання температурно-вологісних полів з незалежними теоретичними дослідженнями для часткових випадків, а також здійснено порівняння результатів моделювання з відомим експериментальними дослідженнями, виконаними для конкретних умов. В основі таких досліджень покладені окремі математичні моделі перенесення тепла і вологи зі сталими коефіцієнтами. Експериментальні дослідження для деревних взірців таких розмірів деревини бука наведені у працях [9,10]. Згідно них, в процесі експерименту зовнішні умови агента сушіння приймалися такими: $t_c = 80 \text{ }^\circ\text{C}$, $\varphi = 60 \%$, $v = 1 \text{ м/с}$; $t_c = 70 \text{ }^\circ\text{C}$, $\varphi = 50 \%$, $v = 2 \text{ м/с}$; $t_c = 50 \text{ }^\circ\text{C}$, $\varphi = 40 \%$, $v = 3 \text{ м/с}$. Відповідно, для першого випадку зміни (t_c, φ, v) теплофізичні параметри матеріалу є такі: $\rho = 734 \text{ кг/м}^3$; $C = 2,42 \text{ кДж/(кг}\cdot\text{K)}$; коефіцієнти вологопровідності $a_{mx} = 0,0106 \text{ см}^2/\text{с}$; $a_{my} = 0,0148 \text{ см}^2/\text{с}$; $\varepsilon = 0,15$; критерій Косовича $Ko = 5,05$; критерії Ликова $Lu_x = 0,0109$; $Lu_y = 0,0114$; критерії Кірпічова $Ki_{ix} = 0,0697$; $Ki_{iy} = 0,026$; $Ki_{ix} = 0,0077$; $Ki_{iy} = 0,00276$; критерій Поснова $Pn = 59,02$. Для розрахунків у другому випадку зміни параметрів середовища (t_c, φ, v) теплофізичні параметри вибиралися: $\rho = 718 \text{ кг/м}^3$; $C = 2,28 \text{ кДж/(кг}\cdot\text{K)}$; $a_{mx} = 0,016 \text{ см}^2/\text{с}$; $a_{my} = 0,022 \text{ см}^2/\text{с}$; $\varepsilon = 0,15$; $Ko = 5,285$; $Lu_x = 0,0904$; $Lu_y = 0,1080$; $Ki_{ix} = 0,736$; $Ki_{iy} = 0,309$; $Ki_{ix} = 0,00567$; $Ki_{iy} = 0,00206$; $Pn = 56,43$.

На рис. 9 та рис. 10 наведені результати порівняння розподілу температурно-вологісних полів з експериментальними дослідженнями [10,11] та даними чисельного експерименту з

математичними моделями зі сталими теплофізичними параметрами [12]. За критерій розходження між результатами використаний коефіцієнт множинної кореляції [13], який використовується для порівняння розрахункових та експериментальних даних перехідних процесів. Даний критерій визначає, чи існує кореляційний зв'язок між експериментальними та розрахунковими даними та має вигляд:

$$|t| = \frac{|\rho_{yy^-}|}{\sqrt{1 - \rho_{yy^-}^2}} \sqrt{n - 2}, \quad (1)$$

де n – кількість точок, для яких проводить порівняння величин. Даний критерій дозволяє встановити ступінь кореляційного зв'язку між величинами. Якщо виконується умова $|t| < t_\alpha$, то з ймовірністю $p = 1 - \alpha$ можна стверджувати, що між величинами σ_{ij} та $\bar{\sigma}_{ij}$ існує «тісний» кореляційний зв'язок. У протилежному випадку з ймовірністю $p = 1 - \alpha$ гіпотеза про такий зв'язок відкидається.

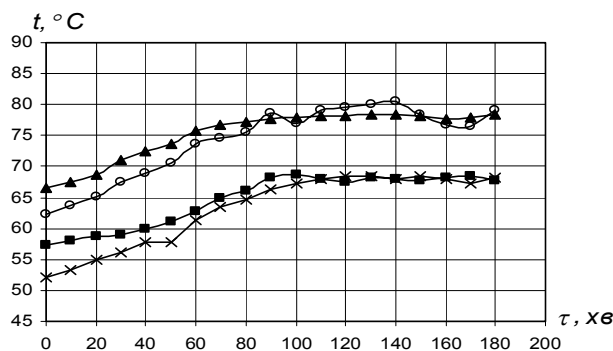


Рис. 9. Залежності розподілу температури у часі для $t_c = 70$ °C, $\varphi = 50$ %, $v = 2$ м/с (■, ▲ – експериментальні дані [11,12]; х, ○ – розрахункові значення за математичною моделлю)

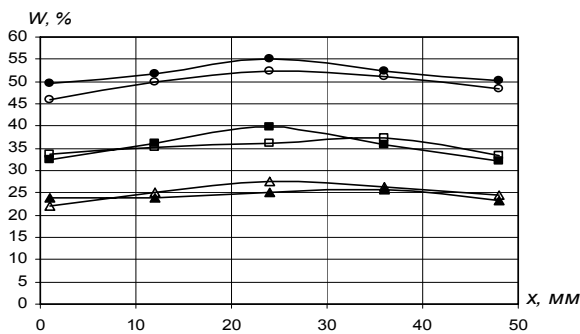


Рис. 10. Залежності розподілу вологості за товщиною пластини для $t_c = 70$ °C, $\varphi = 50$ %, $v = 2$ м/с (●, ■, ▲ – експериментальні дані [11,12] для $\tau = 30$ хв, $\tau = 120$ хв, $\tau = 120$ хв; ○, □, Δ – розрахункові значення за математичною моделлю)

Коефіцієнт кореляції ρ_{yy^-} між розрахунковими (y) та експериментальними (\bar{y}) величинами визначається за відношенням:

$$\rho_{yy^-} = \frac{\sum_{i=1}^n (y_i - y_{сеп}) (\bar{y}_i - \bar{y}_{сеп})}{\sqrt{\sum_{i=1}^n (y_i - y_{сеп})^2 \sum_{i=1}^n (\bar{y}_i - \bar{y}_{сеп})^2}}, \quad (2)$$

$$\text{де } y_{сеп} = \frac{1}{n} \sum_{i=1}^n y_i; \quad \bar{y}_{сеп} = \frac{1}{n} \sum_{i=1}^n \bar{y}_i.$$

Для верифікації математичної моделі використовувалися експериментальні дані [9,14]. Зокрема отримано, що для ($t_c = 70$ °C, $\varphi = 50$ %, $v = 2$ м/с; $\rho = 720$ кг/м³) значення критерія і

коефіцієнта кореляції для розподілу температури такі: $\rho_{yy} = 0,993$; $|t| = 11,58$; для ($t_c = 80$ °С, $\varphi = 60$ %, $v = 1$ м/с; $\rho = 735$ кг/м³) маємо $\rho_{yy} = 0,995$; $|t| = 14,083$. Аналогічно для ($t_c = 70$ °С, $\varphi = 50$ %, $v = 2$ м/с; $\rho = 720$ кг/м³) значення критерія і коефіцієнта кореляції для розподілу вологості за товщиною дошки для моменту часу $\tau = 30$ хв такі: $\rho_{yy} = 0,992$; $|t| = 11,48$; для моменту часу $\tau = 120$ хв: $\rho_{yy} = 0,998$; $|t| = 22,28$.

Затабульоване t для температурного поля для $n = 17$, числі степеней вільності $f = n - 2 = 15$ і рівні значимості складає $t_{маб} = 4,48$. Оскільки умова $|t| \geq t_{маб}$ виконується для двох випадків, то з ймовірністю похибки $\alpha = 0,05$ (5%) можна стверджувати про адекватність характеру зміни наведених даних результатів чисельного моделювання температурних полів та їх експериментальних значень. Аналогічно для розподілу вологості ($n = 5, f = 3$) на рівні значимості $\alpha = 0,05$ маємо $t_{маб} = 4,27$. Оскільки, для двох значень часу ($\tau = 30$ хв, $\tau = 120$ хв) виконується умова $|t| \geq t_{маб}$, то з ймовірністю похибки можна стверджувати про адекватність чисельного моделювання вологісних полів з результатами експериментальних досліджень.

Тому перевіримо адекватність моделювання напружено-деформівного стану.

Порівняємо результати моделювання розвитку напружень у пружній та в'язкопружній областях.

У роботі [15,16] для різних моментів часу обчислено розподіл напружень у пружній області деформування у процесі сушіння. Обчислення здійснено для соснових дошок товщиною 20 см при таких параметрах середовища сушіння: температура $T_c = 88$ °С, відносна вологість $\varphi = 55$ %, швидкість руху агента сушіння $\varphi = 55$ м/с. Із порівняння розподілів напружень у центрі вказаного матеріалу вздовж його товщини легко бачити, що характер напружень однаковий (рис. 11). Щодо відхилення отриманих чисельних результатів від наведених у [15,16], то максимальне відхилення становить не більше 7,9 % після перших трьох годин сушіння і спадає у процесі сушіння.

Інший спосіб показати адекватність результатів – порівняти чисельні результати в'язкопружного деформування деревини, розраховані для двовимірної задачі, із такими ж результатами, обчисленими іншими авторами для одновимірної задачі. Наприклад, у роботі [17] наведені результати розвитку напружень у деревині сосни розміром 20 мм під час сушіння при таких параметрах середовища сушіння: температура $T_c = 88$ °С, відносна вологість $\varphi = 55$ %, швидкість руху агента сушіння $v = 2$ м/с. Із рис. 11 видно, що характер розвитку напружень однаковий, відхилення у перші години процесу сушіння досить значне, але не перевищує 12,4 %, і у процесі сушіння спадає.

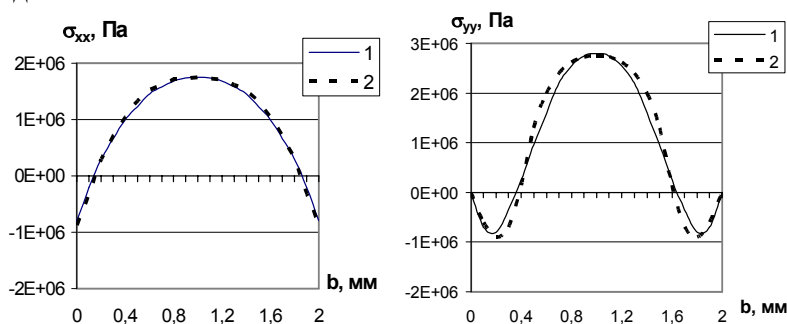


Рис. 11. Розподіл напружень (а) і (б) в центрі матеріалу вздовж товщини деревини сосни розміром 20×20 мм через 2,6 год процесу сушіння за методами: 1 – Рітца; 2 – МСЕ

Висновки. Розроблено прикладну програму чисельного розв'язування двовимірної математичної моделі. При програмуванні використано об'єктно-орієнтований підхід, який дозволив реалізувати програму у вигляді множини об'єктів та класів, що взаємодіють між собою. Такі об'єкти як скінченний елемент, вузол, квадратура, апроксимаційна функція, СЛАР, тощо були реалізовані у вигляді окремих класів.

Описано за допомогою діаграм UML основні етапи роботи прикладної програми. Зокрема, на діаграмах класів представлено основні класи, які реалізують роботу програми, відношення та

зв'язки між ними. На діаграмах послідовності відображено впорядковану в часі взаємодію об'єктів під час виконання різних етапів обчислень.

Здійснено апробацію та верифікацію результатів роботи програми шляхом їхнього порівняння із наявними експериментальними даними та порівняння часткових випадків із відомими результатами.

Розроблене програмне забезпечення мовою Java і спроектовані діаграми UML та алгоритми реалізації методу скінченних елементів можуть бути використані для проектування систем автоматизованого моделювання та аналізу температурно-вологісних полів у капілярно-пористих матеріалах залежно від характеристик деревини.

ЛІТЕРАТУРА

1. **Конисов С.П.** Объектно-ориентированный метод декомпозиции области / С.П. Конисов, И.В. Красноперов, В.Н. Рычков // Вычислительные методы и программирование. – 2003. – Т.4. – №1. – С. 176-193.
2. **Савула Я.Г.** Числовий аналіз задач математичної фізики варіаційними методами. – Львів: Вид-во ЛНУ «ім.Франка». – 2004. – 222 с.
3. **Соколовський Я.І.** Автоматизація процесів моделювання деформаційно-релаксаційних і тепломасообмінних процесів у капілярно-пористих матеріалах / Я.І. Соколовський, О.В. Мокрицька, І.М. Крошній, І.Д. Капран, А.П. Здолбіцький // Системний аналіз та інформаційні технології: матеріали Міжнародної науково-технічної конференції SAIT 2011. – Київ: ННК «ІПСА» НТУУ «КПІ». – 2011. – С. 159.
4. **Zimmermann T.** Object-orient finite element programming: I vgovernment principles / T. Zimmermann, Y. Duboes-Pelerin, P. Bomme // Computing Methods in Applied Mechanics and Engeeneeing. – 1992. – Vol. 98 (no. 2). – P. 291-303.
5. **Rumbaugh J.** The unified modeling language: user guide reading / J. Rumbaugh, G. Booch // Addison-Weslyy. – 1999. – 346 p.
6. **Сегерлинд Л.** Применение метода конечных элементов. – М.: Мир. – 1979. – 378 с.
7. Ya.I. Sokolowskyu, M.B. Denjuk, A.B. Bakalets, A.P. Zdolbitskyu. Mathematical modeling of stressed-strained relaxation fields in wood drying process. // Наукові нотатки. Міжвузівський збірник (за галузями знань „Машинобудування та металообробка”, „Інженерна механіка”, „Металургія та матеріалознавство”). – Луцьк, 2010. – Вип. 27. – С. 4-10.
8. **Соколовський Я.І.** Математична модель деформаційно-релаксаційних процесів у капілярно-пористих матеріалах з параметрами внутрішнього і зовнішнього тепломасоперенесення / Я.І. Соколовський, І.М. Крошній // Вісник Національного університету “Львівська політехніка”: Комп'ютерні науки та інформаційні технології. – Львів: НУ “Львівська політехніка”. – 2011. – Вип. 710. – С. 274-279.
9. **Шубин Г.С.** Сушка и тепловая обработка древесины. – М.: Лесная промышленность. – 1990. – 336 с.
10. **Perre P.** A physical and mechanical model able to predict the stress field in wood over a wide range of drying conditions / P. Perre, J. Passard // Drying Technology. – 2004. – Vol. 22 (no. 1-2). – P. 27-44.
11. **Salin J.-G.** Numerical prediction of checking during timber drying and a new mechano-soorptive creep model // Holz Roh- Werkstoff. – 1992. – Vol. 50. – P. 195-200.
12. **Hakan F.** Oztop. Numerical and experimental analysis of moisture transfer for cinvective drying of someproducts / F. Oztop Hakan, K. Akpinar Ebru // International Communications in Heat and Mass Transfer. – 2008. – Vol. 35 (Issue 2). – P. 169-177.
13. **Томашевский В.** Имитационное моделирование в среде GPSS / В. Томашевский, Е. Жданова. – М.: Бестселлер. – 2003. – 416 с.
14. **Pang S.** Modeling of stress development during and relief during steaming in Pinus radiate lumber // Drying Technology. – 2000. – Vol. 18 (no. 8). – P. 1677-1696.
15. **Уголев Б.Н.** Деформативность древесины и напряжения при сушке. – М.: Лесная промышленность. – 1971. – 174 с.
16. **Уголев Б.Н.** Древесиноведение с основами лесного товароведения: учеб. для лесотехн. вузов / Б.Н. Уголев // М-во образования Рос. Федерации, Моск. гос. ун-т леса. – Изд. 3-е, перераб. и доп. – М.: МГУЛ. – 2002. – 340 с.
17. **Соколовський Я.І.** Технологічні напруження і деформації деревини у процесі сушіння // Науковий вісник: Зб. наук.-техн. праць. – Львів: УкрДЛТУ. – 1999. – Вип.9.3. – С. 168-176.

УДК 004.31

А.П. Здолбіцький, В.В. Місюк

Луцький національний технічний університет

СИСТЕМА НА КРИСТАЛІ AMBER ARM В БАЗИСІ ПЛІС

Здолбіцький А.П., Місюк В.В. Система на кристалі Amber ARM в базисі ПЛІС. У рамках підходу до розробки систем на кристалі реалізовано RISC-ядро з набором інструкцій ARMv2 на базі програмованої логічної схеми Cyclone III фірми Altera.

Ключові слова: FPGA, система на кристалі (СНК), програмована логічна схема.

Форм. 0. Рис. 2. Літ. 10.

А.П. Здолбіцький, В.В. Мисюк. Система на кристалі AMBER ARM в базисі ПЛІС. В рамках підходу до розробки систем на кристалі реалізовано RISC-ядро з набором інструкцій ARMv2 на базі програмуваної логічної схеми Cyclone III фірми Altera.

Ключевые слова: FPGA, система на кристалле (СНК), программируемая логическая схема.

A.P. Zdolbitskiy, V.V. Misyuk. System on Chip AMBER ARM in the base FPGA. As part of the approach to the development of systems-on-chip implemented with RISC-core instruction set ARMv2 based on programmable logic circuit Cyclone III company Altera.

Keywords: FPGA, system-on-chip (SoC), programmable logic circuit.

Аналіз систем та порівняльна характеристика FPGA. Програмовані логічні інтегральні схеми (ПЛІС – Programmable Logic Devices – PLD) являють собою нову елементну базу, яка володіє гнучкістю замовних ВІС і доступністю традиційної «жорсткої» логіки [3,10]. Головною відмінною властивістю ПЛІС є можливість їх налаштування на виконання заданих функцій самим користувачем. Сучасні ПЛІС характеризуються низькою вартістю, високою швидкістю (до 3,5 нс), значними функціональними можливостями (одна ПЛІС може замінити кілька сотень корпусів традиційної «жорсткої» логіки), багатократністю перепрограмування, низькою споживаною потужністю (що дозволяє використовувати їх у виробі з живленням від батарейок), гнучкістю архітектури та ін.

Підтвердженням перспективності нової елементної бази служить щорічна поява нових поколінь ПЛІС, що мають більш досконалу архітектуру, а також постійно зростаючий обсяг випуску ПЛІС провідними зарубіжними виробниками мікросхем: Advanced Micro Devices (AMD), Altera, Xilinx, Atmel, Intel, Texas Instruments та ін [3,10].

Процес проектування цифрового пристрою на основі ПЛІС полягає в описі його функціонування на вхідній мові використовуваного програмного засобу, виконанні автоматизованого синтезу, проведенні моделювання та налаштування обраної ПЛІС за допомогою програматора. При цьому час розробки навіть досить складних проектів може становити всього кілька годин. Для того щоб змінити алгоритм роботи пристрою, достатньо перепрограмувати ПЛІС.

Розробка пристроїв на основі ПЛІС являє собою нову технологію проектування електронних схем, включаючи їх виготовлення і супроводження.

Абревіатура FPGA означає Field Programmable Gate Array – програмована користувачем вентильна матриця. Програмована користувачем вентильна матриця, ПКВМ (англ. Field-Programmable Gate Array, FPGA) – напівпровідниковий пристрій, що може бути налаштований виробником або розробником після виготовлення; звідси назва: «програмується користувачем». ПКВМ програмується шляхом зміни логіки роботи принципової схеми, наприклад, за допомогою вихідного коду мовою проектування (типу VHDL), на якому можна описати цю логіку роботи мікросхеми. ПКВМ є однією з архітектурних різновидів програмованих логічних інтегральних схем (ПЛІС).

ПКВМ можуть бути модифіковані практично в будь-який момент у процесі їх використання. Вони складаються з конфігурованих логічних блоків, подібних перемикачів з безліччю входів і одним виходом (логічні вентиля або gates). У цифрових схемах такі перемикачі реалізують базові виконавчі операції AND, NAND, OR, NOR і XOR. У більшості сучасних мікропроцесорів функції логічних блоків фіксовані і не можуть модифікуватися. Принципова відмінність ПКВМ полягає в тому, що і функції блоків, і конфігурація з'єднань між ними можуть змінюватися за допомогою спеціальних сигналів, що посилюються схемою. У деяких спеціалізованих інтегральних схемах (ASIC) використовуються логічні матриці, аналогічні ПКВМ за структурою, однак вони конфігуруються один раз в процесі виробництва, у той час як ПКВМ можуть постійно перепрограмувати і міняти топологію з'єднань в процесі використання. Однак, така гнучкість вимагає істотного збільшення кількості транзисторів мікросхеми.

ПКВМ включають в себе три головних програмованих елементи: незкомутовані програмовані логічні блоки (ПЛБ), блоки вводу-виводу (БВВ) і внутрішні зв'язки. ПЛБ є функціональними елементами для побудови логіки користувача, БВВ забезпечують зв'язок між контактами корпусу і внутрішніми сигнальними лініями. Програмовані ресурси внутрішніх зв'язків забезпечують управління шляхами сполучення входів і виходів ПЛБ і блоків вводу-виводу на відповідні мережі. Всі канали трасування мають однакову ширину (однакова кількість провідників). Більшість блоків БВВ вписуються або в один рядок (по висоті), або в один стовпець (по ширині) масиву вентилів.

Логічний блок класичної ПКВМ складається з таблиці пошуку на 4 входи і тригера. В останні роки виробники почали переходити на таблиці пошуку з 6 входами у високопродуктивних частинах схеми, пояснюючи це необхідністю підвищення продуктивності.

Логічний блок має таблицю пошуку на 4 входи і вхід синхронізації (clock). Вихід блоку тільки один, це може бути регістрова або нерегістрова вихідна таблиця пошуку. Оскільки сигнали синхронізації в комерційних ПКВМ (а часто й інші сигнали, розпаралелюючих на велику кількість входів – high-fanout signals) трасуються особливим чином спеціальними трасувальними ланцюгами, управління цими сигналами робиться окремо.

Входи розташовані на окремих сторонах логічного блоку, вихідний контакт може трасуватися у двох каналах: або праворуч від блоку, або знизу. Вихідні контакти кожного логічного блоку можуть з'єднуватися з трасувань сегментами в суміжних каналах. Аналогічно, контактний майданчик блоку вводу-виводу (pad) може з'єднуватися з трасуванням елементом у будь-якому суміжному каналі. Наприклад, верхній контактний майданчик чіпа може з'єднуватися з будь-яким з W провідників (де W – ширина каналу) в горизонтальному каналі, розташованому безпосередньо під ним.

Як правило, трасування ПКВМ несегментоване, тобто кожен сегмент провідника з'єднує лише один логічний блок з перемикаючим блоком. Через обхід програмованих перемикачів у блоці перемикача трасування виходить більш довгим. Для збільшення швидкості внутрішньосистемних з'єднань, в деяких архітектурах ПКВМ використовуються більш довгі трасувальні з'єднання між логічними блоками.

У місці перетину вертикальних і горизонтальних каналів створюються перемикаючі блоки. При такій архітектурі для кожного провідника, що входить у перемикальний блок, існують три програмованих перемикача, які дозволяють йому підключатися до трьох інших провідників в суміжних сегментах каналу. Модель чи топологія вимикачів, використовувана в цій архітектурі, є планарною або доменною топологією перемикальних блоків. У цій топології провідник траси номер один підключається тільки до провідника траси номер один у суміжних каналах, провідник траси номер 2 підключається тільки до провідників траси номер 2 і так далі.

Сучасні сімейства ПКВМ розширюють перераховані вище можливості і включають вбудовану функціональність високого рівня. Маючи ці загальні функції, вбудовані в кремній, можна скоротити площу кристала, до того ж ці функції будуть працювати швидше, ніж якщо їх створювати на базі примітивів. До таких засобів належать мультиплексори, блоки цифрової обробки сигналів, вбудовані процесори, швидка логіка вводу-виводу і вбудована пам'ять.

ПКВМ також широко застосовуються для систем перевірки придатності, у тому числі в докремнієвій і післякремнієвій перевірці придатності, а також при розробці вбудованих програм. Це дозволяє компаніям-виробникам інтегральних схем перевіряти працездатність своїх пристроїв до виготовлення їх на заводі, скорочуючи час виходу виробу на ринок.

Загальну структуру кристала FPGA-мікросхеми наведено на рис. 1.2. По периферії верхнього шару кристала розміщуються блоки вводу/виводу (БВВ), що можуть бути запрограмованими для виконання функцій буферів: вхідного, вихідного, із запам'ятовуванням стану та ін. У деяких серіях FPGA-ІМС рівень напруги на двох БВВ може відрізнитись, що дає змогу поєднувати різні за рівнем живлення інтерфейси.

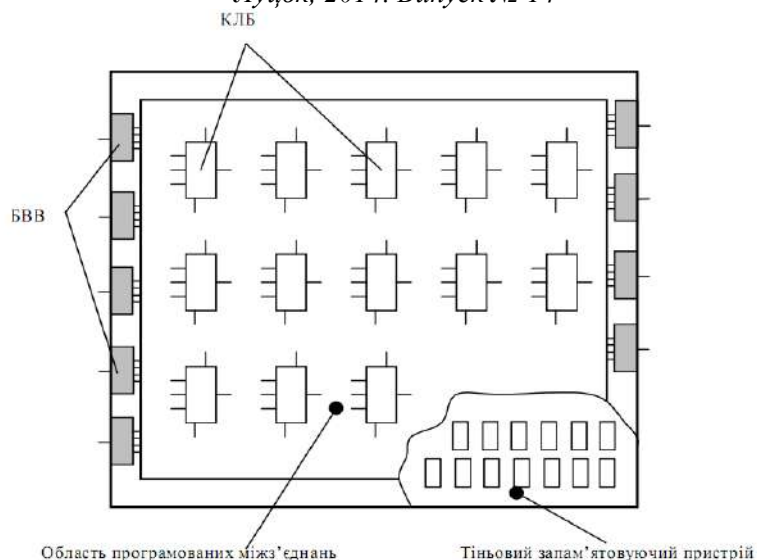


Рис. 1. – Узагальнена структура FPGA-мікросхем

У центрі кристала у вигляді матриці розміщено конфігуровані логічні блоки (КЛБ). Швидкодія мікросхем визначається часовою затримкою “вхід-вихід” одного КЛБ. Структура КЛБ залежить від серії мікросхем, так, наприклад, в ІМС XC2000 кожен КЛБ має один елемент пам’яті (тригер), 2 виходи, 4 входи загального призначення та спеціальний вхід синхронізації (тактовий вхід). КЛБ ІМС цієї серії може генерувати будь-яку логічну функцію чотирьох змінних або дві логічні функції трьох змінних. Змінні для логічних функцій можуть надходити з чотирьох входів та виходу елемента пам’яті.

Область між конфігурованими логічними блоками називається областю програмованих між’єднань і являє собою розвинену ієрархію металічних ліній зв’язку, в місцях перетину яких розміщено спеціальні швидкодіючі транзистори. Функція області між’єднань полягає в забезпеченні зв’язку між будь-якими виводами КЛБ та БВВ.

Необхідний маршрут міжблокових з’єднань у FPGA-ПЛИС реалізується комутацією відповідних ліній за допомогою транзисторів. Нижній шар кристала займає тіньовий запам’ятовуючий пристрій, інформація в елементах якого і визначає логічні функції КЛБ, конфігурацію БВВ та маршрути між’єднань.

У таблиці 1.1 наведено основні характеристики мікросхем FPGA виробництва Xilinx Inc., зокрема, однієї з останніх розробок серії Virtex та масової серії Spartan.

Таблиця 1. – Основні характеристики мікросхем FPGA

Характеристика	Серія Virtex			Серія Spartan	
	XCV50	CV1000	XCS40/XL	XCS20/XL	XCS05/XL
Системна частота (MHz)	200	200	80	80	80
Технологія (мкм)	0.22	0.22	0.35/0.5	0.35/0.5	0.35/0.5
Напруга живлення ядра (В)	2.5	2.5	3.3/5.5	3.3/5.5	3.3/5.5
Кількість системних вентилів (шт.)	57906	112402	13К-40К	7К-20К	2К-5К
Кількість логічних комірок (шт.)	1728	27648	1862	950	238
Максимальне число портів вводу/виводу (шт.)	180	514	205	160	77

Широкий діапазон ІМС FPGA-технології дозволяє проектувати на їх основі широкий спектр електронних пристроїв, серед яких: засоби поєднання різних за живленням інтерфейсів, перетворювачі кодів, периферійні контролери, мікропрограмні пристрої керування, скінченні автомати, універсальні та спеціалізовані процесори, пристрої цифрової обробки сигналів тощо.

Система на кристалі Amber ARM. Ядро процесора Amber з відкритим вихідним кодом є ARM-сумісним 32-бітним RISC-процесором. Саме ядро, вихідний код, знаходиться на сайті [4,5] і є частиною бібліотек з відкритим вихідним кодом. Ядро Amber повністю сумісне з набором команд ARMv2 і підтримується набором інструментів GNU. З даним набором команд сумісне також вільнорозповсюджене ядро STORM Open Soft Core [4]. Ця версія набору інструкцій ARM підтримується вільнорозповсюджуваними компіляторами, оскільки вона не захищена патентами і може бути реалізованою без ліцензії ARM Holdings, на відміну від деяких проектів з відкритим вихідним кодом. Проект Amber є повністю вбудованою в FPGA системою, що включає Amber ядро і деякі периферійні пристрої, наприклад інтерфейси UARTs, таймери, Ethernet MAC.

Є дві версії ядра, передбачені в проекті Amber. Amber 23 має 3-ступінчастий конвеєр, єдиний кеш команд і даних, містить Wishbone інтерфейс і має продуктивність до 0,75 DMIPS на МГц. Версія ядра Amber 25 має 5-ступінчастий конвеєр, окремі кеші для даних та інструкцій, містить Wishbone інтерфейс і має продуктивність 1,0 DMIPS на МГц. Обидва ядра реалізовані з однаковим інтерфейсом і на 100% програмно сумісні між собою.

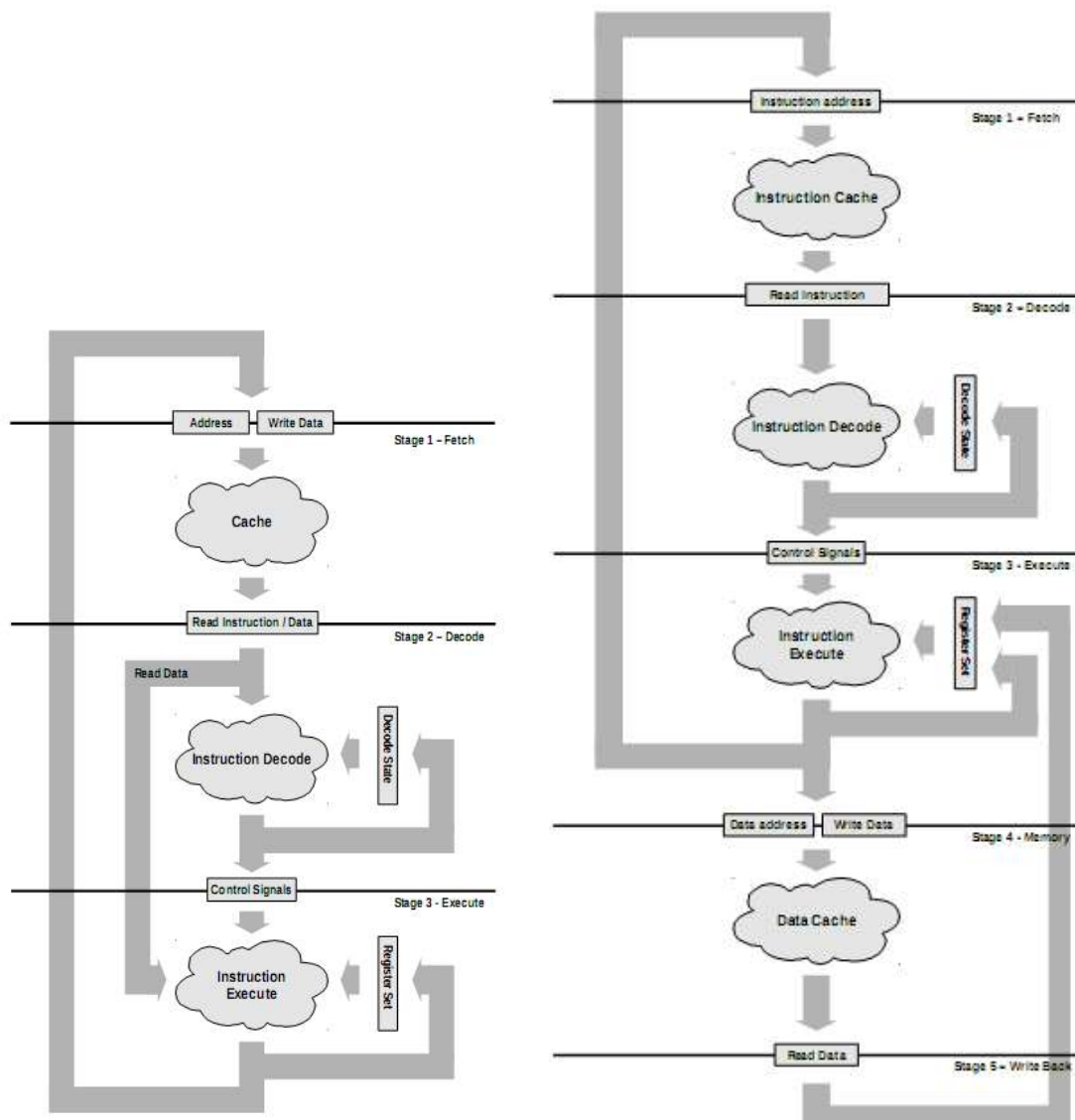


Рис. 2. – Етапи роботи ядер Amber 23 та Amber 25

Amber 23 - дуже маленьке за обсягом 32-бітне ядро, що забезпечує хорошу продуктивність. Інструкції на основі регістрів виконуються за один цикл (такт роботи), винятком є інструкції пов'язані із операціями множення. Команди завантаження і збереження вимагають три цикли. Робота ядра заходить в тупикову ситуація коли відбувається промах кеша, або коли ядро виконує одночасно читання/доступ по шині.

Ядро Amber 25 забезпечує від 30 до 40% кращу продуктивність, ніж ядро Amber 23, але також затрачає на це від 30 до 40% більше енергії і ресурсів мікросхеми. Інструкції на основі регістрів виконуються за один цикл, винятком є інструкції, пов'язані із операціями множення або складними операціями зсуву. Команди завантаження і збереження також виконуються за один цикл, якщо немає конфлікту із вибіркою наступної команди. Робота ядра заходить в тупикову ситуація коли відбувається промах в будь-якому із двох кешів, або коли ядро виконує одночасно читання/доступ по шині.

Портування ядра на кристал Altera Cyclone III. Обидва ядра були перевірені шляхом завантаження Linux 2.4. Версії ядра Linux 2.4 та більш ранні містять конфігурації підтримувані шиною ISA. У версіях ядра Linux 2.6 і пізніших явно не підтримуються ARMv2A ISA і тому потрібно кілька модифікацій вбудованого Amber ядра для запуску. Ядра не містять блок управління пам'яттю (MMU), тому вони можуть працювати тільки в невіртуальному варіанті пам'яті Linux та µClinux. Дані ядра були розроблені у версії Verilog 2001 року і оптимізовані для FPGA синтезу.

Вихідні коди модулів були розроблені для мікросхем Spartan/Virtex фірми Xilinx, що можна бачити на прикладі:

```
// Tag RAMs
`ifdef XILINX_SPARTAN6_FPGA
xs6_sram_256x21_line_en
`endif
`ifdef XILINX_VIRTEX6_FPGA
xv6_sram_256x21_line_en
`endif
`ifndef XILINX_FPGA
generic_sram_line_en
`endif
```

В мікросхемах FPGA є вбудовані блоки пам'яті. Для них існують спеціальні компоненти, в нашому випадку для портування коду на мікросхеми фірми Altera необхідно використати altsyncram. На його основі, задаючи параметри можливо створити модулі пам'яті різної разрядності та об'єму.

Висновки. Здійснено реалізацію системи на кристалі в програмованій логічній інтегральній схемі Altera Cyclone III EP3C10E144C8. Проект побудовано на основі відкритого проекту Amber ARM. В системі реалізовано процесор ARM v2a, послідовний порт, таймер, контролер переривань, контролер пам'яті SDRAM.

ЛІТЕРАТУРА

1. Amber Open Source Project. Amber 2 Core Specification. May 2013.
2. Amber Open Source Project. Amber Project User Guide. May 2013.
3. Donald E.Thomas, Philip R.Moorby. The Verilog Hardware Description Language, 5th edition. - Kluwer Academic Publishers. - 2002. – 404p.
4. <http://opencores.org>.
5. <http://opencores.org/project,amber>.
6. <http://uk.wikipedia.org/wiki/ARM>.
7. <http://www.altera.com>.
8. <http://www.marsohod.org/index.php/prodmarsohod2/amber-arm-soc>.
9. <http://www.xilinx.com>.
10. Verilog. Практика проектування цифрових пристроїв на ПЛІС: Навчальний посібник / Рябенський В.М. Ушкаренко О.О. – Миколаїв: Вид-во "Іліон", 2007. – 324 с.

УДК 004.4'2

А.Ю. Коцюба, В.В. Матвіюк

Луцький національний технічний університет

Імітаційне моделювання симуляції розіграшів національної лотереї "Супер Лото"

Коцюба А.Ю., Матвіюк В.В. Імітаційне моделювання симуляції розіграшів національної лотереї "Супер Лото". Засобами C++ Builder та MS Office розроблено програмний комплекс, за допомогою якого можна будь-яку кількість разів проводити розіграші національної лотереї "Супер Лото". Одержані результати в майбутньому можна досліджувати методами математичної статистики, що дасть можливість відповісти на поставлені в роботі питання.

Ключові слова: імітаційна модель, національна лотерея "Супер Лото", ADO-технологія, OLE-технологія.

Рис. 5, Лит. 8

Коцюба А.Ю., Матвіюк В.В. Имитационное моделирование симуляции розыгрышей национальной лотереи "Супер Лото". Средствами C++ Builder и MS Office разработан программный комплекс, с помощью которого можно любое количество раз проводить розыгрыши национальной лотереи "Супер Лото". Полученные результаты в будущем можно исследовать методами математической статистики, что позволит ответить на поставленные в работе вопросы.

Ключевые слова: имитационная модель, национальная лотерея "Супер Лото", ADO-технология, OLE-технология.

Рис. 5, Лит. 8

Kotsyuba A.Yu, Matviyuk V.V. Imitation modeling of simulation draws National Lottery "Super Lotto". By means of C++ Builder and MS Office developed a software package that allows you to any number of times to conduct the draw of National Lottery "Super Lotto". The results obtained in the future can be studied by methods of mathematical statistics, which give the opportunity to respond to the questions in the question.

Keywords: imitation model, the National Lottery "Super Lotto", ADO-technology, OLE-technology.

Fig. 5, Ref. 8

Постановка наукової проблеми та її значення. Поняття імітаційне моделювання, імітаційна модель це терміни одного з найбільш потужних методів дослідження реально існуючих об'єктів, систем самої різної природи і складності. Суть методу полягає в побудові так званої імітаційної моделі досліджуваного об'єкту і цілеспрямованого експерименту з такою моделлю для отримання відповідей на ті або інші питання поставлені перед дослідником. В даній роботі побудована і реалізована імітаційна модель симуляції розіграшів національної лотереї "Супер Лото". Завдяки цій моделі можна провести дослідження, які б дали змогу відповісти на наступні питання:

- коли і чи можливо виграти у вищевказаній лотереї;
- скільки ставок мінімально потрібно зробити, щоб хоча б повернути витрачені гроші з мінімальною ймовірністю програшу та мінімальними затратами на ставки;
- чи доцільно проводити системні ігри (мається на увазі, що при однакових затратах для випадків системної та звичайної гри, чи буде ймовірність програшу різнитися у вигідну для системної гри сторону);
- якщо доцільно, то яку кількість кульок найкраще вказувати у системній грі.

Звичайно, всі результати ми отримуємо за допомогою так званого генератора псевдовипадкових чисел [-1]. Але експерименти можна робити будь-яку кількість разів та записувати всі результати. При цьому ніяких грошових затрат не буде. В цьому і полягають недолік та перевага симуляції розіграшів побудованої в даній роботі імітаційної моделі.

Крім вищеописаних недоліків та переваг, побудована модель (програмний комплекс) має ще дві досить суттєві переваги:

1. В національній лотереї "Супер Лото" весь час змінюють кількість кульок (номерів), з якої випадковим чином вибирається 6 кульок (номерів). Для нашого програмного комплексу це не проблема. При початковому запуску exe-файла ця кількість вказується. По замовчуванню вона є такою як в сучасних розіграшах.

2. Під час розіграшу 6 різних номерів можна вказувати за допомогою різних алгоритмів. Побудований в даній роботі алгоритм оптимально моделює роботу лототрона (звісно за умови чесного розіграшу): спочатку з масиву всіх впорядкованих необхідних номерів вибирається випадковий, на наступному кроці цей номер вилючається, одержуємо на один номер коротший масив, з яким аналогічно працюємо далі і т.д. Цей алгоритм продовжують до того часу, доки не буде вибрано 6 різних номерів. Подібний процес відбувається і тоді, коли ми хочемо, щоб комп'ютер сам зробив ставки (АВТО) для звичайної або системної гри.

Аналіз останніх досліджень та публікацій. У праці [-1] вивчено питання про генератор псевдовипадкових чисел. Щодо ADO-технології та OLE-технології, то ці теорії досить повно роз-

криті у працях відповідно [-2--5] та [-6, -7]. Основні аспекти мови запитів SQL розглянуто у книзі [-8].

Мега роботи полягає в проектуванні і написанні такого програмного комплексу, який би можна було ефективно використовувати для імітаційного моделювання симуляції розіграшів національної лотереї "Супер Лото".

Методи розробки. Для підключення БД MS Access до додатку була використана ADO-технологія. Для підключення до додатку табличного процесора MS Excel була використана OLE-технологія.

Виклад основного матеріалу дослідження. Опишемо інтерфейс користувача програми, розробленої засобами C++ Builder [-2--7] та мови запитів SQL [-8]. Даний програмний проект складається з трьох форм: стартової, головної та форми-редактора ставок.

Стартова форма (рис 1) призначена лише для встановлення максимальної кількості кульок.



Рис. 1. Стартова форма

В головній формі (рис 2) відображаються всі ставки зроблені гравцями та дані про гравців. В ній проходить процес симуляції розіграшу, виводиться інформація про результат, яку в разі потреби можна занести у звіт (xls-файл). Крім того з цієї форми можна викликати форму-редактор ставок. Передбачено автоматичне сортування наборів номерів: як ставок, так розіграшу; автоматичне виправлення помилок: в разі, якщо користувач неправильно зробив ставки; є можливість перезавантаження, якщо користувач захоче самостійно в базі даних виправити свої помилки.



Рис. 2. Головна форма

У будь-який час користувач може внести зміни або доповнити базу даних, яку опишемо нижче, та перезавантажити головну форму. Але бувають такі випадки, наприклад, системна гра або режим АВТО, коли користувачу для заповнення бази даних необхідна допомога. Саме для цього і створено форму-редактор ставок (рис 3). Тут спочатку необхідно вказати гравця (причому можна додати нового), а далі вручну або за допомогою режиму АВТО створити ставку. Для ручного створення також працює автоматичне виправлення помилок.

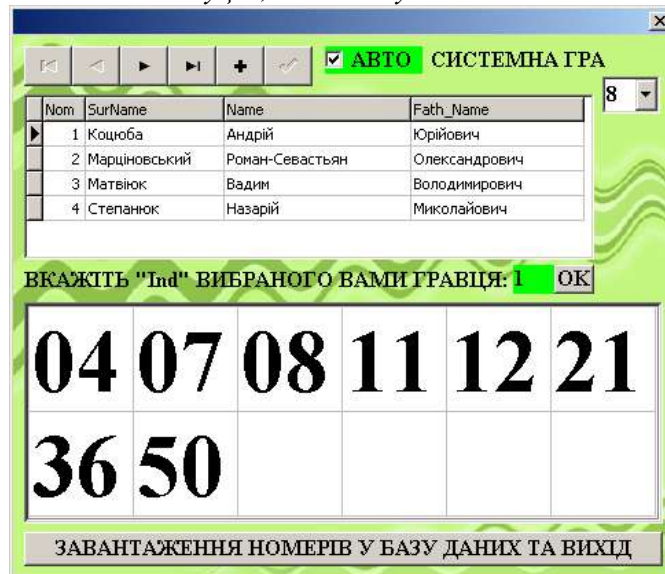


Рис. 3. Форма-редактор ставок

Після натискання на кнопку “Завантаження номерів у базу даних та вихід” буде для першого гравця створено відповідно ще 28 ставок з різних 6 номерів, відсортованих у порядку зростання, закриється ця форма і відкриється перезавантажена головна форма.

Відповідна база даних складається з двох зв’язаних таблиць (рис 4) та запиту, який містить об’єднану з цих таблиць інформацію та відображається у головній формі. Щодо таблиць, то в першу вносяться гравці (вона відображається у формі-редакторі записів), а в другу – ставки і вибираються гравці, які зробили ці ставки.

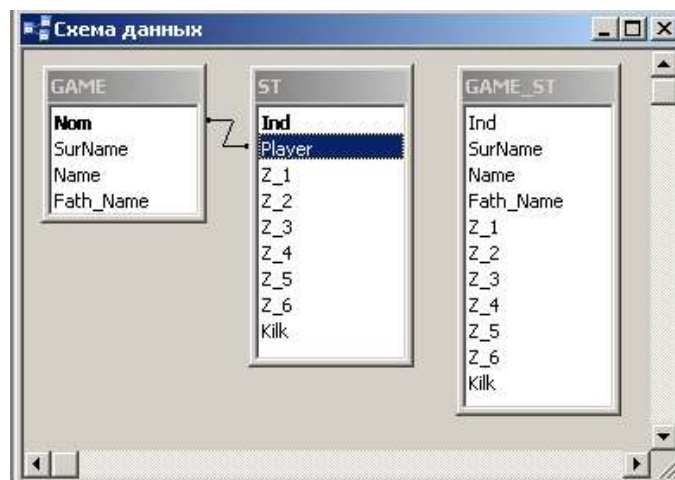


Рис. 5. Фізична модель бази даних

Звернемо увагу на ще один досить цікавий факт ADO-технології. Як правило за допомогою цієї технології загрузка бази даних є стаціонарною, тобто потрібно наперед вказати адресу mdb-файла і якщо на іншому комп’ютері за такою адресою нема бази даних, то exe-файл так як потрібно працювати не буде. Нами ж в цій роботі запропоновано дещо інший підхід:

- по-перше, для ефективної роботи даної технології використані наступні компоненти: TADODConnection, TDataSource, TADODDataSet, TADOQuery, TADODCommand, TDBNavigator, TDBGrid;

- по-друге, перед початковим завантаженням або перезавантаженням форми, на якій буде відображено інформація з бази даних (залежно від SQL-запиту), необхідно встановити у властивості Connected компоненти TADODConnection значення false (це лишнім не буде);

- по-третє, представимо фрагмент коду програми, у якому відловлюється поточна адреса exe-файла і приймається, що вона є адресою mdb-файла:

```
ADODConn->Connected = false;
```

```
String filename = "BAZZA.mdb";
if (!FileExists(ExtractFilePath(Application->ExeName)+filename)) {
    AnsiString ds="Файла бази даних не виявлено \n\n\t" +filename;
    ShowMessage(ds); }
else {
    String WayToBase=ExtractFilePath(Application->ExeName)+filename;
    ADOConn->ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;
User ID=Admin; Data Source="+WayToBase+"; Mode=Share Deny None; Extended
Properties="""; Jet OLEDB:System database="""; Jet OLEDB:Registry
Path="""; Jet OLEDB:Database Password=""";Jet OLEDB:Engine Type=5;
Jet OLEDB:Database Locking Mode=1; Jet OLEDB:Global Partial Bulk Ops=2;
Jet OLEDB:Global Bulk Transactions=1; Jet OLEDB:New Database Password
="""; Jet OLEDB:Create System Database=False; Jet OLEDB:Encrypt Data-
base=False; Jet OLEDB:Don't Copy Locale on Compact=False; Jet OLEDB:
Compact Without Replica Repair=False;Jet OLEDB:SFP=False;";
    ADOConn->Connected = true;
    ADODataSet->Active = true;
    ADODataSet->Open(); }
```

Висновки. Як правило імітаційне моделювання є найбільш ефективним, коли аналітичне дослідження ускладнене або неможливе. За допомогою системи диференціальних рівнянь для ймовірностей станів або рівнянь алгебри для ймовірності станів відповісти на вище поставленні питання досить важко, а деколи і неможливо. Тому доцільно використовувати ідею методу Монте-Карло, яка є надзвичайно простою і полягає у тому, щоб випадковий процес моделювати за допомогою так званих “розіграшів”, тобто за допомогою деякої процедури, яка дає випадковий результат. Особливо ця методика оправдовує себе, коли можна зробити таке програмне за безпечення, яке дасть змогу проробляти велику кількість “розіграшів” і аналізувати їх засобами математичної статистики.

1. Герасимчук О. І. Генератори псевдовипадкових чисел, їх застосування, класифікація, основні методи побудови і оцінка якості / О. І. Герасимчук, В. М. Максимович // Науково-технічний журнал “Захист інформації”, 2003, №3, С. 29-36.

2. Пахомов Б. Самоучитель C/C++ и C++ Builder 2007 / Б. Пахомов. – Санкт-Петербург: “БХВ Петербург”, 2008. – 672 с.

3. Послед Б. С. C++ Builder 6. Разработка приложений баз данных / Б. С. Послед. – Санкт-Петербург: “ДиаСофт”, 2003. – 310 с.

4. Хоменко А. Работа с базами данных в C++ Builder / А. Хоменко, С. Ададунов. – Санкт-Петербург: БХВ-Петербург, 2006. – 475 с.

5. Матеріали сайту www.h-l-l.ru/publ/28-1-0-185, 11.11.2011 р.

6. Бруно Бабэ. Просто и ясно о Borland C++ / Бабэ Бруно. – М., 1996. – 400 с.

7. Матеріали сайту www.realcoding.net, 11.05.2007 р.

8. Кириллов В. В. Структуризованный язык запросов (SQL) / В. В. Кириллов. – Санкт-Петербург: ИТМО, 2005. – 80 с.

УДК 004.891.3: 004.3

А.В.Красій

Хмельницький національний університет

МОДЕЛЮВАННЯ ПРОЦЕСУ ПРОГНОЗУВАННЯ ХАРАКТЕРИСТИК ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ АНАЛІЗУ СПЕЦИФІКАЦІЙ

А.В.Красій. Моделювання процесу прогнозування характеристик програмного забезпечення на основі аналізу специфікацій. Стаття присвячена розробленню математичної моделі специфікації вимог до програмного забезпечення (ПЗ), математичних моделей основних характеристик ПЗ, а також концепції прогнозування характеристик ПЗ на основі аналізу специфікацій та математичної моделі процесу прогнозування характеристик ПЗ, в основі якої лежить штучна нейронна мережа (ШНМ), на входи якої подаються показники специфікації вимог, а вихідними функціоналами є прогнозовані кількісні значення характеристик ПЗ.

Ключові слова: програмне забезпечення (ПЗ), програмний проект, специфікація вимог до ПЗ, кількісні показники специфікації вимог до ПЗ, якісні показники специфікації вимог до ПЗ, експертні кількісні показники специфікації вимог до ПЗ, характеристики ПЗ, штучна нейронна мережа (ШНМ).

Форм. 17. Табл. 0. Рис. 4. Літ. 7.

А.В.Красій. Моделирование процесса прогнозирования характеристик программного обеспечения на основе анализа спецификаций. Статья посвящена разработке математической модели спецификации требований к программному обеспечению (ПО), математических моделей основных характеристик ПО, а также концепции прогнозирования характеристик ПО на основе анализа спецификаций и математической модели процесса прогнозирования характеристик ПО, в основе которой лежит искусственная нейронная сеть (ИНС), на входы которой подаются показатели спецификации требований, а выходными функционалами являются прогнозируемые количественные значения характеристик ПО.

Ключевые слова: программное обеспечение (ПО), программный проект, спецификация требований к ПО, количественные показатели спецификации требований к ПО, качественные показатели спецификации требований к ПО, экспертные количественные показатели спецификации требований к ПО, характеристики ПО, искусственная нейронная сеть (ИНС).

Форм. 17. Табл. 0. Рис. 4. Лит. 7.

A.V.Krasiy. The Modeling of Software Characteristics Prognostication Process on the Basis of Specifications Analysis. The article is devoted to the development of the mathematical model of software requirements specification, of the mathematical models of the main software characteristics, of the concept of prognostication of software characteristics on the basis of specifications analysis, of the mathematical model of software characteristics prognostication process, which is based on artificial neural network (ANN). The inputs of the ANN are indicators of software requirements specification and output functionals are predicted quantitative values of software characteristics.

Key words: software, software project, software requirements specification, quantitative indicators of software requirements specification, qualitative indicators of software requirements specification, expert quantitative indicators of software requirements specification, software characteristics, artificial neural network (ANN).

Form. 17. Tab. 0. Fig. 4. Lit. 7.

Вступ. Процес розроблення програмного забезпечення (ПЗ) тісно пов'язаний з процесом аналізу та оцінювання значущих характеристик ПЗ: тривалості та моделі життєвого циклу, вартості, ефективності, складності, зручності використання, кросплатформеності, надійності та якості ПЗ.

Наразі все помітнішою стає криза у галузі розроблення ПЗ - великі проекти виконуються з відставанням від графіка або з перевищенням кошторису витрат, розроблений продукт не має необхідних функціональних можливостей, продуктивність його низька, якість програмного забезпечення не влаштовує споживачів [2, 3, 7]. При наявності ряду методів та засобів, залученні кращих фахівців для розроблення технологій та стандартів проектування і розроблення ПЗ, процес розроблення ПЗ, як і раніше, залежить від знань та досвіду розробників.

Критичний вплив на програмні проекти та на успішність їх завершення здійснюють питання, пов'язані із аналізом специфікації [3]. Специфікація – це основа для побудови програмного забезпечення. Аналіз специфікації надає кількісну та якісну інформацію для подальшого визначення основних характеристик ПЗ. Крім того, експерти після опрацювання специфікації можуть надати кількісні оцінки якісної інформації для визначення характеристик ПЗ.

Запропонований підхід дасть змогу прийняти мотивоване та обгрунтоване рішення щодо вибору програмного проекту на основі різних важливих характеристик ПЗ. Отже, аналіз специфікації з метою одержання прогнозованих кількісних оцінок є *актуальним* на етапі проектування [3].

Постановка задачі. З результатів попередніх досліджень слідує, що *перспективними напрямками досліджень* наразі є: 1) математичне моделювання специфікації вимог до ПЗ та характеристик ПЗ; 2) побудова концепції та вибір математичного апарату для вирішення задачі

прогнозування характеристик ПЗ; 3) побудова моделі прогнозування характеристик ПЗ на основі аналізу специфікацій.

Специфікація вимог до ПЗ. Існує велика кількість специфікацій та стандартів з їх написання. Основною, найбільш повною з точки зору визначення основних характеристик ПЗ, є *специфікація вимог до ПЗ (SRS)*. Згідно [1], специфікація вимог до ПЗ має наступну структуру (рис.1).

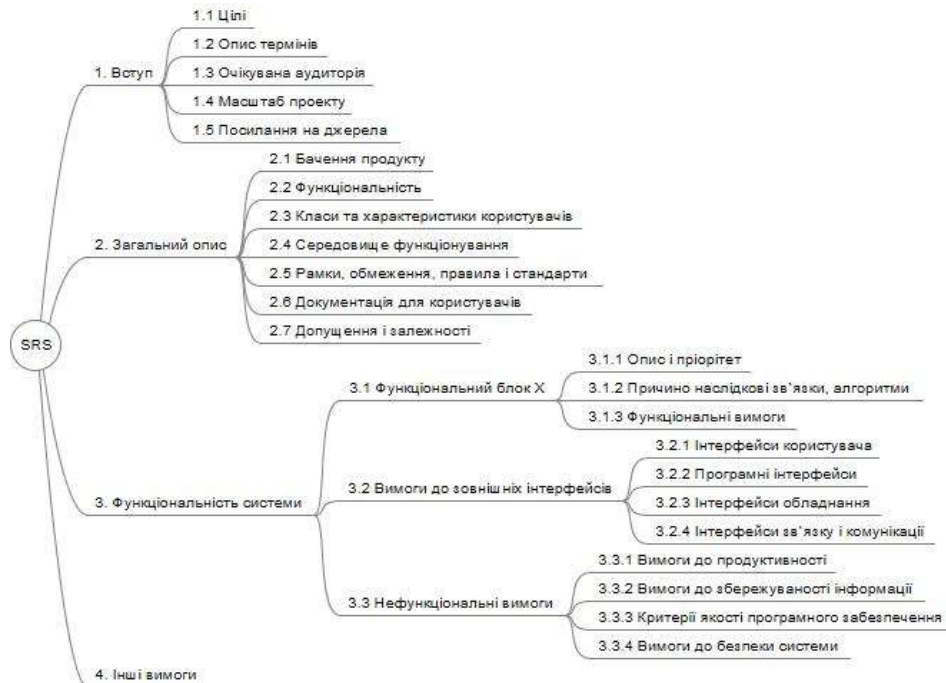


Рис.1. Структура специфікації вимог до ПЗ

Визначимо кількісні показники, які можна одержати із кожного пункту запропонованої стандартної структури SRS [3]: 1.2 – термін виконання, кваліфікація виконавця(ів), кількість виконавців; 1.3 – розмір цільової аудиторії; 1.4 – кількість компонентів системи, розмір кожного компонента, повторюваність; 2.4 – вартість використовуваних операційних систем, баз даних, компіляторів; 2.5 – вартість специфічних засобів розробки; 2.6 – розмір і вартість документації для користувачів; 2.7 – залежність від інших програмних засобів (вартість, складність, надійність); 3.1.1 – перелік всіх функційних вимог, вартість, складність, надійність окремо взятої вимоги; 3.1.2 – складність та надійність окремо взятого алгоритму; 3.1.3 - вартість помилки, реакція на помилку; 3.2.1 – кількість, вартість інтерфейсів користувача; 3.2.2 - кількість, вартість, надійність, складність програмних інтерфейсів; 3.2.3 - кількість, вартість, надійність, складність апаратних інтерфейсів; 3.2.2 - кількість, вартість, надійність, складність інтерфейсів зв'язку і комунікацій; 3.3.1 - перелік всіх нефункційних вимог, вартість, складність, надійність окремо взятої вимоги.

Якісні показники специфікації вимог до ПЗ [3]: 1.3 – профілювання (цільова аудиторія); 2.3 – вміння користувачів працювати із аналогічними продуктами; 2.4 – складність операційних систем, баз даних, компіляторів; 3.1.1 - залежність між вимогами, суперечливість вимог; 3.3.3 – якість програмного забезпечення; 3.3.4 – рівень безпеки програмного забезпечення.

Математична модель специфікації вимог до ПЗ. Перш ніж формалізувати специфікацію вимог до ПЗ, введемо декілька визначень.

Визначення 1. Кількісними показниками (КП) специфікації вимог до ПЗ або її розділів називатимемо корисні для визначення характеристик ПЗ показники специфікації, які мають точне кількісне значення (наприклад, "кількість виконавців - 7", "середня вартість помилки - 89\$" і т.і.).

Визначення 2. Якісними показниками специфікації вимог до ПЗ або її розділів називатимемо корисні для визначення характеристик ПЗ показники специфікації, які виражені лінгвістично (наприклад, "користувачі мають навички роботи із аналогічними продуктами" і т.і.).

Визначення 3. Експертними кількісними показниками (ЕКП) специфікації вимог до ПЗ або її розділів називатимемо корисні для визначення характеристик ПЗ показники, які у специфікації виражені лінгвістично, але після опрацювання групою експертів набули кількісного виразу (наприклад, вищенаведений якісний показник перетворений після опрацювання групою експертів у експертний кількісний показник "середня оцінка навичок роботи користувачів із аналогічними продуктами складає 3,8 (за 5-бальною шкалою)" і т.і.).

Представимо тепер специфікацію вимог до ПЗ (SRS) у наступному формалізованому вигляді:

$$SRS = \langle R1, R2, R3, R4 \rangle,$$

(1)

де $R1$ - множина кількісних та експертних кількісних показників розділу 1 специфікації вимог до ПЗ, $R2$ - множина кількісних та експертних кількісних показників розділу 2, $R3$ - множина кількісних та експертних кількісних показників розділу 3, $R4$ - множина кількісних та експертних кількісних показників розділу 4.

Тоді, враховуючи вищевикладену інформацію, представимо розділи специфікації вимог у вигляді наступних множин:

$$R1 = \{Tv, Qv, Sa, Qcs, SC, Pr of\},$$

(2)

де Tv - термін виконання, Qv - кількість виконавців, Sa - розмір цільової аудиторії, Qcs - кількість компонентів програмної системи, $SC = \{sc_1, \dots, sc_{Qcs}\}$ - множина розмірів усіх компонентів ПЗ (sc_1 - розмір компоненту 1, sc_{Qcs} - розмір компоненту Qcs), $Pr of$ - профілювання, цільова аудиторія (експертний кількісний показник);

$$R2 = \{Cos, Cdb, Cc, Cdt, Sud, Cud, Us, Cxos, Cxdb, Cxc\},$$

(3)

де Cos - вартість використовуваних операційних систем, Cdb - вартість використовуваних баз даних, Cc - вартість використовуваних компіляторів, Cdt - вартість засобів розроблення, Sud - вартість документації користувача, Cud - розмір документації користувача, Us - вміння користувачів працювати із аналогічними продуктами (експертний кількісний показник), $Cxos$ - складність використовуваних операційних систем (експертний кількісний показник), $Cxdb$ - складність використовуваних баз даних (експертний кількісний показник), Cxc - складність використовуваних компіляторів (експертний кількісний показник);

$$R3 = \left\{ Qfr, CFR, CXFR, RFR, Qa, CXA, RA, Cb, Cui, Qmi, CMI, RMI, CXMI, Qai, \right. \\ \left. \{ Cai, Rai, Cxai, Qci, Cci, Rci, Cxci, Qnfr, CNFR, CXNFR, RNFR, Rr, Ric, Sq, Ssq \} \right\},$$

(4)

де Qfr - кількість функційних вимог специфікації вимог до ПЗ, $CFR = \{cfr_1, \dots, cfr_{Qfr}\}$ - множина значень вартості функційних вимог (cfr_1 - вартість функційної вимоги 1, cfr_{Qfr} - вартість функційної вимоги Qfr), $CXFR = \{cxfr_1, \dots, cxfr_{Qfr}\}$ - множина значень складності функційних вимог ($cxfr_1$ - складність функційної вимоги 1, $cxfr_{Qfr}$ - складність функційної вимоги Qfr), $RFR = \{rfr_1, \dots, rfr_{Qfr}\}$ - множина значень надійності функційних вимог (rfr_1 - надійність функційної вимоги 1, rfr_{Qfr} - надійність функційної вимоги Qfr), Qa - кількість використовуваних алгоритмів, $CXA = \{cxa_1, \dots, cxa_{Qa}\}$ - множина значень складності використовуваних алгоритмів (cxa_1 - складність алгоритму 1, cxa_{Qa} - складність алгоритму Qa), $RA = \{ra_1, \dots, ra_{Qa}\}$ - множина значень

надійності використовуваних алгоритмів ($га_1$ - надійність алгоритму 1, $га_{Qa}$ - надійність алгоритму Qa), $Сb$ - середня прогнозована вартість помилки, $Сui$ - вартість інтерфейсів користувача, Qmi - кількість програмних (міжмодульних) інтерфейсів, $СMI = \{сmi_1, \dots, сmi_{Qmi}\}$ - множина значень вартості програмних (міжмодульних) інтерфейсів ($сmi_1$ - вартість програмного інтерфейсу 1, $сmi_{Qmi}$ - вартість програмного інтерфейсу Qmi), $RMi = \{rmi_1, \dots, rmi_{Qmi}\}$ - множина значень надійності програмних (міжмодульних) інтерфейсів (rmi_1 - надійність програмного інтерфейсу 1, rmi_{Qmi} - надійність програмного інтерфейсу Qmi), $СXMI = \{сxmi_1, \dots, сxmi_{Qmi}\}$ - множина значень складності програмних (міжмодульних) інтерфейсів ($сxmi_1$ - складність програмного інтерфейсу 1, $сxmi_{Qmi}$ - складність програмного інтерфейсу Qmi), Qai - кількість апаратних інтерфейсів, $Сai$ - вартість апаратних інтерфейсів, Rai - надійність апаратних інтерфейсів, $Сxai$ - складність апаратних інтерфейсів, Qci - кількість інтерфейсів зв'язку та комунікацій, $Сci$ - вартість інтерфейсів зв'язку та комунікацій, Rci - надійність інтерфейсів зв'язку та комунікацій, $Сxci$ - складність інтерфейсів зв'язку та комунікацій, $Qnfr$ - кількість нефункційних вимог специфікації вимог до ПЗ, $СNFR = \{сnfr_1, \dots, сnfr_{Qnfr}\}$ - множина значень вартості нефункційних вимог ($сnfr_1$ - вартість нефункційної вимоги 1, $сnfr_{Qnfr}$ - вартість нефункційної вимоги $Qnfr$), $СXNFR = \{сxnfr_1, \dots, сxnfr_{Qnfr}\}$ - множина значень складності нефункційних вимог ($сxnfr_1$ - складність нефункційної вимоги 1, $сxnfr_{Qnfr}$ - складність нефункційної вимоги $Qnfr$), $RNFR = \{rnfr_1, \dots, rnfr_{Qnfr}\}$ - множина значень надійності нефункційних вимог ($rnfr_1$ - надійність нефункційної вимоги 1, $rnfr_{Qnfr}$ - надійність нефункційної вимоги $Qnfr$), Rr - залежність між вимогами специфікації (експертний кількісний показник), Ric - суперечливість вимог специфікації (експертний кількісний показник), Sq - очікувана якість програмного забезпечення (експертний кількісний показник), Ssq - очікуваний рівень безпеки ПЗ (експертний кількісний показник);

$$R4 = \{r_1, \dots, r_{nr}\} \quad (5)$$

- множина додаткових характеристик, які можна одержати із розділу 4 специфікації вимог до ПЗ (nr - кількість можливих додаткових характеристик, змінна величина для різних специфікацій). Враховуючи той факт, що розділ 4 є необов'язковим, а за наявності має змінну структуру та величину, надалі не враховуватимемо його показники при визначенні характеристик ПЗ.

В такому випадку математичну модель специфікації вимог до ПЗ з точки зору наявності у ній кількісних та експертних кількісних показників, корисних для визначення характеристик ПЗ, представимо наступним чином:

$$SRS = \left\langle \begin{array}{l} \{Tv, Qv, Sa, Qcs, SC, Pr of\}, \\ \{Cos, Cdb, Cc, Cdt, Sud, Cud, Us, Cxos, Cxdb, Cxc\}, \\ \{Qfr, CFR, CXFR, RFR, Qa, CXA, RA, Cb, Cui, Qmi, CMI, RMI, CXMI, \\ Qai, Cai, Rai, Cxai, Qci, Cci, Rci, Cxci, Qnfr, CNFR, CXNFR, RNFR, Rr, Ric, Sq, Ssq\} \end{array} \right\rangle \quad (6)$$

Математичні моделі характеристик програмного забезпечення. Визначимо залежність кожної основної характеристики ПЗ від кількісних та експертних кількісних показників специфікації вимог до ПЗ (рис. 1) на основі аналізу характеристик ПЗ [3, 6] із залученням експертів. Зрозуміло, що певні показники специфікації будуть впливати на різні характеристики ПЗ.

Враховуючи одержані залежності між характеристиками ПЗ та показниками специфікації вимог до ПЗ, побудуємо математичні моделі характеристик програмного забезпечення.

Математична модель вартості ПЗ має вигляд:

$$C = f \left(\begin{matrix} Tv, Qv, Qcs, SC, Cos, Cdb, Cc, Cdt, Sud, Qfr, CFR, Cb, Cui, Qmi, CMI, Cai, Cci, \\ Qnfr, CNFR, Pr of, Us, Sq, Ssq \end{matrix} \right).$$

(7)

Математичну модель тривалості життєвого циклу ПЗ представимо наступним чином:

$$LcD = \varphi(Tv, Qv, Qcs, SC, Cud, Qfr, Qa, Qmi, Qai, Qci, Qnfr, Ric, Sq, Ssq).$$

(8)

Математичне рівняння для вибору типу моделі життєвого циклу ПЗ має вигляд:

$$SLcM = \phi \left(\begin{matrix} Qv, Sa, Qcs, Qfr, CFR, CXFR, RFR, CXA, RA, Qnfr, CNFR, CXNFR, \\ RNFR, Pr of, Us, Cxos, Cxdb, Cxc, Rr, Ric \end{matrix} \right).$$

(9)

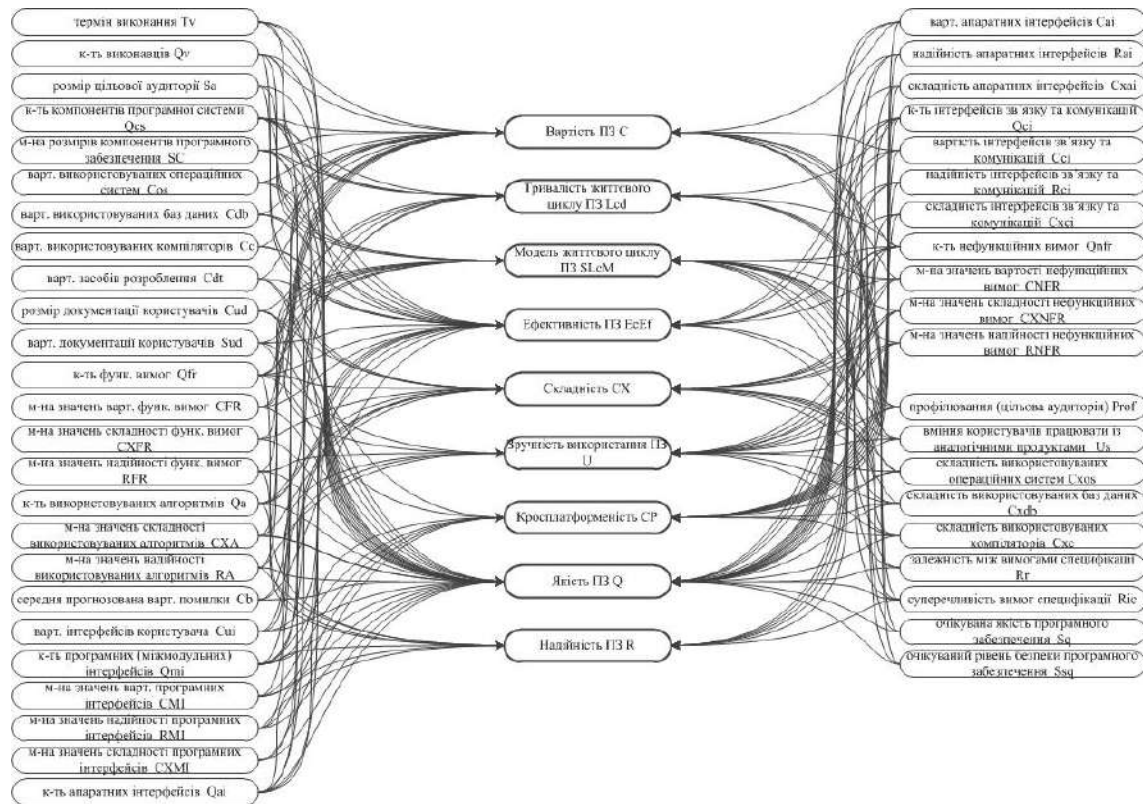


Рис. 2. Вплив КП та ЕКП специфікації на характеристики ПЗ

Математична модель ефективності ПЗ має вигляд:

$$EcEf = \gamma \left(\begin{matrix} Tv, Qv, Sa, Qcs, SC, Cos, Cdb, Cc, Cdt, Sud, Qfr, CFR, Qa, Cb, Cui, CMI, Qmi, \\ Qai, Cai, Qci, Cci, Qnfr, CNFR, Pr of, Us \end{matrix} \right).$$

(10)

Математичну модель складності ПЗ представимо наступним чином:

$$CX = \sigma \left(\begin{matrix} Qcs, SC, Cud, Qfr, CXFR, Qa, CXA, Qmi, CXMI, Qai, Cxai, Qci, Cxci, Qnfr, \\ CXNFR, Cxos, Cxdb, Cxc, Rr, Ric \end{matrix} \right).$$

(11)

Математична модель зручності використання має вигляд:

$$U = \zeta \left(\begin{array}{l} \text{Cud, Sud, RFR, RA, Cui, RMI, Rai, Rci, Qnfr, CNFR, CXNFR, RNFR, Pr of, Us,} \\ \text{CXos, Cxdb, Cxc} \end{array} \right). \quad (12)$$

Математичну модель кросплатформеності ПЗ представимо наступним чином:

$$(13) \quad CP = \xi \left(\begin{array}{l} \text{Qcs, Cdt, Qmi, CMI, RMI, CXMI, Qai, Cai, Rai, Cxai, Qci, Cci, Rci,} \\ \text{Cxc, Cxci, Cxos, Cxdb, Cxc} \end{array} \right).$$

Математична модель якості ПЗ має вигляд:

$$(14) \quad Q = \psi \left(\begin{array}{l} \text{Tv, Qv, Sa, Qcs, SC, Cos, Cdb, Cc, Cdt, Cud, Sud, Qfr, CFR, CXFR, RFR, Qa,} \\ \text{CXA, RA, Cb, Cui, Qmi, CMI, RMI, CXMI, Qai, Cai, Rai, Cxai, Qci, Cci, Rci,} \\ \text{Cxc, Cxci, Qnfr, CNFR, CXNFR, RNFR, Us, Pr of, Cxos, Cxdb, Cxc, Rr, Ric, Sq, Ssq} \end{array} \right).$$

Математичну модель надійності ПЗ представимо наступним чином:

$$R = \omega(Qfr, RFR, Qa, RA, Qmi, RMI, Qai, Rai, Qci, Rci, Qnfr, RNFR, Ric).$$

(15)

В такому разі, маючи специфікацію вимог до ПЗ, можна визначити всі основні характеристики ПЗ за запропонованими формулами. Але проблемою при цьому є визначення, власне, функцій для кожної характеристики (функцій $f, \varphi, \phi, \gamma, \sigma, \zeta, \xi, \psi, \omega$), які враховуватимуть взаємовпливи та різні ступені впливу КП та ЕКП специфікації в межах однієї характеристики.

Концепція прогнозування характеристик ПЗ на основі аналізу специфікацій. Отже, для розв'язку задачі прогнозування характеристик ПЗ на основі кількісних та експертних кількісних показників специфікації вимог до ПЗ потрібно розв'язати наступну систему рівнянь:

$$\left\{ \begin{array}{l} C = f(X_c) \\ LcD = \varphi(X_{LcD}) \\ SLcM = \phi(X_{SLcM}) \\ EcEf = \gamma(X_{EcEf}) \\ CX = \sigma(X_{CD}) \\ U = \zeta(X_U) \\ CP = \xi(X_{CP}) \\ Q = \psi(X_Q) \\ R = \omega(X_R) \end{array} \right. ,$$

(16)

де $X_c = \{x_{c_1}, \dots, x_{c_n}\}$ - множина кількісних та експертних кількісних показників специфікації, які впливають на оцінку вартості ПЗ; n - кількість КП та ЕКП, які впливають на оцінку вартості ПЗ (згідно вищенаведених моделей характеристик ПЗ $n = 23$); $X_{LcD} = \{x_{LcD_1}, \dots, x_{LcD_m}\}$ - множина КП та ЕКП специфікації, які впливають на оцінку тривалості життєвого циклу ПЗ; m - кількість КП та ЕКП, які впливають на оцінку тривалості життєвого циклу ПЗ ($m = 14$); $X_{SLcM} = \{x_{SLcM_1}, \dots, x_{SLcM_k}\}$ - множина КП та ЕКП специфікації, які впливають на вибір моделі життєвого циклу ПЗ; k - кількість КП та ЕКП, які впливають на вибір моделі життєвого циклу ПЗ ($k = 20$); $X_{EcEf} = \{x_{EcEf_1}, \dots, x_{EcEf_l}\}$ - множина КП та ЕКП специфікації, які впливають на оцінку ефективності ПЗ; l - кількість КП та ЕКП, які впливають на оцінку ефективності ПЗ ($l = 25$); $X_{CX} = \{x_{CX_1}, \dots, x_{CX_s}\}$ - множина КП та ЕКП специфікації, які впливають на оцінку складності ПЗ; s - кількість КП та ЕКП, які впливають на

оцінку складності ПЗ ($s = 20$); $X_U = \{x_{U_1}, \dots, x_{U_s}\}$ - множина КП та ЕКП специфікації, які впливають на оцінку зручності використання ПЗ; t - кількість КП та ЕКП, які впливають на оцінку зручності використання ПЗ ($t = 17$); $X_{CP} = \{x_{CP_1}, \dots, x_{CP_p}\}$ - множина КП та ЕКП специфікації, які впливають на оцінку кросплатформеності ПЗ; p - кількість КП та ЕКП, які впливають на оцінку кросплатформеності ПЗ ($p = 17$); $X_Q = \{x_{Q_1}, \dots, x_{Q_a}\}$ - множина КП та ЕКП специфікації, які впливають на оцінку якості ПЗ; a - кількість КП та ЕКП, які впливають на оцінку якості ПЗ ($a = 45$); $X_R = \{x_{R_1}, \dots, x_{R_b}\}$ - множина КП та ЕКП специфікації, які впливають на оцінку надійності ПЗ; b - кількість КП та ЕКП, які впливають на оцінку надійності ПЗ ($b = 13$).

Побудувати багатовимірне відображення $X \rightarrow Y$ означає представити його за допомогою математичних операцій над не більш, ніж двома змінними [4]. Проблема представлення функцій багатьох змінних у вигляді суперпозиції функцій меншої кількості змінних протягом багатьох років розв'язувалась Колмогоровим та Арнольдом, в результаті чого одержано ряд важливих теоретичних результатів [4]: 1) теорема про можливість представлення неперервних функцій декількох змінних суперпозиціями неперервних функцій меншої кількості змінних; 2) теорема про представлення будь-якої неперервної функції трьох змінних у вигляді суми функцій не більше двох змінних; 3) теорема про представлення неперервних функцій декількох змінних у вигляді суперпозицій неперервних функцій однієї змінної і додавання, яка була перероблена Хехт-Нільсеном для нейронних мереж.

Теорема Хехт-Нільсена [4] в неконструктивній формі доводить можливість розв'язку задачі представлення функції довільного вигляду на нейронній мережі і вказує для кожної задачі мінімальні кількості нейронів мережі, необхідних для її розв'язку. З теореми Хехт-Нільсена слідує можливість представлення будь-якої багатовимірної функції декількох змінних за допомогою нейронної мережі фіксованого розміру [4]. Згідно теореми про повноту, будь-яка неперервна функція на замкненому обмеженому просторі може бути рівномірно наближена функціями, які обчислюються нейронними мережами, якщо функція активації нейрона двічі неперервно диференційована і неперервна [4].

Таким чином, штучні нейронні мережі (ШНМ) є універсальними структурами, що дозволяють узагальнити інформацію та виявити залежності між вхідними і результуючими даними, отже, для реалізації вищезазначеної системи рівнянь (16) буде використовуватись саме ШНМ.

Тоді слід розробити ШНМ, яка опрацюватиме множини КП та ЕКП специфікації, здійснюватиме апроксимацію показників та надаватиме прогнозовані оцінки характеристик ПЗ.

Концепцію прогнозування основних характеристик ПЗ на основі аналізу специфікації вимог до ПЗ із використанням штучної нейронної мережі представлено на рис.3.

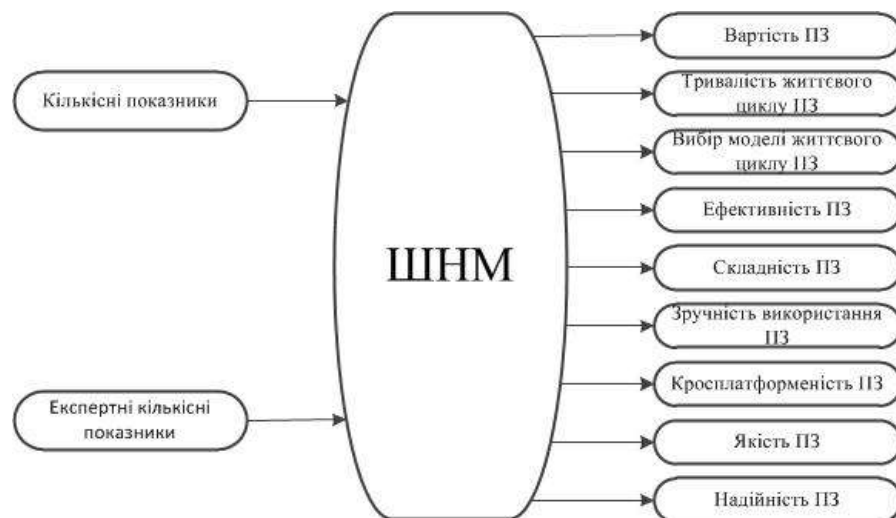


Рис.3. Концепція прогнозування характеристик ПЗ на основі аналізу специфікацій із використанням штучної нейронної мережі

Вибір структури та архітектури нейронної мережі здійснюється відповідно до особливостей і складності задачі. Для розв'язку окремих типів задач вже існують оптимальні конфігурації. Оскільки поставлена задача не може бути зведена до жодного з відомих типів, доводиться розв'язувати проблему синтезу нової конфігурації. При цьому керуватимемось наступними правилами [4]: 1) можливості мережі зростають із збільшенням кількості нейронів мережі, щільності зв'язків між ними та кількості шарів; 2) введення обернених зв'язків поряд із збільшенням можливостей мережі підвищує динамічну стійкість мережі; 3) складність алгоритмів функціонування мережі, введення декількох типів синапсів сприяє посиленню потужності нейронної мережі.

Нейромережна модель прогнозування характеристик програмного забезпечення на основі аналізу специфікацій (НМПХ). Використовуючи запропоновану концепцію прогнозування характеристик ПЗ на основі аналізу специфікацій, розробимо математичну модель процесу прогнозування характеристик ПЗ, поклавши в її основу штучну нейронну мережу (ШНМ). Вибір апарату ШНМ мотивований тим, що штучні нейронні мережі за рахунок можливості апроксимацій нелінійних функцій дають можливість враховувати важливість (ваги) кожного показника специфікації, а також взаємний вплив показників специфікації в межах кожної характеристики ПЗ.

Важкоформалізованою задачею прогнозування характеристик ПЗ є визначення ваг та взаємовпливу кількісних та експертних кількісних показників специфікації в межах кожної характеристики ПЗ. Ця задача може бути вирішена за допомогою використання навченої ШНМ.

Перед розробленням нейромережної моделі прогнозування характеристик ПЗ на основі аналізу специфікацій слід виконати вибір архітектури ШНМ, враховуючи запропоновану концепцію прогнозування характеристик ПЗ. Для цього дослідимо найбільш поширені архітектури ШНМ та їх особливості [5]:

1) багат шаровий перцептрон – найпростіша і найбільш досліджена структура, якої достатньо для задачі прогнозування характеристик ПЗ на основі аналізу показників специфікацій;

2) радіальні базисні мережі (РБМ) вимагають великої кількості навчальних векторів та зазнають значних труднощів, коли є велика кількість входів. Для задачі аналізу показників специфікацій та прогнозування характеристик ПЗ важко зібрати велику кількість навчальної інформації через відсутність великої кількості доступних специфікацій; крім того, має місце велика кількість входів;

3) мережі GRNN (регресійні) – один з видів РБМ, використовуються для аналізу числових рядів. В задачі аналізу показників специфікації та прогнозування характеристик ПЗ зв'язків, аналогічних числовому ряду, немає;

4) мережі PNN (імовірнісні) – один з видів РБМ, призначені для розв'язання імовірнісних задач, зокрема, задач класифікації – встановлення приналежності об'єкту до певного класу. В задачі прогнозування характеристик ПЗ немає потреби в класифікації;

5) карта Кохонена – призначена для кластеризації даних. Ця задача під час аналізу показників специфікації та прогнозування характеристик ПЗ не вирішується;

6) мережі для класифікації входних векторів (LVQ) – використовуються для кластеризації і класифікації. Ці задачі під час прогнозування характеристик ПЗ не розглядаються;

7) мережі Елмана та Хопфілда – це мережі з динамічними оберненими зв'язками, орієнтовані на опрацювання динамічних моделей, що враховують передісторію процесів, які спостерігаються. Задача прогнозування характеристик ПЗ не містить обернених зв'язків і передісторії.

Аналіз відомих архітектур ШНМ показав, що для вирішення задачі аналізу показників специфікацій та прогнозування характеристик ПЗ найбільше підходить багат шаровий перцептрон. При використанні нейромережі іншого типу для розв'язання цієї задачі її природа буде штучно спотворюватись, в результаті чого результати роботи ШНМ будуть невідповідними.

У багат шаровому перцептроні нейрони містяться у шарах, що розташовані один за одним, і взаємозв'язки між нейронами шарів передбачені тільки у напрямку від входу до виходу ШНМ [4].

Відобразимо зазначений підхід узагальненою складною ШНМ архітектури багат шарового перцептрона типу MLP (multi-layer-perceptron). Структура цієї ШНМ представлена на рис.4.

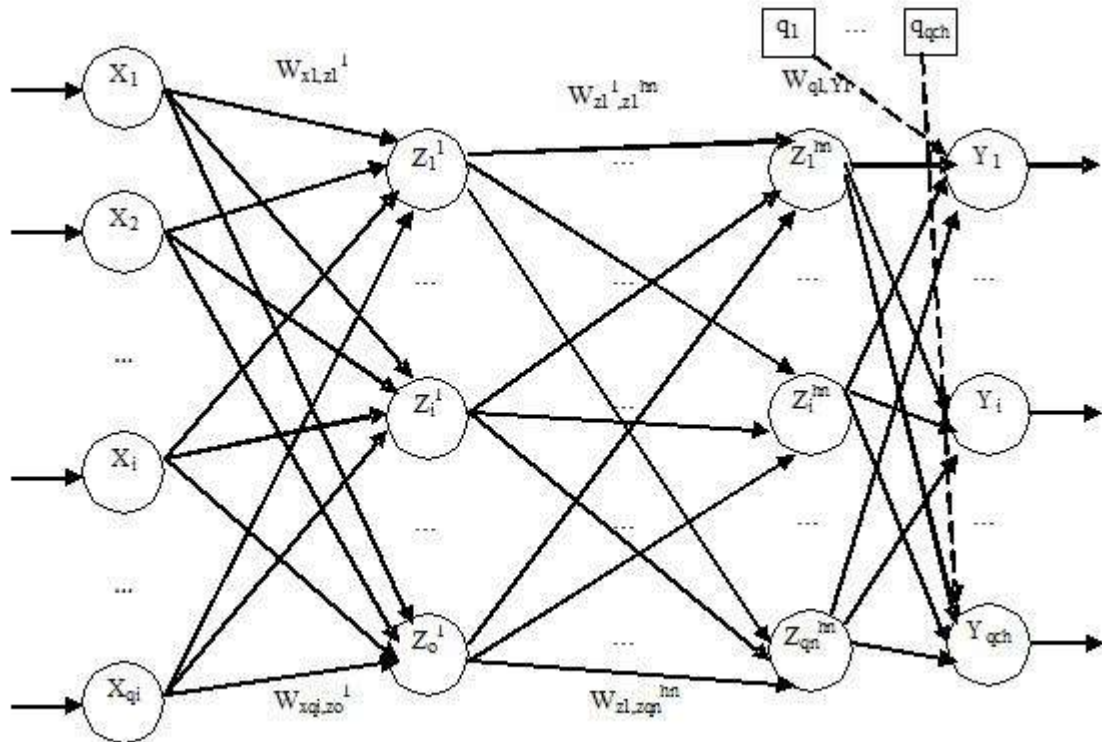


Рис.4. Нейромережна модель прогнозування характеристик ПЗ на основі аналізу специфікацій

Нейрони (і відповідно входи) рецепторного шару визначено множиною $\overline{X} = \{x_i\}$, $i = (\overline{1, qi})$, де x_i - i -й нейрон (вхід) рецепторного шару, qi - кількість входів (кількість КП та ЕКП).

Нейрони першого прихованого шару прямонапрявленого перцептрона визначено множиною $\overline{Z}^1 = \{z_i^1\}$, $i = (\overline{1, o})$, де z_i^1 - i -й нейрон шару z^1 , o - кількість нейронів першого прихованого шару. Рецепторні входи цього шару визначено множиною: $\overline{X}^{z^1} = \{x^{z_i^1}\}$, $i = (\overline{1, o})$, де $x^{z_i^1}$ - i -й вхід, пов'язаний з z_i^1 -м нейроном. Нейрони другого прихованого шару прямонапрявленого перцептрона визначено множиною $\overline{Z}^2 = \{z_i^2\}$, $i = (\overline{1, qn2})$, де z_i^2 - i -й нейрон шару z^2 , $qn2$ - кількість нейронів другого прихованого шару. Рецепторні входи цього шару визначено множиною: $\overline{X}^{z^2} = \{x^{z_i^2}\}$, $i = (\overline{1, qn2})$, де $x^{z_i^2}$ - i -й вхід, пов'язаний з z_i^2 -м нейроном. Нейрони hn -го прихованого шару прямонапрявленого перцептрона визначено множиною $\overline{Z}^h = \{z_i^{hn}\}$, $i = (\overline{1, qn})$, де z_i^{hn} - i -й нейрон шару z^{hn} , qn - кількість нейронів hn -го прихованого шару. Рецепторні входи цього шару визначено множиною $\overline{Z}^{z^h} = \{z^{z_i^{hn-1}}\}$, $i = (\overline{1, qnm})$, де $z^{z_i^{hn-1}}$ - i -й нейрон попереднього ($hn-1$)-го шару, що пов'язаний з z_i^{hn-1} -м нейроном, qnm - кількість нейронів ($hn-1$)-го прихованого шару.

Вектор порогових величин зміщень множини нейронних елементів визначено як: $\overline{q} = \{q_i\}$, $i = (\overline{1, qch})$, де q_i - зміщення i -го нейронного елемента.

Ваги зв'язків відображено ваговими матрицями: $\overline{W}_{x,z^1} = \langle w_{x_i,z_j^1} \rangle$, $i = (\overline{1, qi})$, $j = (\overline{1, o})$, де w_{x_i,z_j^1} - ваговий коефіцієнт зв'язку між x_i -м входом та z_j^1 -м нейроном першого прихованого шару;

$\overline{W_{z^1, z^2}} = \langle w_{z^1, z^2} \rangle$, $i = (\overline{1, o}), j = (\overline{1, qn2})$, де w_{z^1, z^2} - ваговий коефіцієнт зв'язку між z^1 -м нейроном 1-го прихованого шару та z^2 -м нейроном 2-го прихованого шару;

$\overline{W_{z^1, z^{hn}}} = \langle w_{z^1, z^{hn}} \rangle$, $i = (\overline{1, o}), j = (\overline{1, qn})$, де $w_{z^1, z^{hn}}$ - ваговий коефіцієнт зв'язку між z^1 -м нейроном 1-го прихованого шару та z^{hn} -м нейроном hn -го прихованого шару;

$\overline{W_{z^{hn-1}, z^{hn}}} = \langle w_{z^{hn-1}, z^{hn}} \rangle$, $i = (\overline{1, qnm}), j = (\overline{1, qn})$, де $w_{z^{hn-1}, z^{hn}}$ - ваговий коефіцієнт зв'язку між z^{hn-1} -м нейроном $(hn - 1)$ -го прихованого шару та z^{hn} -м нейроном hn -го прихованого шару;

$\overline{W_{q, Y}} = \langle w_{q, Y} \rangle$, $i = (\overline{1, qch})$, де $w_{q, Y}$ - ваговий коефіцієнт зв'язку між q -м зміщенням та Y -м нейроном вихідного шару.

Множина вихідних функціоналів ефекторного шару ШНМ становить $\overline{Y} = \{Y_{ki}\}$, $ki = (\overline{1, qch})$, де Y_{ki} - ki -й функціонал ефекторного шару багатошарового перцептронну, qch - кількість виходів ШНМ (кількість прогнозованих характеристик ПЗ).

Формула для визначення ki -го функціоналу ефекторного шару ШНМ Y_{ki} має вигляд:

$$Y_{ki} = fs \left(fa(z^1) \cdot \left(\sum_{i=1}^o \sum_{j=1}^{qn2} (z^1 \cdot w_{z^1, z^2_j}) + \dots + \sum_{i=1}^o \sum_{j=1}^{qn} (z^1 \cdot w_{z^1, z^{hn}_j}) \right) + \dots + fa(z^{hn-1}) \cdot \left(\sum_{i=1}^{qnm} \sum_{j=1}^{qn} (z^{hn-1} \cdot w_{z^{hn-1}, z^{hn}_j}) \right) + \sum_{i=1}^{qki} (x_i \cdot w_{x_i, z^1}) \right) - q_{ki} \cdot w_{q_{ki}, Y_{ki}}$$

(17)

де $fa(z^1)$ - активаційна функція нейронів 1-го прихованого шару, $fa(z^{hn-1})$ - активаційна функція нейронів $(hn - 1)$ -го прихованого шару, fs - активаційна функція нейронів ефекторного (вихідного) шару ШНМ, qki - кількість КП та ЕКП специфікації, які впливають на ki -й вихід ШНМ (на ki -у прогнозовану характеристику ПЗ), x_i - i -й елемент підмножини КП та ЕКП, які впливають на ki -й вихід ШНМ.

Активаційною функцією нейронів прихованих (асоціативних) шарів є функція гіперболічного тангенсу. Активаційною функцією нейронів ефекторного шару є лінійна функція. Результати лінійної активаційної функції нейронів ефекторних шарів лежать в інтервалі $[-1; 1]$.

Висновки. У даній роботі запропоновано математичну модель специфікації вимог до ПЗ, яка відрізняється від існуючих тим, що представляє специфікацію вимог до ПЗ з точки зору наявності у ній КП та ЕКП, корисних для визначення основних характеристик ПЗ.

Також розроблено математичні моделі основних характеристик ПЗ, згідно яких можна визначити характеристики ПЗ на основі аналізу специфікації вимог до ПЗ вже на початку етапу проектування, що є необхідним для прийняття ґрунтовних рішень щодо вибору програмного проекту.

Запропонована автором концепція прогнозування характеристик ПЗ на основі аналізу специфікацій дала можливість автору розробити математичну модель процесу прогнозування характеристик ПЗ, в основі якої лежить штучна нейронна мережа (ШНМ).

Розроблена нейромережна модель прогнозування характеристик ПЗ на основі аналізу специфікацій (НМПХ) відрізняється від існуючих тим, що дає можливість враховувати ваги та взаємовпливи показників специфікації в межах кожної характеристики ПЗ. Вихідні функціонали ШНМ, що відповідають прогнозованим значенням характеристик ПЗ, дають можливість оцінити

сумарний вплив КП та ЕКП специфікації на характеристики розроблюваного за специфікацією ПЗ і зробити висновки щодо обґрунтованого вибору програмного проекту.

Перспективними завданнями для подальших досліджень є: 1) побудова нейромережного методу прогнозування характеристик програмного забезпечення на основі аналізу специфікацій (НМТПХ); 2) реалізація та навчання нейромережної складової НМТПХ; 3) розроблення інтелектуальної системи прогнозування характеристик програмного забезпечення на основі аналізу специфікацій. На вирішення цих завдань і будуть спрямовані подальші зусилля автора.

Література

1. IEEE 830-1998. Recommended Practice for Software Requirements Specifications – New York: IEEE, 1998
2. Говорущенко Т.О. Аналіз галузі оцінювання якості програмного забезпечення // Вісник НУ "Львівська політехніка" "Комп'ютерні системи та мережі" – Львів: Видавництво НУ "ЛП", 2013
3. Говорущенко Т.О., Красій А.В. Визначення характеристик та вибір моделі життєвого циклу програмного забезпечення на основі аналізу специфікацій // Вісник Хмельницького національного університету – Хмельницький: ХНУ, 2013. - №6, с.201-208
4. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика – М.:Горячая линия – Телеком, 2001 – 382 с.
5. Медведев В.С., Потемкин В.Г. Нейронные сети Matlab 6 / Под общей редакцией к.т.н. В.Г.Потемкина – М.: Диалог-Мифи, 2002. – 496 с.
6. Мищенко В.О., Поморова О.В., Говорущенко Т.А. CASE-оценка критических программных систем. В 3-х томах. Том 1. Качество / Под ред.. Харченко В.С. – Харьков: НАУ «ХАИ», 2012. – 201 с.
7. Поморова О.В., Говорущенко Т.О. Сучасні проблеми оцінювання якості програмного забезпечення // Радіоелектронні і комп'ютерні системи – Харків: НАУ «ХАІ», 2013 – № 5, с.319-327

УДК 004.382, 004.383

В.А. Мельник

Національний університет «Львівська політехніка»

АНАЛІЗ ТА ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЗАСТОСУВАННЯ АПАРАТНИХ ПРИСКОРЮВАЧІВ ДЛЯ ПОБУДОВИ РЕКОНФІГУРОВНИХ КОМП'ЮТЕРНИХ СИСТЕМ

Мельник В.А. Аналіз та дослідження ефективності застосування апаратних прискорювачів для побудови реконфігурованих комп'ютерних систем. В статті проведено порівняльний аналіз типів архітектури реконфігурованих комп'ютерних систем і досліджено ступінь впливу обчислювальної складності задачі на величину прискорення, що може бути отримане в комп'ютерній системі шляхом під'єднання до неї пристроїв реконфігурованої логіки залежно від способу їх під'єднання, а також запропоновано методику визначення доцільності застосування реконфігурованої логіки з позиції організації зв'язку між нею і комп'ютером.

Ключові слова: реконфігуровані комп'ютерні системи, реконфігурована логіка, реконфігурований прискорювач, архітектура комп'ютерних систем.

Форм. 5. Табл. 0. Рис. 5. Літ. 23.

Melnyk V.A. Analysis and study of hardware accelerators application efficiency for reconfigurable computer systems design. The paper provides comparative analysis of the reconfigurable computer systems architecture types and investigates degree of influence of computing complexity to the value of performance that can be achieved in the computer system with connection of reconfigurable logic depending on the method of connection. The method of determination of the reconfigurable logic application expediency regarding its interfacing with computer is proposed.

Key words: reconfigurable computer systems, reconfigurable logic, reconfigurable accelerator, computer systems architecture.

Equat. 5. Tabl. 0. Fig. 5. Ref. 23.

Мельник В.А. Анализ и исследования эффективности применения аппаратных ускорителей для построения реконфигурируемых компьютерных систем. В статье проведен сравнительный анализ типов архитектуры реконфигурируемых компьютерных систем и исследована степень влияния вычислительной сложности задачи на величину ускорения, которая может быть получена в компьютерной системе путем подключения к ней устройств реконфигурируемой логики в зависимости от способа их подключения, а также предложена методика определения целесообразности применения реконфигурируемой логики с позиции организации связи между ней и компьютером.

Форм. 5. Табл. 0. Рис. 5. Лит. 23.

Вступ

Постановка проблеми. Реконфігуровані комп'ютерні системи (РККС) набули сьогодні значного поширення, що пов'язано з можливістю досягнення ними високих показників продуктивності шляхом апаратного прискорення виконання обчислювальних завдань в кристалах реконфігурованої логіки, що входять до їх складу. РККС склали конкуренцію іншим типам високопродуктивних комп'ютерних систем завдяки високим технічним характеристикам сучасних кристалів програмовних логічних інтегральних схем (ПЛІС) – апаратній основі реконфігурованого середовища РККС, та досягненням у галузі технологій проектування спеціалізованих процесорів, що підлягають синтезу в реконфігурованому середовищі РККС.

Підключення синтезованих в реконфігурованому середовищі спеціалізованих процесорів, структура яких в тій чи іншій мірі враховує особливості виконуваних алгоритмів, до комп'ютерної системи на основі універсальних процесорів, дозволяє до 2-3 порядків підняти її продуктивність. Можливість реконфігурування та синтезу в реконфігурованому середовищі спеціалізованих процесорів з новою структурою та функціями дозволяє змінювати функціональну орієнтацію створеної таким чином РККС зі збереженням її високої продуктивності на нових класах задач.

Сьогодні існує велика кількість практичних реалізацій та декілька варіантів архітектури реконфігурованих комп'ютерних систем, які відрізняються методами під'єднання пристроїв реконфігурованої логіки та характеристиками використовуваних інтерфейсів.

Аналіз останніх досліджень і публікацій. Реконфігуровані комп'ютерні системи, їх будову та організацію функціонування на різних рівнях висвітлено в роботі [19]. В роботі [1] розглянуто питання створення високопродуктивної РККС на основі персонального комп'ютера та реконфігурованого прискорювача, побудованого на основі ПЛІС. В роботі [4] висвітлено типи архітектури РККС, сформульовано спосіб виконання обчислень в цих системах, а також визначено проблемні питання їх застосування та окреслено шляхи вирішення цих питань.

Невирішені частини проблеми. Поряд із значним обсягом літератури, яка висвітлює різні аспекти будови, проектування, застосування РККС, описує реалізацію різноманітних прикладних алгоритмів в РККС, сьогодні відсутні літературні джерела, в яких проведено ґрунтовний і цілісний порівняльний аналіз типів архітектури РККС та досліджено вплив обчислювальної складності задачі на величину прискорення, що може бути отримане в комп'ютерній системі шляхом під'єднання до неї пристроїв реконфігурованої логіки. Це в значній мірі стримує розвиток напрямку реконфігурованих обчислень в цілому.

Метою дослідження є проведення порівняльного аналізу типів архітектури РККС і

дослідження ступеня впливу обчислювальної складності задачі на величину прискорення, що може бути отримане в комп'ютерній системі шляхом під'єднання до неї пристроїв реконфігуровної логіки залежно від способу їх під'єднання, а також розроблення методики визначення доцільності застосування реконфігуровної логіки з позиції організації зв'язку між нею і комп'ютером.

1. Дослідження та порівняльний аналіз типів архітектури реконфігуровних комп'ютерних систем

Тривалість t_{exe}^P виконання програми відповідно до описаного в роботі [4] способу опрацювання інформації можна подати таким виразом:

$$t_{exe}^P = T_{UPP}(P_{UPP}) + T_{SP}(P_{RE}) - T_{UPP,SP}^{PAR} + ttr_{data}, \quad (1)$$

де: $T_{UPP}(P_{UPP})$ – тривалість виконання підпрограми P_{UPP} універсальним процесором UPP ; $T_{SP}(P_{RE})$ – тривалість виконання заданих підпрограмою P_{RE} обчислень спеціалізованим процесором SP , синтезованим в реконфігуровному середовищі РККС; ttr_{data} – тривалість пересилання даних між спеціалізованим процесором SP та універсальним процесором UPP чи пам'яттю під час виконання програми P_{in} ; $T_{UPP,SP}^{PAR}$ – час, на протязі якого універсальний процесор UPP та спеціалізований процесор SP виконують обчислення паралельно.

З виразу (1) випливає, що суттєвий вплив на значення тривалості t_{exe}^P виконання програми в РККС має її архітектура. По-перше, від типу використовуваного комп'ютера залежить тривалість $T_{UPP}(P_{UPP})$ виконання підпрограми P_{UPP} універсальним процесором UPP . По-друге, від технічних характеристик реконфігуровного середовища на основі ПЛІС та архітектури синтезованого в ньому спеціалізованого процесора SP залежить тривалість $T_{SP}(P_{RE})$ виконання заданих підпрограмою P_{RE} обчислень спеціалізованим процесором SP . По-третє, суттєвий вплив має якість виконаного користувачем розподілу навантаження між універсальним та спеціалізованим процесорами, в тому числі і на час $T_{UPP,SP}^{PAR}$, на протязі якого універсальний процесор UPP та спеціалізований процесор SP виконують обчислення паралельно. І по-четверте, від типу архітектури залежить значення тривалості пересилання даних між універсальним комп'ютером та реконфігуровним середовищем. Тому актуальним завданням є дослідження архітектури РККС та проведення порівняльного аналізу її типів.

Відповідно до описаних в роботі [4] базових структур комп'ютерних систем з реконфігуровною логікою, розглянемо відповідні їм типи архітектури РККС, а саме:

- архітектура РККС з слабозв'язаною реконфігуровною логікою;
- архітектура РККС з тіснозв'язаною реконфігуровною логікою, підключеною до шини пам'яті;
- архітектура РККС з тіснозв'язаною реконфігуровною логікою, підключеною до шини процесора;
- архітектура РККС з інтегрованою в АЛП універсального процесора реконфігуровною логікою;
- архітектура РККС з інтегрованими в реконфігуровну логіку універсальним і спеціалізованим процесорами.

РККС з інтегрованою реконфігуровною логікою характеризуються найвищою швидкістю взаємодії універсального процесора з спеціалізованим, оскільки ці процесори фізично розміщені в одному кристалі, а це дає можливість реалізувати між ними такі зв'язки, які забезпечать максимальну швидкість виконання обчислювального алгоритму. При цьому вища швидкість взаємодії досягається в тих РККС, в яких спеціалізований процесор використовується як один з операційних пристроїв універсального процесора, розширюючи його систему команд. Дещо нижчою є швидкість взаємодії в РККС, де універсальний процесор є вбудованим в реконфігуровне середовище і виконує функції керування спеціалізованими процесорами. Характеристики продуктивності такого процесора є невисокими, оскільки він повинен займати мінімум площі кристалу, а обмін інформацією між ним і спеціалізованими процесорами здійснюється через внутрішньокристалну системну шину. Для виходу на цю шину універсальний процесор

обладнано відповідним інтерфейсом, причому, якщо процесор є «жорстко» реалізованим в кристалі ПЛІС, тобто як замовний інтегральний елемент, так само «жорстко» реалізованою є й внутрішньокристална системна шина. Цим самим інтерфейсом обладнують і спеціалізовані процесори, що здійснюють на етапі їх проектування. По суті, цей підхід є реалізацією «комп'ютерної системи на програмовному кристалі» (англ. *SoPC – System on a Programmable Chip*). Зусиллями ряду фірм сьогодні створено відповідні інтерфейси, найпоширенішими з яких є *AMBA* [6] та *Wishbone* [23].

Принциповим недоліком архітектури РККС з інтегрованою реконфігуровною логікою є обмеження її ресурсів ємністю кристалу. Для нарощення ресурсів та підвищення продуктивності, а також надання РККС властивості масштабованості, до універсального процесора підключають зовнішню реконфігуровну логіку, що, власне, і є реалізацією тіснозв'язаної та слабозв'язаної типів архітектури. Зрозуміло, що швидкість взаємодії універсального процесора з спеціалізованим тут є значно нижчою.

Пристрої, на яких розміщують зовнішню по відношенню до універсального процесора реконфігуровну логіку, отримали назву реконфігуровних прискорювачів, оскільки їх можна під'єднати до довільного універсального комп'ютера, у тім числі й персонального, обладнаного сумісним інтерфейсом, з метою прискорення виконання ним обчислювальних завдань. Універсальний комп'ютер, до якого під'єднують реконфігуровний прискорювач, називають хост-комп'ютером.

Сучасний реконфігуровний прискорювач будують у вигляді однієї або декількох плат з розміщеними на них одним або декількома кристалами ПЛІС та одним або кількома роз'ємами для підключення зовнішніх пристроїв (плат розширення). Плати розширення підключають до базових плат через ці роз'єми. До плат розширення відносяться: плати, які підключаються до локальної шини базової плати, модулі *DIME (DSP and Image processing Module for Enhanced FPGAs)* [10] та плати введення-виведення (включаючи *WSDP*, волоконно-оптичні *Quad GLink*, *FPDP* і *A/D* плати).

2. Організація функціонування РККС з слабозв'язаною реконфігуровною логікою

В даній архітектурі прискорювач підключається до периферійної шини комп'ютера, зазвичай *PCI* чи *PCI Express*, *Ethernet* або *USB*. Застосування цієї архітектури вимагає оптимізації потоків вхідних та вихідних даних і є ефективним при їх малих об'ємах, оскільки шина легко перенавантажується, і пересилання даних до та з прискорювача може зайняти більше часу, ніж виконання алгоритму на центральному процесорі. Цей підхід є, проте, відносно дешевим.

Загальну схему РККС на основі універсального комп'ютера з слабозв'язаним прискорювачем зображено на рис. 1.

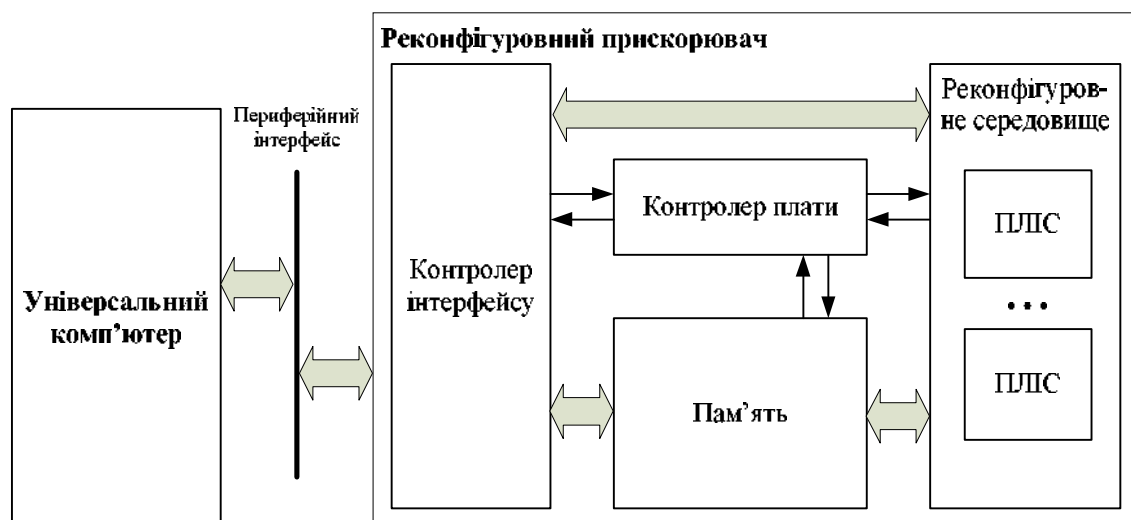


Рис. 1. Загальна схема РККС на основі універсального комп'ютера з слабозв'язаним прискорювачем

В архітектурі РККС на основі універсальних процесорів з слабозв'язаними прискорювачами в якості прискорювачів використовуються:

- змінні плати універсального комп'ютера, наприклад *ClearSpeed Advance X620* і *e620 Accelerator Boards* [8], або *NALLATECH H100 Series* [14], які взаємодіють з універсальним комп'ютером через один з периферійних інтерфейсів, найчастіше *PCI-Express*, *USB*, *Ethernet*;
- багатоплатні кластерні прискорювачі, побудовані на базі багатьох ПЛІС, наприклад, *COPACOBANA (Cost Optimized Parallel COde Breaker)* компанії *SciEngines* – кластерний прискорювач, побудований компанією *SciEngines* на базі багатьох ПЛІС, який характеризується невисокою вартістю [9] і обладнаний 120 кристалами ПЛІС *Xilinx Spartan 3-1000*, *HS21 BladeCenter* [18] компанії *IBM*, або суперкомп'ютер *Maxwell* на основі програмовних логічних інтегральних схем, розроблений альянсом *FHPCA* [12]. Такі прискорювачі також взаємодіють з універсальним комп'ютером через периферійний інтерфейс.

3. Організація функціонування РККС з тіснозв'язаною реконфігуровною логікою

Архітектура РККС на основі універсальних процесорів з тіснозв'язаними прискорювачами передбачає пряме під'єднання прискорювачів до процесорів. Яскравими представниками прискорювачів з тіснозв'язаною архітектурою є співпроцесори, які встановлюються в роз'єм процесора. Платформи комп'ютерів, які використовують зазначені процесори, характеризуються досить широкими можливостями інтеграції різних пристроїв. Після появи платформ з декількома процесорами стало можливим створення нових продуктів, що можуть бути встановлені в роз'єм процесора. На рис. 2 наведено схему підключення основних елементів платформи *AMD*, на якій зображено різні варіанти підключення прискорювачів на базі ПЛІС [5].

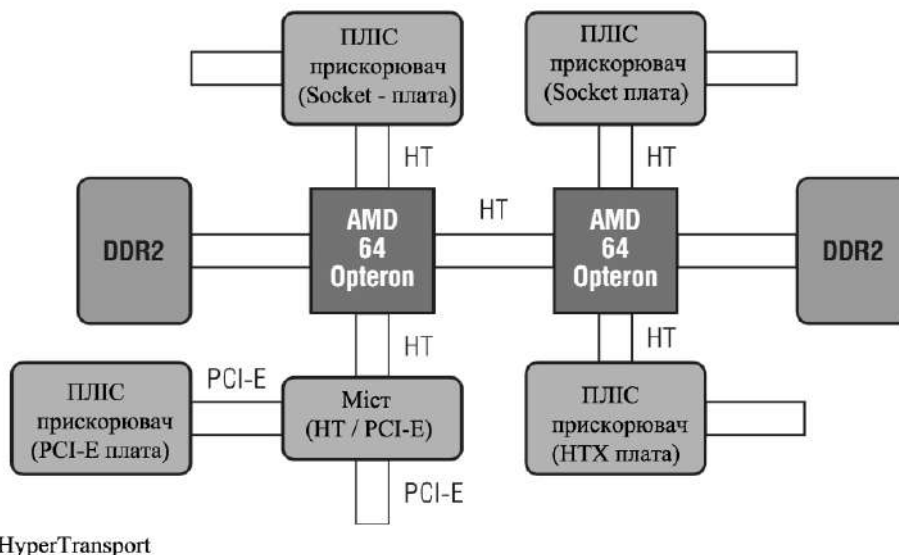


Рис. 2. Схема підключення прискорювачів до шини HyperTransport (платформа AMD)

Для даної платформи можливі наступні варіанти підключень:

- Плата з інтерфейсом *HT* (*HyperTransport* – двонаправлена послідовно/паралельна комп'ютерна шина з високою пропускнуною спроможністю і малими затримками) [15], яка безпосередньо встановлюється в процесорний роз'єм (на рисунку 2.7 позначена як *Socket-плата*).
- Плата з інтерфейсом *HT*, яка встановлюється в роз'єм *HTX* (*HyperTransport eXpansion* – специфікація слотного типу інтерфейсу, який має пряме з'єднання з мікропроцесором через *HyperTransport*).
- Плата з інтерфейсом *PCI Express*, яка під'єднується до роз'єму *HT* через міст *HT/PCI-Express*.

Прикладна програма взаємодіє з прискорювачем з допомогою ядра *HyperTransport*, що є сукупністю апаратних та програмних засобів для управління шиною та підключеними до неї пристроями. На програмному рівні ядро *HyperTransport* представлено драйвером, через який прикладна програма може керувати прискорювачем, а також здійснювати з ним обмін інформацією. У свою чергу доступ до даних прискорювач здійснює за допомогою контролера

прямого доступу до пам'яті.

На рис. 3 наведено схему для підключення основних елементів платформи *Intel* [5].

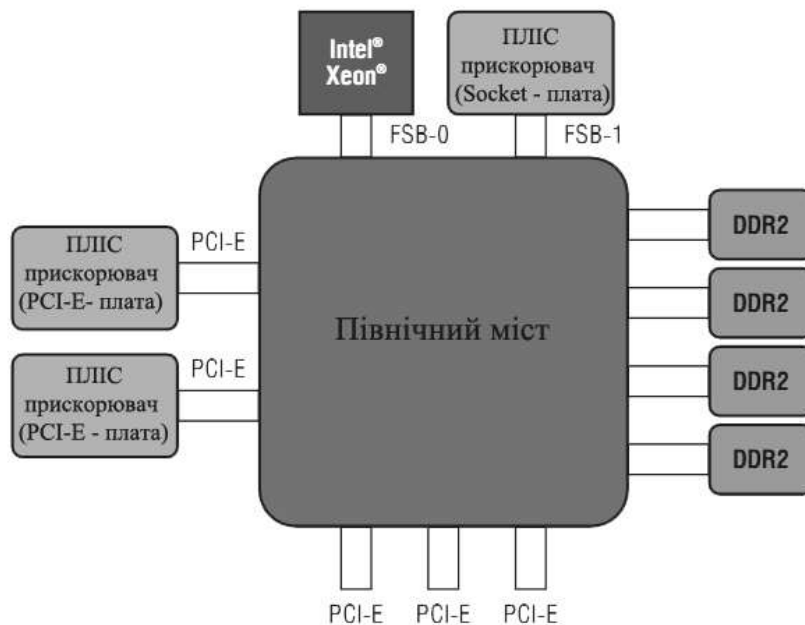


Рис. 3. Схема підключення прискорювачів у платформі Intel

Для даної платформи можливі наступні варіанти підключень:

- плата з інтерфейсом *FSB*, яка встановлюється безпосередньо в роз'єм процесора;
- плата з інтерфейсом *PCI Express*.

Прискорювачі з інтерфейсом *FSB* [13] встановлюються в роз'єм процесора платформи *Intel* [17]. *FSB* забезпечує високу швидкість та мінімальні затримки в порівнянні з іншими портами та шинами комп'ютера.

Є можливість підключення до комп'ютера не одного прискорювача, а декількох. В цьому випадку декілька модулів підключають між собою за кільцевою топологією через зовнішній, встановлений на платі, інтерфейс. Така схема підключення називається стековою, коли один модуль встановлюється поверх іншого, і забезпечує високий ступінь паралелізації, що є одним з ключових моментів у нарощенні продуктивності обчислювальної системи.

В архітектурі РККС на основі універсальних процесорів з тіснозв'язаними прискорювачами використовуються:

- тісноінтегровані в мережну структуру прискорювачі, зокрема реконфігуровні процесорні блоки *RP100* фірми *DRC* [11], реконфігуровний прискорювач *RCHTX* фірми *Celoxica* [7]. Тут передбачається пряме під'єднання прискорювачів до процесорів;
- тісноінтегровані в мережну структуру прискорювачі з прямим з'єднанням з мережними внутрішніми зв'язками. Ця архітектура також передбачає використання двох окремих мереж: процесорної та ПЛІС, але з їх накладанням, наприклад, з під'єднанням ПЛІС безпосередньо до мережного інтерфейсного кристала (*network interface chip – NIC*). Різниця тут полягає в тому, що ПЛІС під'єднані безпосередньо до основної мережі через *NIC*'и, що дає можливість одному процесору мати доступ до однієї або більше ПЛІС, тобто будь-яка ПЛІС може зв'язатися безпосередньо з будь-яким процесором. Прикладом такої архітектури є система *XDI* фірми *Cray* [22];
- прискорювачі з прямим з'єднанням з пристроями пам'яті. Тут прискорювачі з'єднані безпосередньо з системною шиною розподіленої пам'яті. Цим самим створюється мережа з топологією типу „кожен з кожним” і вимагається проведення диспетчеризації апаратних засобів для підтримки когерентності з'єднаних блоків пам'яті та пристроїв. Прикладом такої архітектури є архітектура *RASC (Reconfigurable Application Specific Computing)* [21] фірми *SGI* та її комунікаційна структура *NUMALink* [20];
- прискорювачі на основі архітектури *IMPLICIT+EXPLICIT* [16] фірми *SRC*, яка є гібридом вище згаданих типів архітектури.

Останні два типи архітектури передбачають з'єднання прискорювачів безпосередньо з системною шиною розподіленої пам'яті. Цим самим створюється мережа з топологією типу

“кожен з кожним” і вимагається високий рівень диспетчеризації апаратних засобів для підтримки когерентності з'єднаних блоків пам'яті та прискорювачів. Головним недоліком цього підходу є висока вартість.

4. Експериментальні дослідження характеристик реконфігурованих комп'ютерних систем залежно від складності задачі

Як показали висвітлені в [1, 2, 3] експериментальні дослідження слабозв'язаних та тіснозв'язаних реконфігурованих прискорювачів, організація зв'язку між ними і комп'ютером значній мірі визначає доцільного їх застосування в тій чи іншій області. Дослідження проводились на наступних платформах:

- процесор *Intel Celeron 2.0 ГГц*; ОЗП 1 Гб; ОС *Windows XP Professional SP2*;
- процесор *AMD Athlon 64XP 2.0 ГГц*; ОЗП 1 Гб; ОС *Windows XP Professional SP2*.

Для оцінювання слабозв'язаної архітектури використано прискорювач *XPRESSFX100-11* від компанії *PLDA* на ПЛІС *Virtex4 FX100* компанії *Xilinx*. Для оцінювання тіснозв'язаної архітектури використано таку саму ПЛІС *Virtex4 FX100* компанії *Xilinx* з розрахунком її підключення до високошвидкісної шини пропускнуою здатністю 16 Гб/с, яка при досліджуваних об'ємах даних не створює затримки передачі. В ПЛІС прискорювача реалізовано контролер інтерфейсу *PCI-Express* та спеціалізований процесор виконання тестового алгоритму.

Оцінювання було виконано на алгоритмах:

- перетворення *RGB* → *YUV*;
- цифрової фільтрації;
- дискретного косинусного перетворення (ДКП);
- швидкого перетворення Фур'є (ШПФ) на 1024 точки.

Апаратні витрати на реалізацію спеціалізованих процесорів в ПЛІС прискорювача склали:

- перетворення *RGB* → *YUV* – 22% площі ПЛІС;
- цифровий фільтр – 23% площі ПЛІС;
- дискретне косинусне перетворення – 29% площі ПЛІС;
- швидке перетворення Фур'є – 35% площі ПЛІС,

з яких 17% припадає на реалізацію контролера *PCI-Express*.

Передавання даних на прискорювач *PLDA XPRESSFX100-11* відбувається по 8 байт. Реалізовані в прискорювачі спеціалізовані процесори мають таку кількість вхідних та вихідних портів:

- перетворення *RGB* → *YUV* – 3 вхідних та 3 вихідних порти;
- цифровий фільтр – 4 вхідних, 4 вихідних;
- ДКП – 8 вхідних, 8 вихідних;
- ШПФ-1024 – 2 вхідних, 2 вихідних.

Обмін даними з прискорювачем було організовано декількома каналами. На графіках нижче (рис. 4 і 5) наведено залежності часу виконання зазначених алгоритмів на слабозв'язаному (рис. 4) та тіснозв'язаному (рис. 5) прискорювачах від обсягу вхідних даних при їх передачі одним каналом та декількома.

З наведених графіків видно, що частка тривалості пересилання даних є визначальною при значних об'ємах цих даних, причому при виконанні деяких алгоритмів на слабозв'язаному прискорювачі коефіцієнт прискорення $\epsilon < 1$ (тобто спостерігається програв в продуктивності порівняно з універсальним процесором). Це стосується перетворення *RGB* → *YUV*, причому у випадку передачі даних одним каналом та трьома, та алгоритму цифрової фільтрації у випадку передачі даних одним каналом. Для алгоритму ШПФ-1024 коефіцієнт прискорення становить 250 разів для тіснозв'язаної архітектури проти 10 разів для слабозв'язаної архітектури, а для алгоритму ДКП – 120 проти 2,5.

Виходячи з наведених вище результатів експериментальних досліджень виглядає доцільним перед вибором алгоритмів, які мають бути реалізовані в прискорювачі, виконати попереднє оцінювання ефективності такого рішення. Розглянемо далі як це можна зробити.

5. Методика визначення доцільності застосування прискорювача в реконфігурованій комп'ютерній системі

Під час визначення доцільності застосування прискорювача для виконання деякої програми P_{in} , яка передбачає опрацювання даних об'ємом Q_{data} , необхідно дослідити виконання наступного

співвідношення:

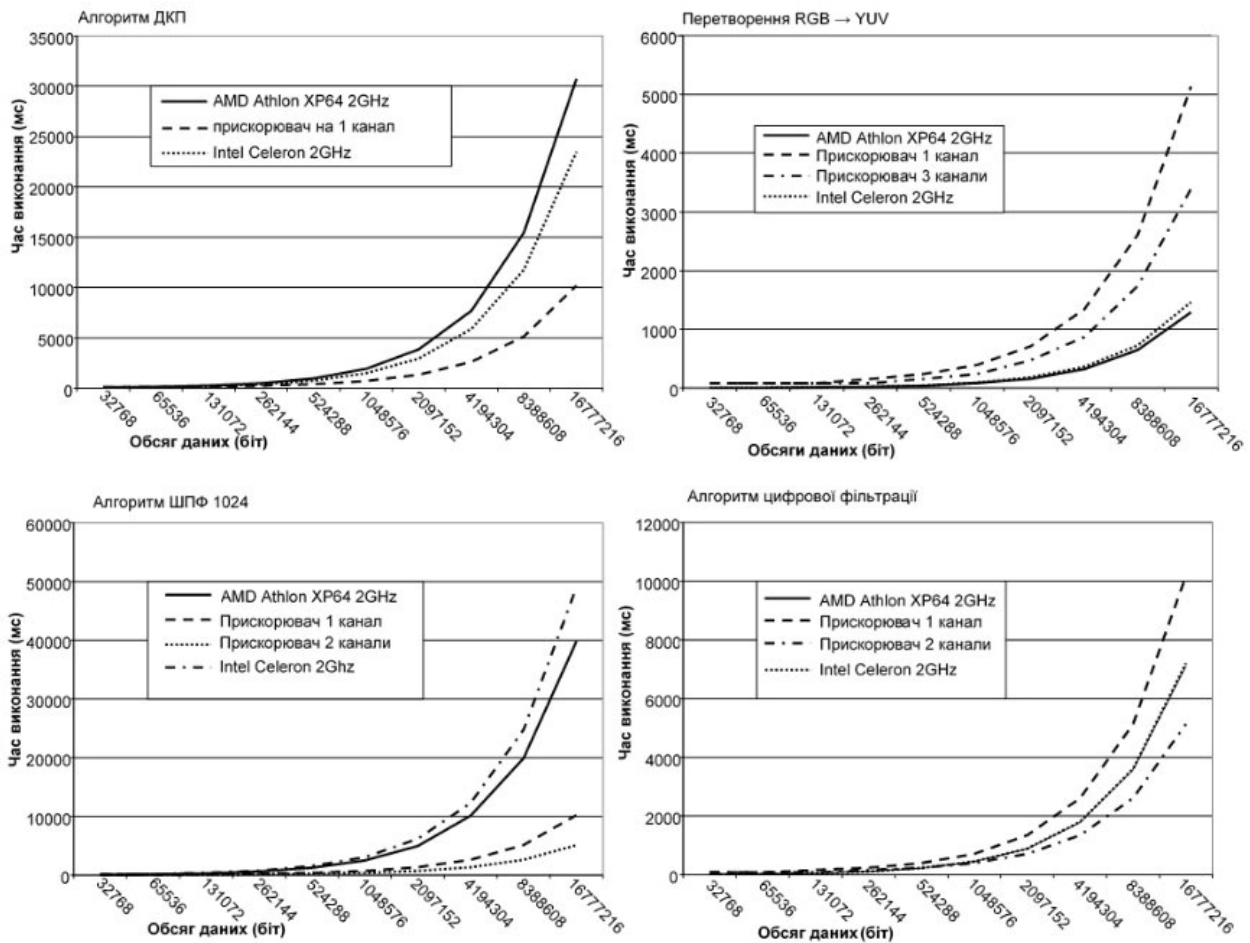


Рис. 4. Графіки залежності часу виконання алгоритмів на слабозв'язаному прискорювачі від обсягу вхідних даних при їх передачі одним каналом та декількома

$$\frac{Q_{data}}{V_{int}} < T_{UPP}(P_{in}) - T_{UPP}(P_{UPP}) + T_{SP}(P_{RE}), \quad (2)$$

де Q_{data} – кількість даних, що пересилаються між універсальним комп'ютером та прискорювачем; V_{int} – пропускна спроможність інтерфейсу, через який ці дані пересилаються; $T_{UPP}(P_{in})$, $T_{UPP}(P_{UPP})$ – тривалість виконання, відповідно, програми P_{in} та підпрограми P_{UPP} універсальним процесором UPP ; $T_{SP}(P_{RE})$ – тривалість виконання заданих підпрограмою P_{RE} обчислень спеціалізованим процесором SP , синтезованим в реконфігурованому середовищі РККС.

Тривалість виконання програм P_{in} та P_{UPP} на універсальному комп'ютері можна визначити, знаючи характеристики обчислювальної платформи, з наступного виразу:

$$T_{UPP}(x) = \sum_{i=1}^{IQ(x)} \left(\frac{qclk(I_i)}{F_{UPP}} \right), x \in (P_{in}, P_{UPP}), \quad (3)$$

де: $qclk(I_i)$ – кількість тактів виконання i -ї інструкції тієї вітки програми, яка має найвищу програмну складність; IQ – кількість інструкцій у вітці програми з найвищою програмною складністю; F_{UPP} – тактова частота роботи процесора універсального комп'ютера.

Тривалість $T_{SP}(P_{RE})$ виконання заданих підпрограмою P_{RE} обчислень спеціалізованим процесором SP можна визначити на основі отриманих від засобів логічного синтезу даних про максимальну тактову частоту F_{SP} роботи цього процесора, та знаючи розрядність DW_{SP} слів даних, які він обробляє і кількість $qclk(SP)$ тактів, необхідних для їх обробки (ці дані

визначаються на етапі створення програмної моделі процесора):

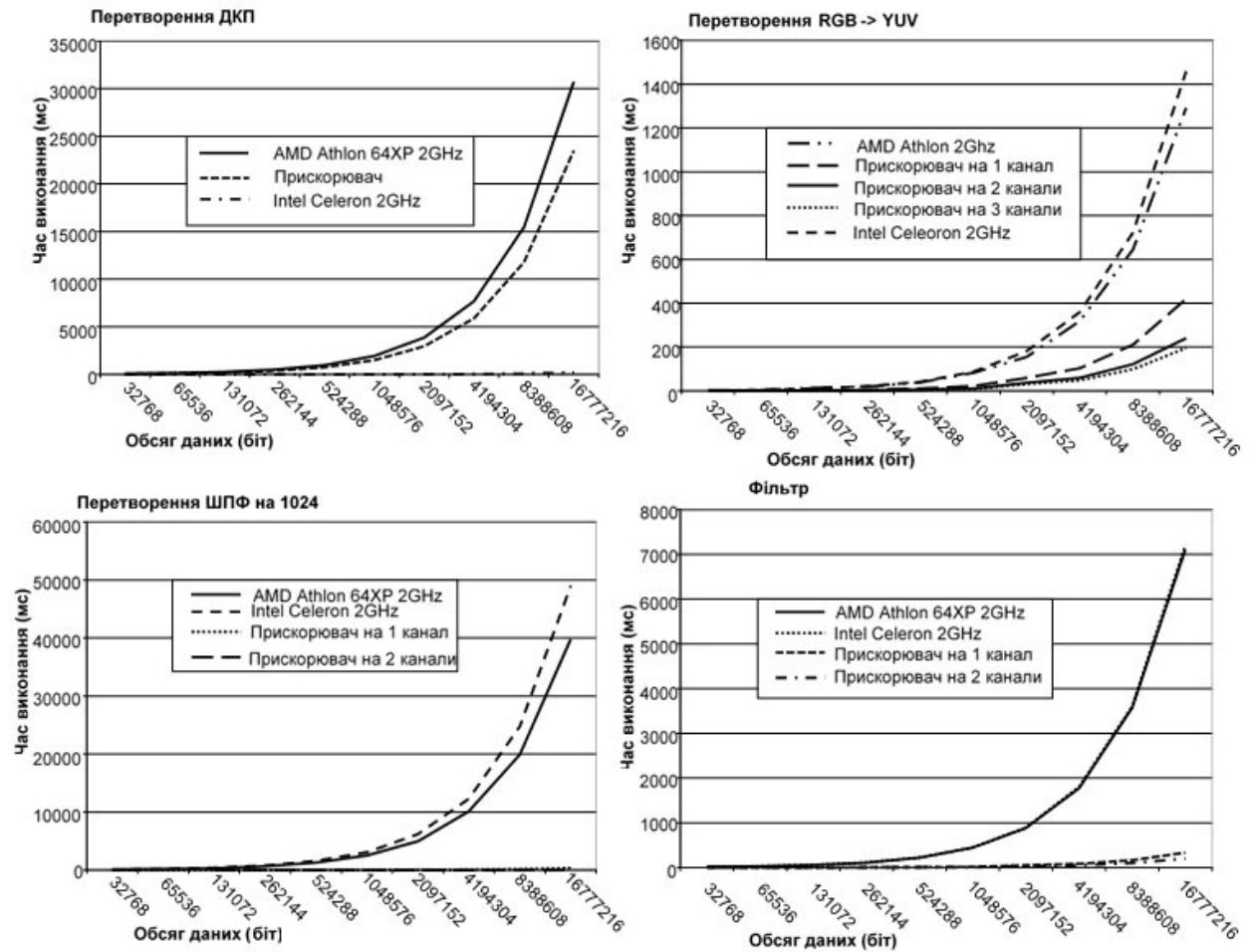


Рис. 5. Графіки залежності часу виконання алгоритмів на тіснозв'язаному прискорювачі від обсягу вхідних даних при їх передачі одним каналом та декількома

$$T_{SP}(P_{RE}) = \frac{qclk(SP)}{F_{SP} \cdot DW_{SP}} \quad (4)$$

Підставивши дані у вирази (3) і (4), можна обчислити коефіцієнт ефективності K_E застосування прискорювача для виконання програми P_{in} з позиції організації зв'язку між ним і комп'ютером:

$$K_E = \frac{Q_{data}}{V_{int} \cdot \left(\sum_{i=1}^{IQ(P_{in})} \left(\frac{qclk(I_i)}{F_{UPP}} \right) - \sum_{i=1}^{IQ(P_{UPP})} \left(\frac{qclk(I_i)}{F_{UPP}} \right) + \frac{qclk(SP)}{F_{SP} \cdot DW_{SP}} \right)}, \quad (5)$$

Таким чином, відповідно до запропонованої методики, визначення доцільності застосування прискорювача передбачає виконання наступних кроків:

1. визначення тривалості $T_{UPP}(P_{in})$ та $T_{UPP}(P_{UPP})$ виконання програм P_{in} та P_{UPP} на універсальному комп'ютері за виразом (3);
2. визначення тривалості $T_{SP}(P_{RE})$ виконання заданих підпрограмою P_{RE} обчислень спеціалізованим процесором SP за виразом (4);
3. обчислення коефіцієнта ефективності K_E застосування прискорювача для виконання програми P_{in} за виразом (5).

Значення коефіцієнта ефективності повинно знаходитись в межах $0 < K_E < 1$, причому чим меншим буде це значення, тим вищою буде ефективність застосування прискорювача для виконання програми P_{in} .

Висновки

Проведено дослідження та порівняльний аналіз типів архітектури РККС. Показано, що для нарощення ресурсів та підвищення продуктивності, а також надання РККС властивості масштабованості, РККС будують шляхом під'єднання до процесора зовнішніх по відношенню до нього пристроїв реконфігуровної логіки, що є реалізацією тіснозв'язаної та слабозв'язаної типів їх архітектури. і обґрунтовує застосування реконфігуровних прискорювачів для побудови РККС.

Сформульовано основи організації функціонування РККС з слабозв'язаною реконфігуровною логікою та проведено оцінювання технічних характеристик існуючих реконфігуровних прискорювачів цих РККС.

Сформульовано основи організації функціонування РККС з тіснозв'язаною реконфігуровною логікою, проведено оцінювання технічних характеристик існуючих реконфігуровних прискорювачів цих РККС, а саме: тісноінтегрованих в мережну структуру прискорювачів, тісноінтегрованих прискорювачів з прямим під'єднанням до мережного інтерфейсного кристалу, прискорювачів з прямим з'єднанням з пристроями пам'яті та гібридних прискорювачів.

З метою визначення областей доцільного застосування слабозв'язаних та тіснозв'язаних реконфігуровних прискорювачів з позиції організації зв'язку між ними і комп'ютером проведено експериментальні дослідження характеристик РККС залежно від складності задачі та встановлено,

що при виконанні тестових алгоритмів з обчислювальною складністю $O\left(\frac{N}{2} \cdot \log N\right)$ на базі слабозв'язаної архітектури досягається прискорення на один порядок, а на базі тіснозв'язаної – на два порядки. У випадку виконання обчислювально складніших алгоритмів показники прискорення вищі. Разом з тим дослідження показали, що алгоритми, які характеризуються невисокою обчислювальною складністю та інтенсивним потоком даних, на зразок перетворення $RGB \rightarrow YUV$, на базі слабозв'язаної архітектури виконувати недоцільно.

Показано методику визначення доцільності застосування прискорювача з позиції організації зв'язку між ним і комп'ютером.

1. Мельник А.О. Використання реконфігуровних прискорювачів для підвищення продуктивності персональних комп'ютерів / А.О. Мельник, В.А. Мельник, З.Т. Сарайрех // Науковий вісник Чернівецького ун-ту. Комп'ютерні системи та компоненти. – Чернівці: Чернівецький національний університет імені Юрія Федьковича, 2010. - Т.1 Вип.1. – С. 20-25.
2. Мельник А.О. Реалізація програмних спеціалізованих процесорів в реконфігуровних прискорювачах універсальних комп'ютерів / А.О. Мельник, В.А. Клименко, Л.О. Цигилик, А.В. Юрчук // Вісник НУ "Львівська політехніка" "Комп'ютерні системи та мережі", –2009. –№ 609. –С. 69-77
3. Мельник А.О. Хамелеон – система високорівневого синтезу спеціалізованих процесорів / А.О. Мельник, В.А. Клименко, Л.О. Цигилик, А.В. Юрчук // Радіоелектронні і комп'ютерні системи, Харків «ХАІ». –2009. –С. 189-193.
4. Мельник В.А. Основи побудови та організації функціонування реконфігуровних комп'ютерних систем / В.А. Мельник // Науковий журнал «Вісник Хмельницького Національного Університету», Серія «Технічні науки». – №6. – 2012 р. – С. 212 – 217.
5. Шматок А. Аппаратные ускорители приложений на базе ПЛИС / А. Шматок // Современная электроника, -№6, 2007. – С. 60-63.
6. AMBA Specification (Rev 2.0). ARM Limited, 1999. – 230 p.
7. Celoxica Ltd. RCHTX-XV4 High Performance Computing (HPC) Application Acceleration Board Datasheet. Version 1.0. 2006 [Електронний ресурс] // – Режим доступу: http://www.hypertransport.org/docs/tech/rchtx_datasheet_screen.pdf
8. ClearSpeed Technology. ClearSpeed Advance X620 Accelerator Card User Guide. // Document No. 60-UG-1302 Revision: 3.E September 2008. -16p. [Електронний ресурс] // – Режим доступу: http://support.clearspeed.com/resources/documentation/X620_Advance_Card_User_Guide_Rev3.E.pdf
9. СОРАСОВАНА – Special Purpose Hardware for Code Breaking [Електронний ресурс] // – Режим доступу: <http://www.copacobana.org>

10. Devlin Malachy. DIME – The First Module Standard for FPGA Based High Performance Computing / Malachy Devlin, Allan J. Cante // FPL '99 Proceedings of the 9th International Workshop on Field-Programmable Logic and Applications. Springer-Verlag, London, UK, 1999. – P. 425 – 430.
11. DRC Computer Corporation. RPU100-L60 DRC Reconfigurable Processor Unit. A breakthrough in coprocessor technology [Електронний ресурс] // – Режим доступу: http://www.drccomputer.com/pdfs/DRC_RPU100_datasheet.pdf
12. FHPCA Supercomputer: Maxwell [Електронний ресурс] // – Режим доступу: <http://www.epcc.ed.ac.uk/projects-portfolio/fhpca-supercomputer-maxwell>
13. FPGA Accelerated Computing Solutions. Intel Xeon Front Side Bus [Електронний ресурс] // – Режим доступу: http://www.nallatech.com/images/stories/product/facs/fsb-accelerator-module/FSB_Accelerators_v1-5.pdf
14. H100 Series FPGA Application Accelerators. Version 1.9. September 2008 // [Електронний ресурс] // – Режим доступу: <http://www.skyblue.de/nallatech/5595.pdf>
15. HyperTransport Consortium. "HyperTransport Link Specification", 2005.
16. IMPLICIT+EXPLICIT ARCHITECTURE [Електронний ресурс] // – Режим доступу: <http://www.srccomp.com/techpubs/implicitexplicit.asp>
17. Intel Xeon Processor E5 Family [Електронний ресурс] / – Режим доступу: http://www.intel.com/products/processor/xeon5000/index.htm?iid=servproc+body_xeon5000
18. International Business Machines Corporation. BladeCenter HS21 Type 8853 Installation and User's Guide, Fourth Edition (August 2007) - 66p. // [Електронний ресурс] // – Режим доступу: http://publib.boulder.ibm.com/infocenter/bladectr/documentation/topic/com.ibm.bladecenter.hs21.doc/hs21_install_ug.pdf
19. Scott Hauck, André DeHon. "Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation"/ Morgan Kaufmann, 2008. – 944 p.
20. SGI NUMALink Interconnect Fabric [Електронний ресурс] // – Режим доступу: <http://www.sgi.com/products/servers/numalink.html>
21. Silicon Graphics, Inc. Reconfigurable Application-Specific Computing User's Guide. 2004-2006, 196p. [Електронний ресурс] // – Режим доступу: <http://techpubs.sgi.com/library/manuals/4000/007-4718-004/pdf/007-4718-004.pdf>
22. Ulmer, Craig D., Hilles, Ryan, Thompson, David C. Reconfigurable computing aspects of the Cray XD1. Presentation at the Cray User Group (CUG 2005), May 16-19, 2005. Albuquerque, USA. -11p. [Електронний ресурс] // – Режим доступу: http://www.craigulmer.com/portfolio/unlocked/050516_cug_rc_aspects_of_xdl.pdf
23. Wade D. Peterson, Silicore Corporation. Specification for the WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores. Revision: B.3, Released: September 7, 2002. – 140 p.

УДК 004.002

Melnyk V.M., Melnyk K.V., Zhyharevych O.K.

Lutsk National Technical University

NETWORK EMULATION FOR JAVA-BASED APPLICATIONS THROUGH SOCKET FACTORIES

Melnyk V.M., Melnyk K.V., Zhyharevych O.K. Network emulation for java-based applications through socket factories. Network emulation discussed in the article provides the capability of evaluating distributed applications on a stand-alone system. Applications can be exposed to adverse repeatable network conditions without requiring complex testbeds. This paper describes the design and implementation of a portable and object-oriented network emulator targeted to the development and test of Java-based Internet applications. The emulator is based on instrumented sockets is able to emulate the behavior of links with a given bandwidth and communication delay. The emulator is organized modularly, so that it is possible to plug-in user-defined classes for bandwidth and delay figures generation. Carrying out experiments with emulated sockets is as simple as running the tested application code on a single host.

KEYWORDS: Network, emulator, Internet application, sockets, Java-development.

Fig. 1., Ref. 6.

Мельник В.М., Мельник К.В., Жигаревич О.К. Мережева емуляція для java-додатків за допомогою об'єднань сокетів. Емуляція мережі забезпечує можливість оцінки розподілених додатків в автономній системі. Додатки можуть мати схильність виявлення умов повторності мережі, не вимагаючи складних тестових систем. Описується розробка і реалізація портативного об'єктно-орієнтованого емулятора мережі, орієнтованого на створення і тестування додатків для Інтернет, розроблених на Java. Емулятор заснований на інструментальних сокетах і здатний емулювати поведінку зв'язків із заданою пропускнуою і затримуючою здатністю. В організованій за модульним принципом, так що можна підключити його в класах користувачів для генерації пропускнуої здатності та залежностей затримки. Проведення експериментів з подібними сокетами є простим, як і виконання тестового коду програми на одному хості.

КЛЮЧОВІ СЛОВА: мережа, емулятор, Інтернет-додаток, сокети, розробка в Java.

Рис. 5., Літ. 7.

Мельник В.М., Мельник К.В., Жигаревич О.К. Сетевая эмуляция для java-приложений с использованием сокетных объединений. Эмуляция сети обеспечивает возможность оценки распределенных приложений в автономной системе. Приложения могут иметь склонность изъятия условий повторности сети, не требуя сложных систем тестирования. Описывается разработка и реализация портативного объектно-ориентированного эмулятора сети, ориентированного на разработку и тестирование приложений интернет, разработанных на Java. Эмулятор, созданный на инструментальных сокетах и способный эмулировать поведение связей с заданной пропускной и задерживающей способностью. Он организован за модульным принципом, так что можно подключать его в классах пользователей для генерации пропускной способности и зависимостей задержки. Проведение экспериментов с описанными выше сокетами простое, как и исполнение тестового кода программы на одном хосте.

КЛЮЧЕВЫЕ СЛОВА: сеть, эмулятор, Интернет-приложение, сокет, разработка в Java.

Рис. 5., Лит. 7.

Introduction

The amount of Internet-oriented applications has increased during recent years. This process concerns area of the applications development and the re-building area of many Internet enabled applications needed to be in Internet in order to have their lives prolonged. From the other site, not so many tools have been introduced to support such application development. In many cases, if the final product is developed in an Internet setting, the phases of development and testing are carried out on a single machine and sometimes inside a LAN. However, bandwidth and communication delay differ of some magnitude orders when operating on a single host or in a LAN, rather than through links with long hauls or with dial-up connections. Some big difference between the deployment and the development environment is likely to hide design issues affecting system performances that can be relevant for the user. In this case, because of differences in two environments the debugging phase could be less effective. For the matter let us to implement a complex user interface as an Applet, interacting with a server by means of remote method invocation, the Java object oriented version of RPC. It would be useful to test interface responsiveness and friendliness of use in its operative environment before deploying the product. To cope with this problem, we can test either the application directly on the field or using a network emulator because of sensitivity RPC-based programs to network delay. The solution of course has some drawbacks and advantages: the effectiveness of an application testing in the real operative environment is counterly balanced by the difficulty of setting some testbed. Some different problem is that developer is not able to reproduce or control the test conditions. The guarantee of emulation is a simple set up of experiments and their repeatability. It can be for good solution if it is not required a big amount of details.

A simple flexible network emulator for Java applications is presented for messages intercepting between the application and the network API. It also manipulates the communication to behave as in the network presence with delay and limited bandwidth. The emulator is based on a customized socket (emulation socket) and able to reproduce the Internet links behavior with characteristics of delay and bandwidth for defined user. These customized sockets can operate on a single workstation, and used to test and debug an Internet application on a separated system.

Motivated work

Every time we try to except some desirable properties for network emulator: portability, ease of use, and configurability. They become really needed to study Internet applications by using the network emulator. The requirement of portability is particularly significant in considering that Internet applications are to be deployed on different hardware or software platforms. In case of using a specific operation system or hardware it could be particularly bothered.

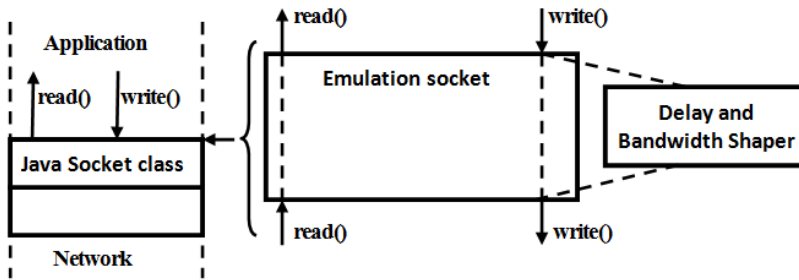


Fig. 1 – The emulation socket layer operates between the application level and the socket level

The emulator for network must be easy in use and highly configurable. In the purpose to observe the application's behavior, the programmer should be required to specify the parameters of communication only, and do not to deal with specific tools of operation system or details of network protocols.

Some network emulators and traffic shapers are described in several articles. The network simulator [1] has

been rebuilt for enhancing of emulation facilities [2]. The real traffic can be injected into the simulator for studying of the experimental algorithms behavior during operating with real traffic or testing a protocol implementation with reproducible conditions. The network simulator is a powerful tool with providing of wide variety of protocols for wired and wireless networks. However, network simulator is more targeted to test and estimate protocols with real-traffic sources, but no applications. If there is a want to use network simulator to study the applications behavior in one network conditions, the phase of setup is quite complex. So, few machines may need to run the experiment, one machine is acting as the emulator and the others executing the nodes of application.

Running this experiments is some easier with a simple but effective network emulator [3] which allows to run experiments on a autonomous system. Here the communication between the protocol layers is intercepted to be analyzed. It emulates the real network presence by introducing queues with finite size, bandwidth limitation, delay of communication and loss links. Running on FreeBSD the implementation, operates at the interface between TCP and IP. It suffers from dependency of operating system. It also is not so easy to run experiments with user defined time and varying network conditions. Practically, each time it is needed to change the conditions of network and the system must be reconfigured by means of the command line or some shellscript.

The wide-area network emulation tool is described in [4] running on UNIX machines. A set of hosts are adopted by delay line and connected by a LAN, running a distributed application can appear as if connected through long-haul links with adjustable communication characteristics. But the tool is only able to introduce artificial delay in the communication between two or more processes and is impossible to control the bandwidth. Another network emulator is supporting applications using UDP, is described in [5]. It is intended to facilitate the adaptive application development of operating in wireless networks and subjected to wide network fluctuations.

To accomplish the requirements discussed above here is designed a portable, Java-based object oriented emulator, which can be simply extended by adding user-defined classes to the emulator package. The work has been influenced by both methods described above. The idea of inserting has been taken from these to configure the layer that emulates the network. In the system the tools for the software stack are intercepted to communicate between the application and the socket APIs used to access the network. The reasonable for this choice is that in Java is possible to use only application level APIs for preserving programs portability.

Assumptions and functionality

A Java-based distributed application for Internet runs on top of a host set interconnected by long-haul links. Each host runs one or more JVMs that represent the application nodes. In accordance with these assumptions, the system is modeled as a directed graph made of nodes and links. They are defined as follows:

The *node* is as a JVM instance which is running a component of application and different JVMs running on the same host are considered as different nodes. The *link* for communication is as a both-ended uni-directional channel between nodes. This channel is intended at the level of *application* and everything under the socket APIs is considered as a black-box with given properties of bandwidth and delay. This presumption has some good impact on the design of our system. The link delay represents the latency from the time the sender node writes a data set; to the time those data become available for a read at the destination node. The single-directional links allow defining asymmetric inter-node communication behavior.

The aim of the tool is to support the execution of all application nodes on a single machine, or in a LAN, emulating the behavior of an Internet environment, where the emulated network configuration is defined through a directed graph like it was above. This functionality can be done by associating each link with a special customized socket able to shape the traffic over the link combined with user's requirements. The way these sockets manipulate the communication behavior saves up the blocking and/or non-blocking semantics of socket read and write operations.

Links can be parameterized in term of delay and bandwidth, in the following ways: *Fixed* way means that the delay and/or the bandwidth of the link does not change during application's lifetime; *Trace-driven* way gives the figures of delay and/or bandwidth which are expressed by a set of samples included in a file. Each sample is hold for numbered

seconds given as specified in the system file of configuration. This option is particularly useful to feed the emulator with real network delay and bandwidth traces (collected as the ping utility).

User-defined way contains some pre-defined interfaces which allow us to build user-defined classes that form delay and/or bandwidth figures. Such classes implement a method that generates delay on the fly and values of bandwidth, respectively, which are used by the emulation socket associated to a link for manipulating its communication behavior according to the desired figure. So, developers can specify link parameters as random variables with customized mean value, disperse, etc.

Emulation Sockets

An Emulation Socket is acting as a filter. The application writes some data into the socket each time, and the data flow is aroused according to given bandwidth and delay characteristics. Fig. 1 shows emulation sockets inserting between the application layer and the java.net.Socket system class. Each method call to write() is intercepted by the emulation socket layer and forwarded to the network according to a shaping policy defined in an external module. Shaping is performed only on the stream of the outgoing data, and no operation is performed on the incoming stream of the data. This means that dual directional communication is formed always on the side of the sender.

Each emulation socket is characterized by D and B parameters, representing the delay and the bandwidth associated to the link, respectively. D and B are functions depended of time.

To support traffic shaping, each emulation sockets includes the following structures of the data and threads (fig. 2). *PendingData* is a data structure used to emulate the delay, where data written by the application are stored. To preserve sequential ordering of socket-based streams, PendingData is a circular array managed with FIFO ordering. *SendingData* is a structure of the data used to emulate the bandwidth. Mover is a thread which extracts data contained in Pending Data at the appropriate time, as described below, and inserts them into Sending Data. Sender is a thread which extracts data, with FIFO order, contained in Sending Data and inserts them into the network through the plain socket.

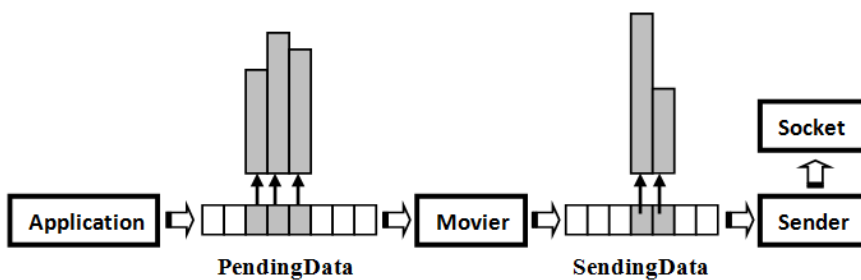


Fig. 2 – Emulation socket structure

Data are extracted at a B rate.

The emulation socket behaves as follows:

- If an application, at time t_1 , writes some bytes into an emulation socket, then data are inserted in Pending Data together with their extraction time ($t_1 + D$), so data remains in Pending Data for a time not lower than D.
- The Mover is thread which sleeps until the next extraction time ($t_1 + D$). It

moves *all* data having an extraction time lower than $t_1 + D$ from Pending Data to Sending Data. To get some clearance in how the data order is preserved, lets suppose that another write-operation is performed at time $t_2 > t_1$. This may happen when data written at time t_2 are subject to a lower delay, and as a consequence they may have an extraction time lower than $t_1 + D$. However, since streams associated to sockets have FIFO semantics, data written at time t_2 may be delivered to the application only after all previous data has been delivered, that is at $t_1 + D$ time.

- The Sender thread wakes up and in Every T seconds extracts from Sending Data an amount of bytes $A = B \cdot T$ (the bandwidth accumulated during the time of sleeping) and inserts them into the network.

The emulation socket class is defined as a subclass of the java.net.Socket system class. Since it inherits all methods and fields from the Socket class, it can replace plain sockets in an application with little changes. For RMI-based applications, the developer can use Java socket factories to develop the instrument code [6][7]. Socket factories allow a client-server application to communicate by using customized sockets. When the application creates a socket, it actually delegates the creation to the installed socket factory. The factory returns an instance of socket which can implement a behavior that is different from the standard one, while preserving the same interface towards the application. For example, the socket factory can return sockets for data compressing or encrypting, or using a protocol different from TCP.

Using emulation socket with RMI-based applications is straightforward. Instead of using the standard constructor of a remote object, the programmer has to create the object by means of an alternative constructor which lets her specify the socket factories provided by developed package: EmuSocketClientFactory and EmulationServer-SocketFactory. The first of them is used by the client code to establish a server connection; the latter is used by the server code to accept incoming connections. This procedure requires the availability of source code.

A more combined instrumentation is required by those applications that directly use sockets. Each creation of socket must be replaced by the creation of an emulation socket class instance. Now we are investigating on how to use the SocketImplFactory pattern. This would require a couple of code lines to be added to the application and leaving unchanged the rest of the code. To deal with the problem of source code unavailability, we developed another version of the emulation socket package which exchanges the java.net.Socket class of the Java Runtime Environment with the

```
#Node section
node node0 127.0.0.1 ports from 1000 to 1999
node node1 127.0.0.1 ports from 2000 to 2999
node node2 127.0.0.1 ports from 3000 to 3999

#Link section
link from node0 to node1 bandwidth 1000B/s
link from node0 to node2 delay 200ms bandwidth
1000B/s
link from node1 to node0 bandwidth class:
MyBandwidthShaper
link from node1 to node2 delay file:
node1-node2-delay.data seconds 1
link from node2 to node0 delay 150ms
link from node2 to node1 delay file:
node2-node1-delay.data seconds 1
```

Fig. 3 – A network configuration file example

shaping version. For the purpose of executing the application, the JVM must be launched with the -Xbootclasspath option. This way the JVM settings for the system classes' location are overridden.

Network configuration is collected in a file shared by all JVMs. Syntax of the network configuration file is straight, as shown in fig. 3. The file of configuration is composed of a node section and a link section. The node section is used to associate an IP address and a range of port numbers to each node. The reason is that, when the application creates a new socket, the emulator must identify the target node, return an instance of emulation socket which has the specified bandwidth and delay properties. For example, when running all JVMs on the same workstation, all nodes share the same

address. But since they use different port ranges, the emulator can still identify the correct node. When the application is started, each JVM must be launched setting a special property (node-Name) to indicate to the JVM the node name which is associated with.

Fig. 3 shows the three nodes application configuration. The three JVMs must be launched on the same host, with the nodeName proper to node0, node1, and node2, respectively. The first one will use ports numbered from 1000 to 1999, the second one will use port numbers from 2000 to 2999, and so on. The link section describes delay and bandwidth characteristics of two-end links, e.g. link from node0 to node2 is characterized by constant delay and bandwidth respectively equals to 200 ms and 1000 B/s; the bandwidth link values from node1 to node0 are generated by MyBandwidthShaper class; communication delay from node1 to node2 is conveyed by samples contained in a file, and each sample must be in 1 second hold.

When two Java programs, a client and a server, communicate by means of emulation sockets, the following is happening:

1. The Application of the server creates an EmulationServer-Socket instance and it is waiting on a appointed port.
2. To create a server connection, the client should create a socket, passing as constructor parameters of the server hostname and the number of the port. The emulation socket verifies the constructor parameters to determine the target node and understand from the network configuration which are the bandwidth and delay that must be associated with the communication between the client to the server.
3. The EmulationServer-Socket determines the node identity to request the connection (using the hostname and the number of the port for the client socket), and returns to the server process an emulation socket instance which shapes the traffic passing from the server to the client.
4. From this moment, the both programs can exchange information by invoking standard methods of write() and read().

Performance

Now a partial evaluation of the emulator is present. The experiments were made on Athlon 1600+PC running JDK1.4 over Linux. The tests were performed with an improved resolution of time, obtained by re-building the Linux kernel in order to set the operating system clock resolution to 1 ms (default time is 10 ms).

There is tested the emulation accuracy over a range of different bandwidth and delay values. The scalability of system has also been investigated by repeating the experiments with a node number increasing. The tests concerning bandwidth and delay were executed independently. While testing bandwidth, delay was set to 0, and while testing delay, the bandwidth was set to very high values.

To test the emulation accuracy of delay, there was been developed a simple client-server application, where one client sends 1-byte messages repeatedly to the server. The emulator was configured to introduce a communication delay going from the client to the server, and both of them were measured the elapsed time between sending the message and receiving it. To stress the system scalability, there was repeated a similar test with 20 clients and 1 server. Fig. 4 shows the measured delay in comparison with the configured delay for values going from 0 to 500 ms. The tests for bandwidth were carried out accuracy by making the client communicate with large amounts of bytes to the server. The emulator was configured to introduce a limitation of the bandwidth in the client to the server communication, and both of them in the process of communication measured the observed bandwidth. In this case also the test was repeated with 20 clients

and 1 server to evaluate the scalability of system. Fig. 5 shows the observed bandwidth in comparison with the configured one for values increasing from 64KB/s to 512KB/s.

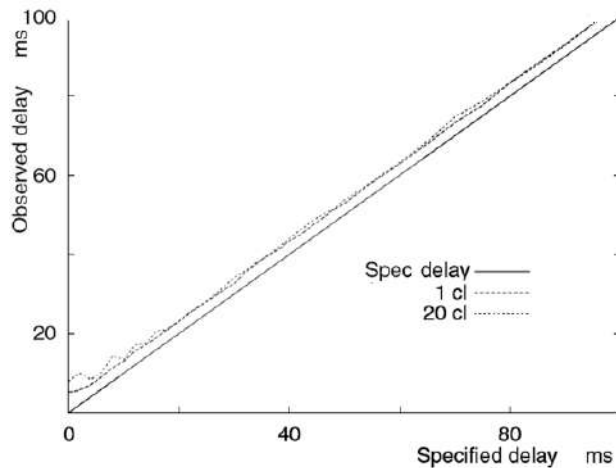


Fig. 4 – The accuracy of delay emulation

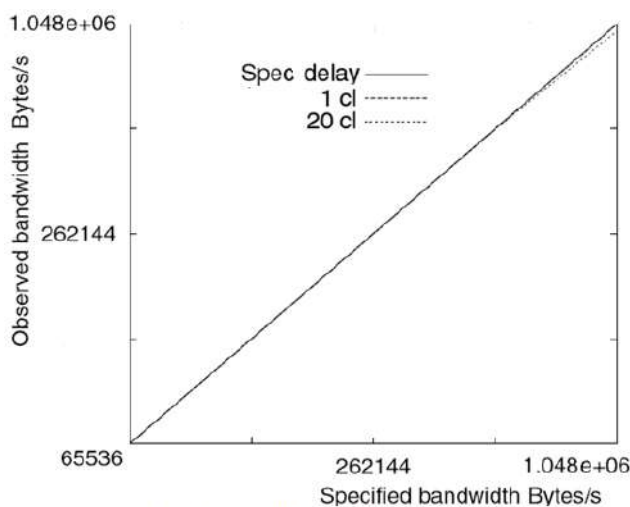


Fig. 5 – Bandwidth emulation accuracy

great customizability which are unavailable with other tools. When limitations on time resolution are not critical when emulating the Internet setting, the very small values of delay are unrealistic.

Discussion

Emulation sockets exhibit a good behavior for both bandwidth and emulation of delay. The observed values of bandwidth are really near the configured values, and even when 20 clients run together the system scales good. The figure of bandwidth starts scattering from the ideal value to values closing to 1MB/s. That is an unrealistic value if it is referred to Internet applications. In agreement with the delay, the observed values are slightly higher than the configured value. For almost of the values this difference is nearly constant, so that it can be removed by subtracting the measured value from the configured one. When the configured delay approximates to zero, system performances start to decrease because of the overhead introduced by emulation sockets. In this case the performances of the system remain good, as with Internet applications delay values near zero are not significant.

Performances of Java-based programs are lower than compiled languages, because of interpretation overhead. For this reason, it can be expected that scalability, overhead and time resolution achievable with emulation sockets will be worse than those achievable. However, Java offers byte code portability across different operating systems and hardware platforms. Emulation sockets are Java-implemented and they can be used as an emulation facility without being constrained to use a particular operation system or hardware.

On the other hand it is considered, that our solution provides a good tradeoff between functionality and performance. Emulation sockets are easy to use and allow

References

- [1] The ns simulator. <http://www.isi.edu/nsnam/ns>.
- [2] K. Fall. Network emulation in the vint/ns simulator. In *Proceedings of IEEE International Symposium on Computers and Communications*, 1999.
- [3] L. Rizzo. Dummynet: a simple approach to the evaluation of network protocols. *Computer Communication Review*, Jan. 27, 1997.
- [4] D.B. Ingham and G.D. Parrington. Delayline: a wide-area network emulation tool. *USENIX Computing Systems*, 7 (3): 313 – 332, 1994.
- [5] N. Davies, G.S. Blair, K. Cheverst, and A. Friday. A network emulator to support the development of adaptive applications. In *Proceedings of 2-nd USENIX Symposium on Mobile and Location Independent Computing*, A. Arbor, U.S.A, 1995.
- [6] Sun Microsystems. *Creating a Custom RMI Socket Factory*. <http://java.sun.com/j2se/1.3/docs/guide/rmi/rmisocketfactory.doc.html>.
- [7] W. Grosso. *Java RMI*. O'Reilly, 2001.

УДК 004.415.3

Пех П.А., Кузава О.В., Мацібора А.С., Шафета Ю.О.

Луцький національний технічний університет

ПРОГРАМНИЙ КОМПЛЕКС ДЛЯ ДОСЛІДЖЕННЯ ВХІДНИХ ПОТОКІВ СИСТЕМ МАСОВОГО ОБСЛУГОВУВАННЯ (СМО)

Пех П. А., Кузава О. В., Мацібора А. С., Шафета Ю. О. Програмний комплекс для дослідження вхідних потоків систем масового обслуговування (СМО). В статті запропоновано програмний комплекс засобами C++Builder для визначення на базі експериментальних даних параметрів інтенсивності та стабільності вхідного потоку СМО. Перевіряється також гіпотеза про Пуассонівський чи Ерлангівський характер розподілу вхідного потоку за χ^2 -критерієм.

Ключові слова: C++Builder проект, вхідний потік, розподіл Пуассона, розподіл Ерланга.

Пех П. А., Кузава О. В., Мацібора А. В., Шафета Ю. А. Програмный комплекс для исследования входящих потоков систем массового обслуживания (СМО). В статье предложено программный комплекс для определения на базе экспериментальных данных параметров интенсивности и стабильности входящего потока СМО. Проверяется также гипотеза об Пуассоновском или Эрланговском характере распределения входящего потока по χ^2 -критерию.

Ключевые слова: C++Builder проект, входящий поток, распределение Пуассона, распределение Эрланга.

Pekh P., Kuzava O., Macibora A., Shafeta J. Software system for research input streams queuing systems (QS). The paper proposed software system means C++Builder to determine the parameters of intensity and stability of the input stream QS based on experimental data. Also checked the hypothesis of Poisson or Erlang the distribution of exponential or Erlang the distribution of by χ^2 -test.

Keywords: C++Builder project, the input stream, the Poisson distribution, Erlang distribution

Постановка задачі. Розглянемо випадковий процес надходження вимог до СМО за інтервали часу фіксованої величини τ з метою дослідження його природи та параметрів. Кількість вимог, що надходять до СМО на протязі певного інтервалу часу τ , будемо вважати випадковою величиною R . Імовірність того, що за час τ до СМО надійде точно $R=r$ вимог, позначимо $P_r(\tau)$.

Потік вимог, для якого властиві ознаки стаціонарності, ординарності і відсутності післядії, прийнято називати найпростішим Π [1,2]. Для такого потоку імовірність $P_r(\tau)$ надходження r вимог протягом інтервалу часу τ описується розподілом Пуассона:

$$P(R=r) = P_r(\tau) = \frac{(\lambda\tau)^r}{r!} \cdot e^{-\lambda\tau}, \quad r = 0, 1, 2, \dots$$

Імовірність надходження r вимог протягом інтервалу часу τ в процесі Ерланга E [1,2] з післядією дорівнює імовірності надходження вимог у кількості не меншій, ніж $k-1$ штук у найпростішому процесі Π :

$$P_r^k(\tau) = e^{-k\lambda\tau} \sum_{n=k-1}^{kr+k-1} \frac{(k\lambda\tau)^n}{n!}, \quad r = 0, 1, 2, \dots$$

Задача, яка розв'язується у даній роботі, полягає в тому, щоб на базі експериментальних досліджень встановити вид розподілу вхідного потоку вимог і розрахувати його параметри. Інакше кажучи, необхідно, виходячи з аналізу експериментальних даних і встановлення кількісних параметрів інтенсивності та стабільності вхідного потоку, сформулювати гіпотезу про вид розподілу цього потоку, і за фактичним значенням χ^2 -критерію прийняти або відкинути її. Розроблення C++Builder проекту для вирішення цих завдань і становить предмет розгляду даної роботи.

Основна частина. Нехай за результатами експериментальних досліджень отримані $n=100$ значень кількості надходжень вимог до СМО протягом інтервалу часу $\lambda=5$ (табл.1). Звичайно, і обсяг вибірки n , і значення спостережених даних m_i можуть бути і іншими. Для вирішення всіх вище зазначених завдань стосовно такого роду емпіричних даних нами розроблено C++Builder проект. Він складається з головної та восьми підлеглих форм (табл.2), кожна з яких вирішує ту чи іншу частину задачі. Кількість форм за потреби може бути збільшена.

Таблиця 1. Результати спостережень кількості надходжень вимог m_i до СМО ($\lambda=5$).

i	m_i	i	m_i	i	m_i	i	m_i	i	m_i
1	3	21	5	41	7	61	0	81	8
2	4	22	4	42	2	62	6	82	4
3	8	23	3	43	5	63	7	83	8
4	1	24	6	44	4	64	4	84	2
5	4	25	6	45	7	65	9	85	9
6	3	26	2	46	3	66	6	86	0
7	5	27	4	47	7	67	2	87	4
8	2	28	3	48	5	68	14	88	7
9	4	29	7	49	6	69	5	89	3
10	5	30	5	50	5	70	5	90	7
11	5	31	7	51	1	71	6	91	5
12	3	32	3	52	6	72	6	92	1
13	8	33	4	53	4	73	3	93	6
14	4	34	8	54	3	74	9	94	4
15	2	35	6	55	7	75	5	95	5
16	6	36	9	56	5	76	8	96	6
17	5	37	4	57	4	77	6	97	5
18	1	38	8	58	2	78	2	98	3
19	4	39	6	59	6	79	5	99	7
20	3	40	5	60	10	80	3	100	4

Таблиця 2. Найменування та призначення форм і відповідних їм файлів

№ з/п	Найменування форми	Найменування файла	Призначення форми
1	Form1	MainFormUnit1	Головна форма проекту
2	Form2	GrupDanychUnit2	Групування даних
3	Form3	TabApr1Unit3	Апроксимація вхідного потоку розподілом Пуассона
4	Form4	GrafApr1Unit4	Графічна інтерпретація 1-го випадку
5	Form5	TabApr3Unit5	Апроксимація вхідного потоку розподілом Ерланга при $k=3$
6	Form6	GrafApr3Unit6	Графічна інтерпретація 2-го випадку
7	Form7	TabApr5Unit7	Апроксимація вхідного потоку розподілом Ерланга при $k=3$
8	Form8	GrafApr5Unit8	Графічна інтерпретація 3-го випадку
9	Form9	ContentUnit9	Зміст задач проекту

Вигляд головної форми показаний на рис.2. Більшість компонент цієї форми – це командні кнопки Button, кожна з яких забезпечує перехід на відповідну підлеглу форму, а з кожної підлеглої форми можна повернутись лише на головну.



Рис.1 – Вигляд головної форми проекту

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

```
#include "MainFormUnit1.h"  
#include "GrupDanychUnit2.h"  
#include "TabApr1Unit3.h"  
#include "GrafApr1Unit4.h"  
#include "TabApr3Unit5.h"  
#include "GrafApr3Unit6.h"  
#include "TabApr5Unit7.h"  
#include "GrafApr5Unit8.h"  
#include "ContentUnit9.h"
```

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

```
#include "GrupDanychUnit2.h"  
#include "MainFormUnit1.h"
```

З головною формою ми зв'язали також глобальний клас `data`

```
class DATA  
{public:  
    int n;  
    float a[250];  
    float s[20];  
};  
extern DATA data;
```

який містить масив `a[250]` для зберігання спостережених даних та масив `s[20]` для формування ряду розподілу `s[20]`. У такий спосіб ми досягли того, що масив вхідних даних буде досяжний в кожній підлеглий формі.

Розглянемо більш детально форму `Form3`, з допомогою якої здійснюється апроксимація експериментальних даних розподілом Пуассона (рис.2). Обробник подій кнопки `Button1` (Виконати розрахунки) форми `Form3` розпочинається описом змінних та масивів,

Form3 - Апроксимація вхідного потоку найпростішим потоком

Кількість інтервалів: 15

Виконати розрахунки

Очистити таблиці

До головного меню

	$r[i]$	$n[i]$	$r[i]*n[i]$	$r[i]-rc$	$(r[i]-rc)^2$	$n[i]/n$	$P(r[i])$	$n*P(r[i])$
0 стрічка	0,0000	2,0000	0,0000	-4,9300	24,3049	0,0200	0,0072	0,7227
1 стрічка	1,0000	4,0000	4,0000	-3,9300	15,4449	0,0400	0,0356	3,5627
2 стрічка	2,0000	8,0000	16,0000	-2,9300	8,5849	0,0800	0,0878	8,7820
3 стрічка	3,0000	13,0000	39,0000	-1,9300	3,7249	0,1300	0,1443	14,4317
4 стрічка	4,0000	17,0000	68,0000	-0,9300	0,8649	0,1700	0,1779	17,7871
5 стрічка	5,0000	18,0000	90,0000	0,0700	0,0049	0,1800	0,1754	17,5381
6 стрічка	6,0000	15,0000	90,0000	1,0700	1,1449	0,1500	0,1441	14,4104
7 стрічка	7,0000	10,0000	70,0000	2,0700	4,2849	0,1000	0,1015	10,1491
8 стрічка	8,0000	7,0000	56,0000	3,0700	9,4249	0,0700	0,0625	6,2544
9 стрічка	9,0000	4,0000	36,0000	4,0700	16,5649	0,0400	0,0343	3,4260
10 стрічка	10,0000	1,0000	10,0000	5,0700	25,7049	0,0100	0,0169	1,6890
11 стрічка	11,0000	0,0000	0,0000	6,0700	36,8449	0,0000	0,0076	0,7570
12 стрічка	12,0000	0,0000	0,0000	7,0700	49,9849	0,0000	0,0031	0,3110
13 стрічка	13,0000	0,0000	0,0000	8,0700	65,1249	0,0000	0,0012	0,1179
14 стрічка	14,0000	1,0000	14,0000	9,0700	82,2649	0,0100	0,0004	0,0415

Рис.2 – Вигляд форми проекту, на якій подається розрахункова таблиця для перевірки гіпотези про Пуассонівський характер вхідного потоку (з параметром стабільності $k=1$)

формуванням ряду розподілу, конструюванням таблиці `StringGrid1`, вибраної нами для виведення всієї розрахункової таблиці:

```
void __fastcall TForm3::Button1Click(TObject *Sender)
{int nn; //Кількість інтервалів групування даних
int i; //Робоча змінна
int j; //Робоча змінна
int k; //Робоча змінна
int l; //Робоча змінна
int kr; //Кількість рядків таблиці
float t[20][10]; //Масив t[20][10] для зберігання значень
// оброблення ряду розподілу
float s0,s1,s2,s3,s4,s5,s6,s7,s8; //Суми по кожному стовпчику
// масиву t[20][10]
float d0,d1,d2,d,p,m,rc; //Допоміжні змінні
float amin,amax;
float h; //Крок таблиці
amin=data.a[0]; //Найбільший елемент масиву a[i] спостережених даних
amax=data.a[0]; //Найменший елемент масиву a[i] спостережених даних
for (i = 0; i< data.n; i++){
if (data.a[i]<amin)
amin=data.a[i];
if (data.a[i]>amax)
amax=data.a[i];
}
kr=amax-amin+1;
nn=kr;
Edit1->Text=IntToStr(kr);
StringGrid1->RowCount=nn+4; //Кількість рядків таблиці StringGrid1
h=1;
//Формування значень стовпчика r[i] таблиці StringGrid1
```

```

for (i = amin; i<kr; i++)
    Memo1->Lines->Strings[i]=FloatToStrF(i+amin, ffFixed, 8, 0);
//Обнулення комірок масиву s[i]
for (i = 0; i < 20; i++) data.s[i]=0;
//Формування значень масиву s[i]
for (i = 0; i < data.n; i++)
    for (j=0; j<kr; j++)
        if ((data.a[i]>=amin +j*h) && (data.a[i]<amin+(j+1)*h))
            data.s[j]=data.s[j]+1;
// Іменування стрічок таблиці StringGrid1
for (i=1; i<=nn+4; i++)
    StringGrid1->Cells[0][i] =IntToStr(i-1) + " стрічка";
StringGrid1->ColCount=11; //Кількість стовпців таблиці StringGrid1
// Іменування стовпців таблиці StringGrid1
StringGrid1->Cells[1][0] ="    r[i]";
StringGrid1->Cells[2][0] ="    n[i]";
StringGrid1->Cells[3][0] ="    r[i]*n[i]";
StringGrid1->Cells[4][0] ="    r[i]-rc";
StringGrid1->Cells[5][0] ="    (r[i]-rc)^2";
StringGrid1->Cells[6][0] ="    n[i]/n";
StringGrid1->Cells[7][0] ="    P(r[i])";
StringGrid1->Cells[8][0] ="    n*P(r[i])";
StringGrid1->Cells[9][0] ="((n[i]-n*P(r[i]))^2)/(n*P(r[i]))";
//Зчитування даних з масиву s[i] у компонент Memo2
k=0;
for (i = 0; i< data.n; i++){
    Memo2->Lines->Strings[k]=FloatToStrF(data.s[i], ffFixed, 8, 0);
    k++;
}

```

Далі формуються або розраховуються дев'ять стовпчиків таблиці StringGrid1:

1.Стовпчик $r[i]$, у якому записуються всі можливі значення кількості вимог:

```

//Зчитування даних з компонента Memo1 у стовпчик r[i]
//та їх запис у масив a[i][j] та таблицю StringGrid1
k=0;
s0=0;
for (i = 0; i<nn; i++){
    j=0;
    t[i][j]=StrToFloat(Memo1->Lines->Strings[k]);
    s0=s0+t[i][j];
    StringGrid1->Cells[j+1][i+1]=FloatToStrF(t[i][j], ffFixed, 8, 2);
    k++;
}
t[nn][0]=s0;

```

2. Стовпчик $n[i]$, у якому записуються частоти, з якими зустрічалися значення $r[i]$:

```

//Зчитування даних з компонента Memo2
//у стовпчик емпіричних частот n[i]
//та їх запис у масив a[i][j] та таблицю StringGrid1
s1=0;
for (i = 0; i<nn; i++) {
    j=1;
    t[i][j]=data.s[i];
    s1=s1+t[i][j];
    StringGrid1->Cells[j+1][i+1]=FloatToStrF(t[i][j], ffFixed, 8, 2);
}

```

```
t[nn][1]=s1;  
t[nn+1][0]=s0/ data.n;  
t[nn+1][1]=s1/ data.n;
```

3. Столпчик $r[i]*n[i]$, у якому обчислюються добутки значень елементів двох попередніх столпчиків:

```
//Обчислення значень столпчика  $r[i]*n[i]$   
s2=0;  
  
for (i = 0; i < nn; i++) {  
    j=2;  
    t[i][j]= t[i][0]*t[i][1];  
    s2=s2+t[i][0]*t[i][1];  
}  
t[nn][2]=s2;  
t[nn+1][2]=s2/data.n;  
rc=t[nn+1][2];
```

4. Столпчик $r[i]-rc$, у якому обчислюються різниці значень столпчика $r[i]$ та середнього значення кількості вимог rc , причому rc розраховується за даними попереднього столпчика :

```
//Обчислення значень столпчика  $r[i]-rc$   
s3=0;  
for (i = 0; i < nn; i++) {  
    j=3;  
    t[i][j]= t[i][0]-rc;  
    s3=s3+t[i][0]-rc;  
}  
t[nn][3]=s3;  
t[nn+1][3]=s3/ data.n;
```

5. Столпчик $(r[i]-rc)^2$:

```
//Обчислення значень столпчика  $(r[i]-rc)^2$   
s4=0;  
for (i = 0; i < nn; i++) {  
    j=4;  
    t[i][j]= t[i][3]*t[i][3];  
    s4=s4+t[i][3]*t[i][3];  
}  
t[nn][4]=s4;  
t[nn+1][4]=s4/data.n;
```

6. Столпчик $n[i]/nn$:

```
//Обчислення значень столпчика  $n[i]/nn$   
s5=0;  
for (i = 0; i < nn; i++) {  
    j=5;  
    t[i][j]= t[i][1]/ data.n;  
    s5=s5+t[i][1]/ data.n;  
}  
t[nn][5]=s5;  
t[nn+1][5]=s5/ data.n;
```

7. Столпчик $P(r[i])$, у якому обчислюються значення імовірностей випадкової величини:

```
//Обчислення значень столпчика імовірностей  $P(r[i])$   
//за законом Пуассона  
s6=0;  
t[0][6]=exp(-rc);
```



```
d0=t[0][6];
d1=1;
d2=1;
for (i = 1; i < nn; i++) {
    j=6;
    d1=d1*rc;
    d2=d2*i;
    d=(d1/d2)*d0;
    t[i][j]=d;
    s6=s6+d;
}
t[nn][6]=s6;
t[nn+1][6]=s6/data.n;
```

8. Столпчик $n \cdot P(r[i])$, у якому обчислюються значення теоретичних частот:
//Обчислення значень столпчика теоретичних частот $n \cdot P(r[i])$

```
s7=0;
for (i = 0; i < nn; i++) {
    j=7;
    t[i][j]= data.n*t[i][6];
    s7=s7+data.n*t[i][6];
}
t[nn][7]=s7;
t[nn+1][7]=s7/ data.n;
```

9. Столпчик $(n[i]-n \cdot P(r[i]))^2 / n \cdot P(r[i])$, у якому обчислюються значення квадратів відхилень емпіричних та теоретичних частот, віднесених до теоретичних частот :

```
//Обчислення значень столпчика відхилень емпіричних частот від
//теоретичних
s8=0;
for (i = 0; i < nn; i++) {
    j=8;
    t[i][j]= (t[i][1]-t[i][7])*(t[i][1]-t[i][7])/t[i][7];
    s8=s8+(t[i][1]-t[i][7])*(t[i][1]-t[i][7])/t[i][7];
}
t[nn][8]=s8;
t[nn+1][8]=s8/ data.n;
i=nn+2;
for (j = 0; j<=7; j++){
    t[nn+2][j] =0;
}
```

Завершується програмний код виведенням на екран всіх вхідних та розрахованих величин. Значення суми останнього столпчика, а це є фактичне значення χ^2 -критерію, порівнюється з гранично допустимим і приймається або відкидається гіпотеза про те, що вхідний потік вимог описується розподілом Пуассона зі знайденими параметрами.

```
//Зчитування даних з масиву t[i][j] у компонент Memo2
for (i = 0; i < nn; i++){
    Memo2->Lines->Strings[i]=FloatToStrF(t[i][5],ffFixed,8,2);
}
//Зчитування даних з масиву t[i][j] у компонент Memo3
for (i = 0; i < nn; i++){
    Memo3->Lines->Strings[i]=FloatToStrF(t[i][6],ffFixed,8,2);
}
//Виведення на екран значень таблиці StringGrid1
for (i = 0; i<=nn+3; i++)
```



```
for (j = 0; j<=8; j++)  
    StringGrid1->Cells[j+1][i+1]=FloatToStrF(t[i][j], ffFixed, 8, 4);  
ShowMessage("Розв'язок задачі завершено.");  
}
```

У проєкті передбачено побудову багатокутників розподілу емпіричних частот та теоретичних імовірностей кількості вимог, що надходять до СМО протягом інтервалів часу фіксованої величини (рис.3). Решта форм проєкту сконструйовані аналогічно.

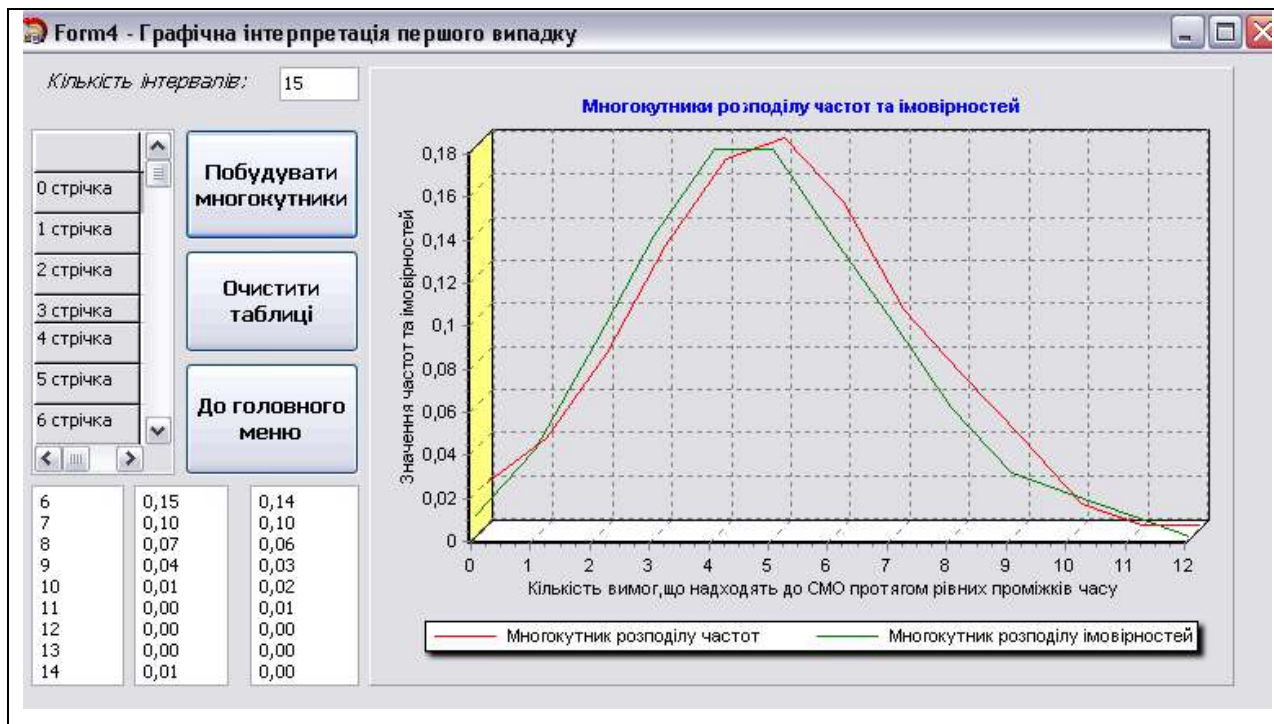


Рис.3 – Вигляд форми проєкту, на якій будуються багатокутники розподілу частот та імовірностей кількості вимог у випадку Пуассонівського вхідного потоку з параметром стабільності $k=1$

Висновки

В роботі розроблено С++Builder проєкт, за допомогою якого засобами сучасних інформаційних технологій розв'язується актуальна задача дослідження вхідних потоків СМО. Ця задача полягає у тому, щоб, виходячи з аналізу експериментальних даних і встановлення кількісних параметрів інтенсивності та стабільності вхідного потоку, сформулювати гіпотезу про вид розподілу, і за фактичним значенням χ^2 -критерію прийняти або відкинути її.

1. Д.Л. Дудюк та інші. Моделювання та оптимізація об'єктів і систем керування: Навчальний посібник для студентів вищих навчальних закладів. –К.:ІЗМН, 1998.-248 с.
2. Дудюк Дмитро та інші. Елементи теорії автоматичних ліній Навчальний посібник для студентів вищих навчальних закладів. –Київ-Львів:ІЗМН, 1998.-192 с.

УДК 004.415.3

Пех П.А., Серета А.О., Калінін Б.Ю., Лавренюк О.С.

Луцький національний технічний університет

ПРОГРАМНИЙ КОМПЛЕКС ДЛЯ ДОСЛІДЖЕННЯ ПРОЦЕСУ ОБРОБЛЕННЯ ВИМОГ У СИСТЕМАХ МАСОВОГО ОБСЛУГОВУВАННЯ (СМО)

Пех П. А., Серета А. О., Калінін Б. Ю., Лавренюк О. С. Програмний комплекс для дослідження процесу оброблення вимог у системах масового обслуговування (СМО). В статті запропоновано програмний комплекс засобами C++Builder для визначення на базі експериментальних даних параметрів інтенсивності та стабільності процесу оброблення вимог в СМО.

Перевіряється також гіпотеза про експоненціальний чи Ерлангівський характер розподілу часу оброблення вимог за χ^2 -критерієм.

Ключові слова: C++Builder проект, процес оброблення вимог, експоненціальний розподіл, розподіл Ерланга.

Пех П. А., Серета А. О., Калінін Б. Ю., Лавренюк А. С. Програмный комплекс для исследования процесса обработки требований в системах массового обслуживания (СМО). В статье предложено программный комплекс для определения на базе экспериментальных данных параметров интенсивности и стабильности процесса обработки требований в СМО. Проверяется также гипотеза об экспоненциальном или Эрланговском характере распределения времени обработки требований по χ^2 -критерию.

Ключевые слова: C++Builder проект, процесс обработки требований, экспоненциальное распределение, распределение Эрланга.

Pekh P., Seretda A., Kalinin B., Lavrenjuk A. Software system for research of processing requirements in queuing systems (QS). The paper proposed software system means C++Builder to determine the parameters of intensity and stability of processing requirements QS based on experimental data. Also checked the hypothesis of exponential or Erlang the distribution of time processing requirements in QS by χ^2 -test.

Keywords: C++Builder project, the processing requirements, the exponential distribution, Erlang distribution

Постановка задачі. Розглянемо випадковий процес оброблення вимог в СМО з метою дослідження його природи та параметрів. Тривалість часу оброблення вимог в СМО будемо вважати випадковою величиною T . Імовірність того, що випадкова величина T прийме значення, менше деякого довільного значення t , описується інтегральною функцією розподілу, тобто $F(t) = P(T < t)$.

Процес оброблення вимог, для якого властиві ознаки стаціонарності, ординарності і відсутності післядії, прийнято називати найпростішим [1,2]. Для такого процесу інтегральна функція розподілу часу оброблення вимог має експоненціальний вигляд :

$$F(t) = P(T < t) = 1 - e^{-\lambda t}.$$

Інтегральна функція розподілу імовірностей проміжків часу в процесі Ерланга Е [1,2] з інтенсивністю λ та параметром стабільності k має вигляд:

$$F^k(t) = 1 - e^{-k\lambda t} \sum_{n=0}^{k-1} \frac{(k\lambda t)^n}{n!}$$

Задача, яка розв'язується у даній роботі, полягає в тому, щоб на базі експериментальних досліджень встановити вид розподілу процесу оброблення вимог і розрахувати його параметри. Інакше кажучи, необхідно, виходячи з аналізу експериментальних даних і встановлення кількісних параметрів інтенсивності та стабільності процесу оброблення вимог, сформулювати гіпотезу про вид розподілу цього процесу, і за фактичним значенням χ^2 -критерію прийняти або відкинути її. Розроблення C++Builder проекту для вирішення цих завдань і становить предмет розгляду даної роботи.

Основна частина. Нехай за результатами експериментальних досліджень отримані $n=100$ значень часу оброблення вимог t_i в СМО (табл.1). Звичайно, і обсяг вибірки n , і значення спостережених даних t_i можуть бути і іншими. Для вирішення всіх вище зазначених завдань стосовно такого роду емпіричних даних нами розроблено C++Builder проект. Він складається з головної та восьми підлеглих форм (табл.2), кожна з яких вирішує ту чи іншу частину задачі. Кількість форм за потреби може бути збільшена.

Таблиця 1. Результати спостережень часу оброблення вимог t_i у СМО.

i	t_i	i	t_i	i	t_i	i	t_i	i	t_i
1	50	21	67	41	83	61	19	81	182
2	172	22	40	42	23	62	76	82	30
3	17	23	86	43	36	63	33	83	48
4	56	24	7	44	50	64	90	84	118
5	13	25	17	45	10	65	19	85	10
6	128	26	96	46	30	66	72	86	69
7	26	27	14	47	83	67	13	87	140
8	66	28	160	48	24	68	37	88	78
9	198	29	22	49	188	69	45	89	5
10	44	30	35	50	8	70	32	90	93
11	52	31	108	51	16	71	113	91	16
12	7	32	28	52	176	72	130	92	103
13	82	33	60	53	12	73	9	93	38
14	32	34	18	54	34	74	116	94	58
15	18	35	99	55	72	75	21	95	10
16	42	36	50	56	37	76	147	96	30
17	58	37	73	57	46	77	24	97	49
18	6	38	88	58	6	78	62	98	16
19	54	39	9	59	64	79	38	99	124
20	25	40	29	60	203	80	11	100	146

Таблиця 2. Найменування та призначення форм і відповідних їм файлів

№ з/п	Найменування форми	Найменування файла	Призначення форми
1	Form1	MainFormUnit1	Головна форма проекту
2	Form2	GrupDanUnit2	Групування даних
3	Form3	TabErl1Unit3	Наближення розподілу часу оброблення експоненціальним розподілом ($k=1$)
4	Form4	GrafApr1Unit4	Графік функції експоненціального розподілу
5	Form5	TabErl13Unit5	Наближення розподілу часу оброблення розподілом Ерланга ($k=3$)
6	Form6	GrafErl3Unit6	Графік функції розподілу Ерланга ($k=3$)
7	Form7	TabErl15Unit7	Наближення розподілу часу оброблення розподілом Ерланга ($k=5$)
8	Form8	GrafErl5Unit8	Графік функції розподілу Ерланга ($k=5$)
9	Form9	ContentUnit9	Зміст задач проекту

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

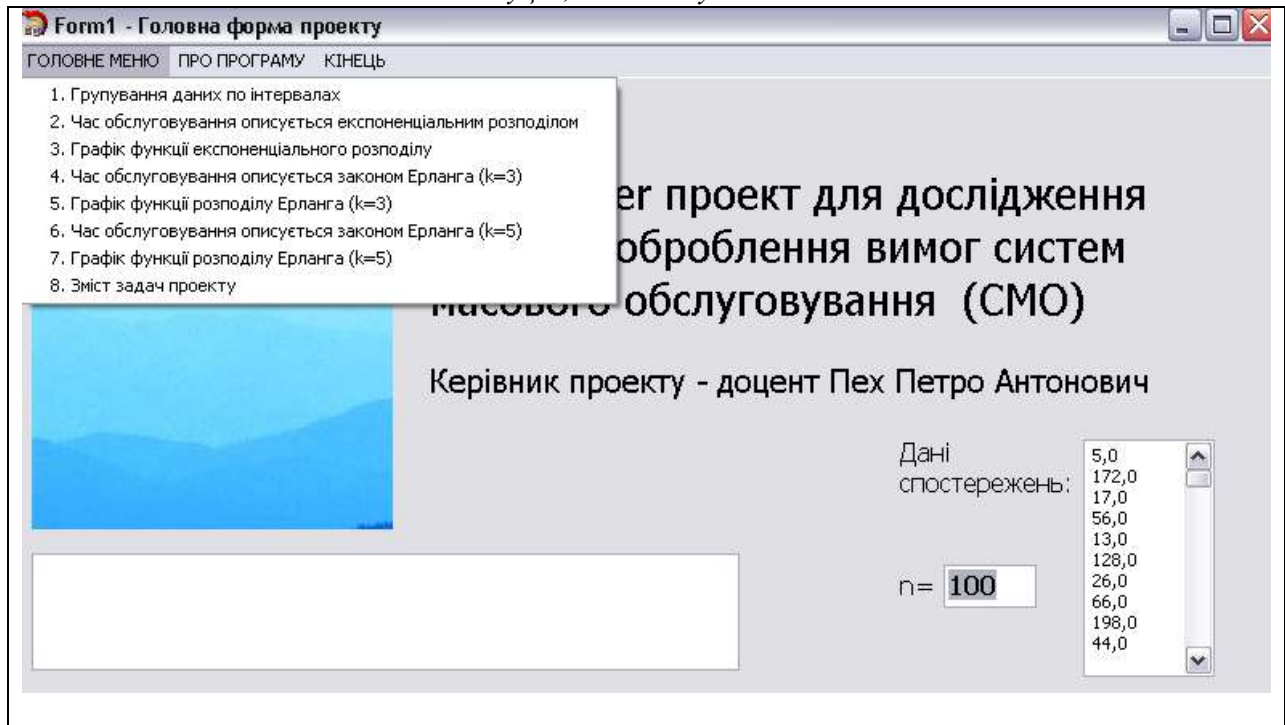


Рис.1 – Вигляд головної форми проекту

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

```
#include "MainFormUnit1.h"  
#include "GrupDanUnit2.h"  
#include "TabEr11Unit3.h"  
#include "GrafEr11Unit4.h"  
#include "TabEr13Unit5.h"  
#include "GrafEr13Unit6.h"  
#include "TabEr15Unit7.h"  
#include "GrafEr15Unit8.h"  
#include "ContentUnit9.h"  
#include "ContentUnit9.h"
```

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

```
#include "GrupDanUnit2.h"  
#include "MainFormUnit1.h"
```

З головною формою ми зв'язали також глобальний клас `data`

```
class DATA  
{public:  
    int n;  
    float a[250];  
    float s[20];  
};  
extern DATA data;
```

який містить масив `a[250]` для зберігання спостережених даних та масив `s[20]` для формування ряду розподілу `s[20]`. У такий спосіб ми досягли того, що масив вхідних даних буде досяжний в кожній підлеглий формі.

Розглянемо більш детально форму `Form3`, з допомогою якої здійснюється апроксимація експериментальних даних експоненціальним розподілом (рис.2). Обробник подій кнопки `Button1` (Виконати розрахунки) форми `Form3` розпочинається описом змінних та масивів,

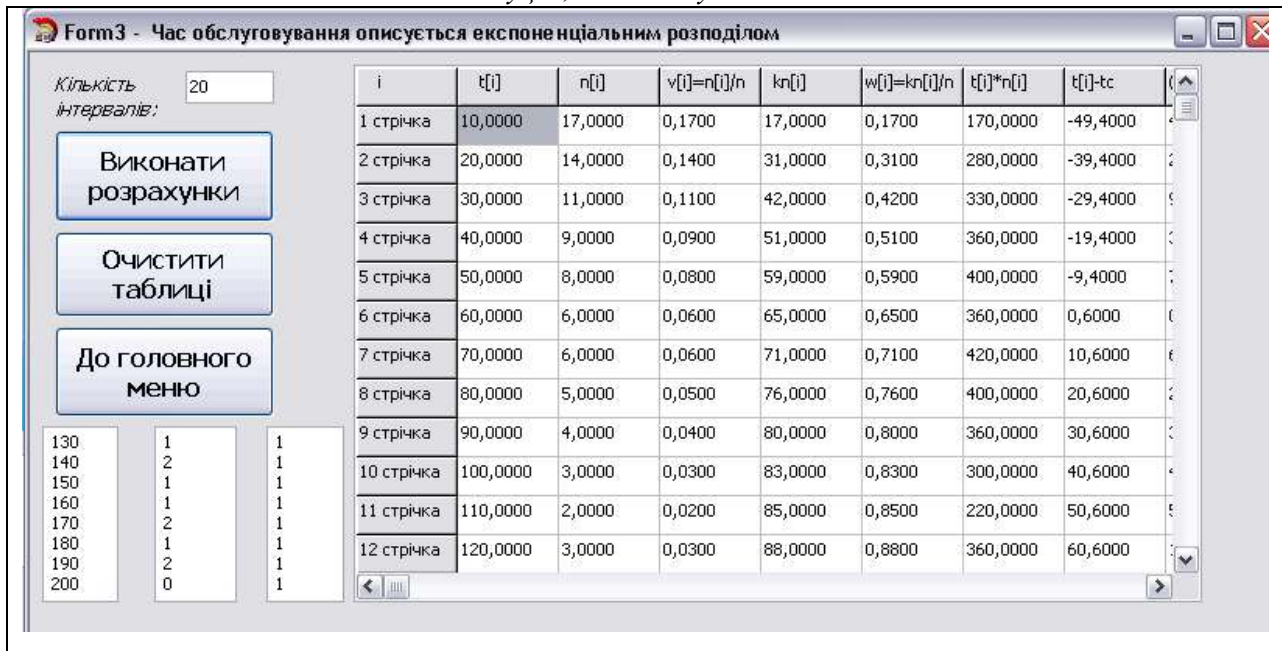


Рис.2 – Видгляд форми проекту, на якій подається розрахункова таблиця для перевірки гіпотези про експоненціальний (параметр стабільності $k=1$) розподіл часу оброблення вимог

формуванням ряду розподілу, конструюванням таблиці `StringGrid1`, вибраної нами для виведення всієї розрахункової таблиці:

```
void __fastcall TForm3::Button1Click(TObject *Sender)
{int nn; //Кількість інтервалів групування даних
int i; //Робоча змінна
int j; //Робоча змінна
int k; //Робоча змінна
int l; //Робоча змінна
int kr; //Кількість рядків таблиці
float t[30][30]; //Масив t[25][10] для зберігання значень
//оброблення ряду розподілу
float s0,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11; //Суми по кожному
//стовпчику масиву t[20][10]
float d0,d1,d2,d,p,m,tc,dc,kc; //Допоміжні змінні
float amin,amax;
float h; //Крок таблиці
//Таблиця StringGrid1 містить результати розрахунків параметрів
//процесу обслуговування
StringGrid1->RowCount=30; //Гранична кількість рядків таблиці
//StringGrid1
StringGrid1->ColCount=14; //Гранична кількість стовпців таблиці
//StringGrid1
amin=data.a[0]; //Найбільший елемент масиву a[i] спостережених
даних
amax=data.a[0]; //Найменший елемент масиву a[i] спостережених даних
for (i = 0; i < data.n; i++){
if (data.a[i]<amin)
amin=data.a[i];
if (data.a[i]>amax)
amax=data.a[i];
}
h=5;
kr=amax/(2*h); //Кількість інтервалів групування даних
```

```
nn=kr;
Edit1->Text=IntToStr(kr);
//Обнулення значень таблиці StringGrid1
for (i = 0; i<kr+4; i++)
    for (j = 0; j<14; j++)
        StringGrid1->Cells[j+1][i+1]="0,0";

//Таблиця StringGrid1 містить результати розрахунків параметрів
//вхідного потоку
StringGrid1->RowCount=kr+4; //Фактична кількість рядків таблиці
//StringGrid1
// Іменування стрічок таблиці StringGrid1
for (i=1; i<=kr+3; i++)
    StringGrid1->Cells[0][i] =IntToStr(i)+" стрічка";
//Фактична кількість стовпців таблиці StringGrid1
StringGrid1->ColCount=13;
// Іменування стовпців таблиці StringGrid1
StringGrid1->Cells[0][0] ="    i";
StringGrid1->Cells[1][0] ="    t[i]";
StringGrid1->Cells[2][0] ="    n[i]";
StringGrid1->Cells[3][0] =" v[i]=n[i]/n";
StringGrid1->Cells[4][0] ="    kn[i]";
StringGrid1->Cells[5][0] ="w[i]=kn[i]/n";
StringGrid1->Cells[6][0] =" t[i]*n[i]";
StringGrid1->Cells[7][0] =" t[i]-tc";
StringGrid1->Cells[8][0] ="((t[i]-tc)^2)/v[i]";
StringGrid1->Cells[9][0] =" F(t[i])";
StringGrid1->Cells[10][0] =" n*F(t[i])";
StringGrid1->Cells[11][0] =" nт[i]";
StringGrid1->Cells[12][0] ="((n[i]-nт[i])^2)/(nт[i])";
for (i = 0; i<30; i++)
    for (j = 0; j<30; j++)
        t[i][j]=0.0;
//Формування значень стовпчика t[i] таблиці StringGrid1
for (i = 0; i<kr; i++){
    Memo1->Lines-
>Strings[i]=FloatToStrF((amin+h)*(i+1),ffFixed,4,0);
}
//Обнулення комірок масиву s[i]
for (i = 0; i < 30; i++) data.s[i]=0;
//Формування значень масиву s[i]
for (i = 0; i < data.n; i++)
    for (j=1; j<=kr; j++)
        if ((data.a[i]>=amin+(j-1)*2*h) && (data.a[i]<amin+j*2*h))
            data.s[j-1]=data.s[j-1]+1;

//Занесення значень масиву s[i] в стовпчик n[i] таблиці StringGrid1
for (j=1; j<=kr; j++){
    StringGrid1->Cells[2][j]=FloatToStrF(data.s[j],ffFixed,8,0);
    Memo2->Lines->Strings[j]=FloatToStrF(data.s[j],ffFixed,8,0);
}
```

Далі формуються або розраховуються всі стовпчики таблиці StringGrid1:

1. Стовпчик $t[i]$, у якому записуються середини інтервалів можливих значень часу оброблення вимог:

```
//Зчитування даних з компонента Memo1 у стовпчик t[i]
```

```
//та їх запис у масив t[i][j] та таблицю StringGrid1
s0=0;
for (i = 0; i<nn; i++){
    j=0;
    t[i][j]=(amin+h)*(i+1);
    s0=s0+t[i][j];
    StringGrid1->Cells[j+1][i+1]=FloatToStrF(t[i][j],ffFixed,6,0);
};
t[nn][0]=s0;
t[nn+1][0]=s0/data.n;
```

2. Столпчик $n[i]$, у якому записуються частоти, з якими зустрічалися значення $t[i]$:

```
//Зчитування даних з компонента Memo2 у столпчик
// емпіричних частот n[i]
//та їх запис у масив t[i][j] та таблицю StringGrid1
s1=0;
for (i = 0; i<nn; i++) {
    j=1;
    t[i][j]=data.s[i];
    s1=s1+t[i][j];
    StringGrid1->Cells[j+1][i+1]=FloatToStrF(t[i][j],ffFixed,8,2);
};
t[nn][1]=s1;
t[nn+1][1]=s1/data.n;
```

3. Столпчик $v[i]=n[i]/n$, у якому записуються частоти, з якими зустрічалися значення $t[i]$:

//Обчислення значень столпчика частостей $v[i]=n[i]/n$

```
s2=0;
for (i = 0; i < nn; i++) {
    j=2;
    t[i][j]= t[i][1]/data.n;
    s2=s2+t[i][1]/data.n;
}
t[nn][2]=s2;
t[nn+1][2]=s2/data.n;
```

4. Столпчик $kn[i]$, у якому записуються кумулятивні частоти, з якими зустрічалися значення $t[i]$:

//Обчислення значень столпчика $kn[i]$ $v[i]=n[i]/n$ $t[i]*n[i]$

```
t[0][3]=t[0][1];
s3=t[0][1];
for (i = 1; i < nn; i++) {
    j=3;
    t[i][3]= s3+t[i][1];
    s3=t[i][3];
}
t[nn][3]=s3;
t[nn+1][3]=s3/data.n;
```

5. Столпчик $w[i]=kn[i]/n$, у якому записуються кумулятивні частоти, з якими зустрічалися значення $t[i]$:

//Обчислення значень столпчика $w[i]=kn[i]/n$

```
s4=0;
for (i = 0; i<nn; i++) {
    j=4;
    t[i][4]=t[i][3]/data.n;
    s4=s4+t[i][3]/data.n;
}
```

```
StringGrid1->Cells[j+1][i+1]=FloatToStrF(t[i][j], ffFixed, 8, 2);  
};  
t[nn][4]=s1;  
t[nn+1][4]=s1/data.n;
```

6. Столпчик $t[i]*n[i]$, у якому розраховуються добутки елементів відповідних стовпчиків:
//Обчислення значень стовпчика $t[i]*n[i]$

```
s5=0;  
for (i = 0; i<nn; i++) {  
    j=5;  
    t[i][j]=t[i][0]*t[i][1];  
    s5=s5+t[i][0]*t[i][1];  
    StringGrid1->Cells[j+1][i+1]=FloatToStrF(t[i][j], ffFixed, 8, 2);  
};  
t[nn][5]=s5;  
t[nn+1][5]=s5/data.n;  
tc=t[nn+1][5];
```

7. Столпчик $t[i]-tc$, у якому розраховуються різниці елементів стовпчика $t[i]$ та середнього значення часу оброблення вимог tc , обчисленого у попередньому стовпчику:

```
//Обчислення значень стовпчика  $t[i]-tc$   
s6=0;  
for (i = 0; i<nn; i++) {  
    j=6;  
    t[i][j]=t[i][0]-tc;  
    s6=s6+t[i][0]-tc;  
    StringGrid1->Cells[j+1][i+1]=FloatToStrF(t[i][j], ffFixed, 8, 2);  
};  
t[nn][6]=s6;  
t[nn+1][6]=s6/data.n;
```

8. Столпчик $(t[i]-tc)^2/v[i]$, у якому розраховуються елементи за даними двох відповідних попередніх стовпчиків:

```
//Обчислення значень стовпчика  $(t[i]-tc)^2/v[i]$   
s7=0;  
for (i = 0; i<nn; i++) {  
    j=7;  
    t[i][j]=t[i][6]*t[i][6]*t[i][2];  
    s7=s7+t[i][6]*t[i][6]*t[i][2];  
    StringGrid1->Cells[j+1][i+1]=FloatToStrF(t[i][j], ffFixed, 8, 2);  
};  
t[nn][7]=s7;  
t[nn+1][7]=sqrt(s7);  
dc=t[nn][7];  
kc=tc*tc/dc;  
t[nn+2][7]=kc;
```

9. Столпчик $F(t[i])$, у якому розраховуються значення теоретичних імовірностей

```
//Обчислення значень стовпчика  $F(t[i])$   
s8=0;  
for (i = 0; i<nn; i++) {  
    j=8;  
    t[i][j]=1-exp(-t[i][0]/tc);  
    StringGrid1->Cells[j+1][i+1]=FloatToStrF(t[i][j], ffFixed, 8, 2);  
};
```

10. Столпчик $n*F(t[i])$, у якому розраховуються значення теоретичних частот


```
//Обчислення значень стовпчика n*F(t[i])
s9=0;
for (i = 0; i<=nn; i++) {
    j=9;
    t[i][j]=t[i][8]*data.n;
    StringGrid1->Cells[j+1][i+1]=FloatToStrF(t[i][j], ffFixed, 8, 2);
};
```

11. Стовпчик $nt[i]$, у якому розраховуються значення накопичених теоретичних частот

```
//Обчислення значень стовпчика nt[i]
t[0][10]=t[0][9];
s10=t[0][9];
for (i = 1; i<nn; i++) {
    j=10;
    t[i][j]=t[i][9]-t[i-1][9];
    s10=s10+ t[i][9]-t[i-1][9];
    StringGrid1->Cells[j+1][i+1]=FloatToStrF(t[i][j], ffFixed, 8, 2);
};
t[nn][10]=s10;
```

12. Стовпчик $((n[i]-nt[i])^2)/nt[i]$, у якому розраховуються значення відхилень

```
//Обчислення значень стовпчика ((n[i]-nt[i])^2)/nt[i]
s11=0;
for (i = 0; i<nn; i++) {
    j=11;
    t[i][j]=(t[i][1]-t[i][10])*(t[i][1]-t[i][10])/t[i][10];
    s11=s11+ (t[i][1]-t[i][10])*(t[i][1]-t[i][10])/t[i][10];;
    StringGrid1->Cells[j+1][i+1]=FloatToStrF(t[i][j], ffFixed, 8, 2);
};
t[nn][11]=s11;
```

Завершується програмний код виведенням на екран всіх вхідних та розрахованих величин. За значенням суми останнього стовпчика, а це є фактичне значення χ^2 -критерію, яке порівнюється з гранично допустимим і приймається або відкидається гіпотеза про те, що процес оброблення вимог описується експоненціальним розподілом.

У проєкті передбачено також побудову графіків емпіричної та теоретичної інтегральних функцій розподілу часу оброблення вимог у випадку $k=1$ (рис.3).

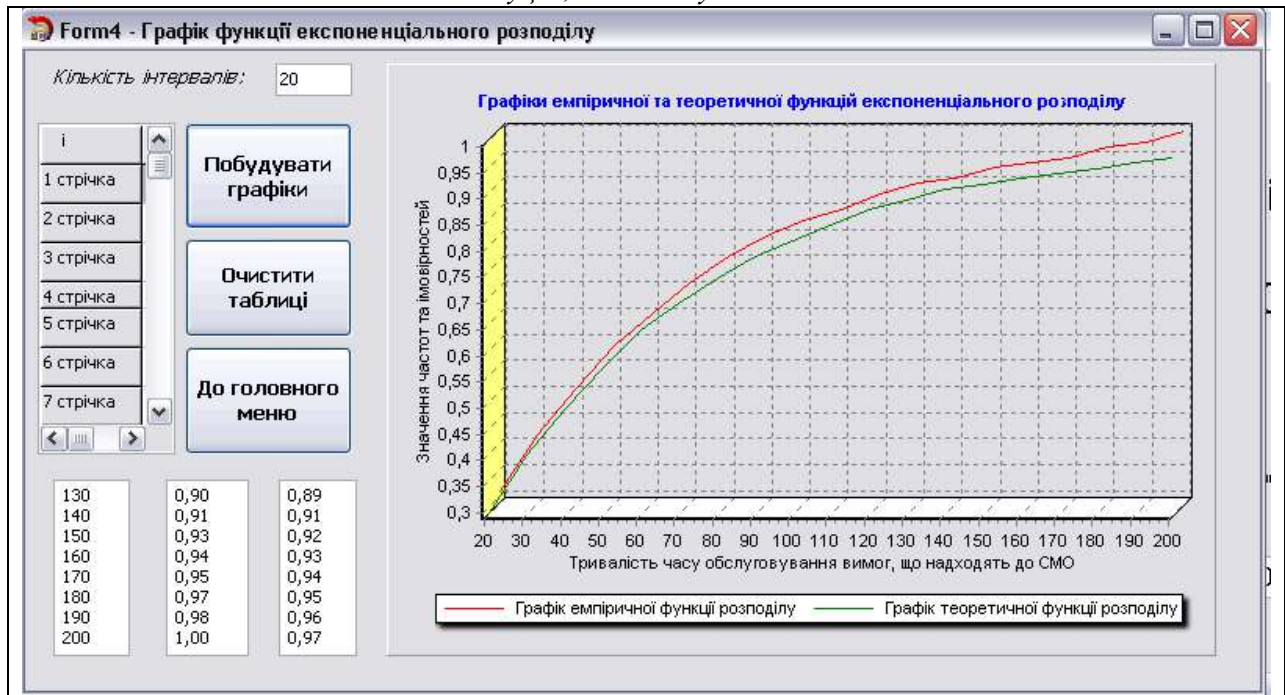


Рис.3 – Вигляд форми проекту, на якій будуються графіки емпіричної та теоретичної інтегральних функцій розподілу часу оброблення вимог у випадку $k=1$

Висновки

В роботі розроблено C++Builder проект, за допомогою якого засобами сучасних інформаційних технологій розв'язується актуальна задача дослідження процесу оброблення вимог у СМО. Ця задача полягає у тому, щоб, виходячи з аналізу експериментальних даних і встановлення кількісних параметрів інтенсивності та стабільності процесу оброблення вимог, сформулювати гіпотезу про вид розподілу, і за фактичним значенням χ^2 -критерію прийняти або відкинути її.

1. Д.Л. Дудюк та інші. Моделювання та оптимізація об'єктів і систем керування: Навчальний посібник для студентів вищих навчальних закладів. –К.:ІЗМН, 1998.-248 с.
2. Дудюк Дмитро та інші. Елементи теорії автоматичних ліній: Навчальний посібник для студентів вищих навчальних закладів. –Київ-Львів:ІЗМН, 1998.-192 с.

УДК 004.492.3

О.С. Савенко, С.М. Лисенко, А.Ф. Кришук
Хмельницький національний університет

ЕФЕКТИВНІСТЬ ДІАГНОСТУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ НА НАЯВНІСТЬ БОТ-МЕРЕЖ АНТИВІРУСНОЮ МУЛЬТИАГЕНТНОЮ СИСТЕМОЮ

Савенко О.С., Лисенко С.М., Кришук А.Ф. Ефективність діагностування комп'ютерних систем на наявність бот-мереж антивірусною мультиагентною системою. В статті розглянуто дослідження ефективності діагностування комп'ютерних систем на наявність бот-мереж використовуючи антивірусну мультиагентну систему, як засіб діагностування. Проведено порівняння розробленої антивірусної системи з існуючими. Визначено рівень ефективності діагностування комп'ютерних систем на наявність бот-мереж.

Ключові слова: агент, мультиагентна система, антивірусне діагностування, ефективність
Табл. 2, Рис. 2, Літ. 11

Савенко О.С., Лисенко С.М., Кришук А.Ф. Эффективность диагностирования компьютерных систем на наличие бот-сетей антивирусной мультиагентной системой. В статье рассмотрены исследования эффективности диагностирования компьютерных систем на наличие бот-сетей используя антивирусную мультиагентную систему, как средство диагностики. Проведено сравнение разработанной антивирусной системы с существующими. Определен уровень эффективности диагностирования компьютерных систем на наличие бот-сетей.

Ключевые слова: агент, мультиагентная система, антивирусное диагностирование, эффективность

Savenko O.S., Lysenko S.M., Kryshchuk A.F. The computer systems diagnosis efficiency for botnet presence by antivirus multi-agent systems. In article the computer systems diagnosis efficiency for botnet presence by antivirus multi-agent systems is investigated. A comparison of the developed system with existing antivirus systems is held. The level of efficiency of computer systems diagnosing for botnets presence is evaluated.

Keywords: agent, multiagent system, antivirus diagnosis, efficiency.

Вступ. Бот-мережі - один з головних і найпопулярніших інструментів сучасної кіберзлочинності. Комп'ютерні мережі, які складаються з великої кількості комп'ютерних систем, інфікованих програмами-ботами, автоматично виконують ті чи інші дії в інтересах власника або керуючого такою мережею [1-5].

Сучасні бот-мережі, виконують повний спектр зловмисних дій у корпоративних і приватних комп'ютерних систем (КС): від класичних збору адрес електронної пошти та подальшого розсилання «спаму» до розкрадання інформації для банківських рахунків та комерційного шпигунства. Бот-мережі найчастіше використовуються для DDoS-атак на Інтернет-ресурси, а також на засмічування соціальних мереж пропагандистськими, рекламними та небезпечними повідомленнями.

Аналіз останніх досліджень і публікацій. В [6] запропоновано метод визначення рівня присутності бот-мережі шляхом аналізу проявів дій ботів в ймовірно інфікованій КС. Залучення такого підходу виконується у випадку підозрілої поведінки програм на певній кількості КС корпоративної мережі. Антивірусне діагностування здійснювалося на базі побудованої антивірусної мультиагентної системи (МАС) [7, 8].

В роботі пропонувалося будувати карту-схему зв'язків корпоративної мережі, яка формувалася певними записами в кожному антивірусному агенті МАС. На базі даної інформації усі агенти здійснювали комунікативний обмін даними.

В процесі роботи на кожній комп'ютерній системі здійснювалося антивірусне діагностування присутніми в агенті сенсорами. Отримані результати антивірусного діагностування аналізувалися на предмет того, який із сенсорів спрацював та який рівень підозрілості він видав. Якщо спрацювали сенсори сигнатурного аналізатора S_1 та контрольних сум S_2 результати R_{S_1} та R_{S_2} інтерпретувалися як 100% виявлення шкідливого ПЗ.

Якщо спрацювали сенсори евристичного R_{S_3} та поведінкового R_{S_4} аналізаторів з подоланням певного порогу $n \leq \max(R_{S_3}, R_{S_4}) \leq 100$, то здійснювався аналіз рівня підозрілості і приймалося рішення щодо можливого блокування дій ПЗ, і подальшого його видалення.

Ключовим моментом дослідження була ситуація, коли результати антивірусного діагностування належать проміжку $m \leq \max(R_{S_3}, R_{S_4}) < n$. В цьому випадку антивірусний агент КС здійснював опитування решти агентів на предмет прояву схожої поведінки на інших КС корпоративної мережі. Якщо в результаті опитування антивірусний агент отримав інформацію від одного або більше агентів КС щодо схожої підозрілої поведінки програмного забезпечення, то здійснювалася перебудова наявної карти мережі з помітками ймовірно інфікованих КС. З такого

набору вибиралася одна КС і змінювався її тип підключення до мережі таким чином, щоб робота бота в нових умовах була неможливою.

Рівень присутності бот-мереж визначався після опрацювання проявів бот-мережі на «перепідключеній» КС, помічених комп'ютерних системах та інших КС корпоративної мережі.

Засобами нечіткої експертної системи здійснювалося визначення рівня присутності бот-мереж, яка спростовувала або підтверджувала факт присутності бот-мережі.

Визначення рівня присутності бот-мережі шляхом аналізу проявів дій ботів в ситуації навмисної зміни типу підключення ймовірно інфікованої КС функціонувало за алгоритмом 1 [9, 10].

Для усіх КС корпоративної мережі
поки $КС_i$ функціонує
якщо $R_{S_1} = true$ або $R_{S_2} = true$ тоді блокувати і видаляти ПЗ;
якщо $R_{S_3} = true$ або $R_{S_4} = true$ і $n \leq \max(R_{S_3}, R_{S_4}) \leq 100$ тоді блокувати і видаляти ПЗ;
інакше якщо $R_{S_3} = true$ або $R_{S_4} = true$ і $m \leq \max(R_{S_3}, R_{S_4}) < n$ тоді опитати інші агенти; вибрати КС для перепідключення; проаналізувати рівень присутності бот-мережі в корпоративній мережі;
інакше якщо $R < m$ очікувати результати $R_{S_1}, R_{S_2}, R_{S_3}, R_{S_4}$.

Алгоритм 1. Діагностування КС на наявність бот-мереж

Розроблена МАС містить також сенсора S_7 , який дозволяє здійснювати емуляцію запуску та виконання над потенційно шкідливим ПЗ певних дій. Реакції на вказані дії дозволяють зробити висновок про присутність в ньому поліморфного коду.

Виходячи з властивостей поліморфних вірусів на сенсор S_7 виконує:

- провокативні дії по відношенню до ймовірно інфікованого файлу;
- повторні запуски підозрілого файлу для ймовірної модифікації власного коду;
- виявлення шкідливого програмного забезпечення (ШПЗ) шляхом аналізу його поведінки та можливої зміни свого тіла, що базується на принципах відомих рівнів поліморфізму.

Під провокативними діями мається на увазі виявлення властивості поліморфних вірусів створення своєї копії зі зміною власного тіла при його видаленні. Дана властивість часто призводить до того, що оригінал може бути виявлено та видалено, а нова копія бота буде невидимою для антивірусу.

Здійснення повторних запусків підозрілого ПЗ може показати ймовірну «зміну» тіла програми в результаті виконання шифрування. Виявлення такої зміни можливе завдяки побудові «відбитків» К еталонного та модифікованого файлів K' та їх подальшого порівняння. «Відбиток» K формується визначеною двійковою послідовністю $K = \alpha, \beta, \chi, \delta, \varepsilon$, де α - назва файлу; β - розмір файлу; χ - дата останньої зміни; δ - системний атрибут; ε - 128 бітний код MD5.

Сенсор S_7 також виконує роль поведінкового аналізатора, який здійснює аналіз дій з урахуванням моделей поліморфних вірусів різних рівнів. На основі знань про поліморфну природу поведінки вірусів та поведінок бот-мереж є можливим їх виявлення шляхом динамічного порівняння відомих поведінок з поведінками нових бот-мереж. Виявлення поліморфного коду здійснюється з урахуванням відкидання можливих команд-сміття, перестановок команд, команд вибору шифрувальника, самих команд шифрувальника тощо. Відомі поведінки ботів та поведінки досліджуваних об'єктів представляються послідовностями, які в подальшому порівнюються.

Для порівняння шаблонних поведінок з тією, що виконує потенційно небезпечна програма, використано алгоритм приблизного порівняння, який розв'язує задачу k-приблизного збігу. Використаний алгоритм вимагає $O(kn)$ [11].

Виходячи з концепції функціонування антивірусної МАС, в кожному агенті відбувається очікування спрацювання евристичного S_3 та поведінкового S_4 сенсорів. У випадку їх спрацювання, а також виявлення ними факту розпаковування певного файлу, виконується завантаження підозрілого програмного об'єкта в сенсор-емулятор S_7 (див. рис.1).

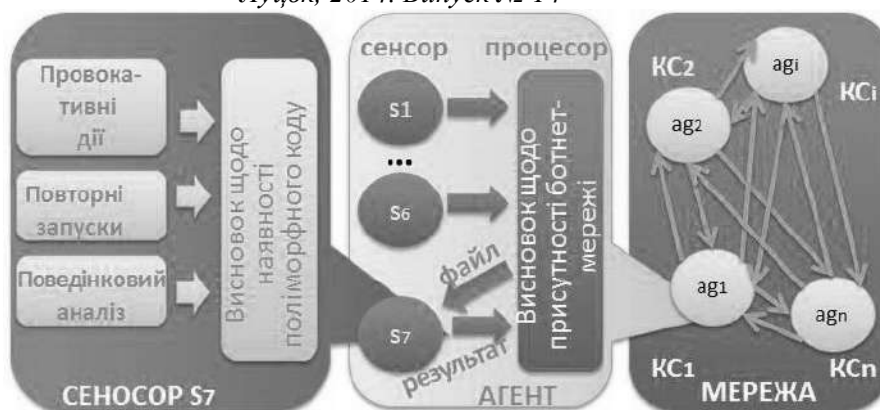


Рис. 1. Робота сенсора S7 в агенті мультиагентної системи, що функціонує в локальній мережі

Якщо в результаті провокативних дій виявлено створення нового файлу, або якщо у випадку повторних запусків виявлено зміну тіла файлу, або в результаті аналізу поведінки програми виявлено поведінку вірусу певного рівня поліморфізму, то сенсор S7 повідомляє процесор агента про необхідність блокування даного файлу, а також розсилання інформації про даний файл іншим агентам мультиагентної системи. Якщо перераховані дії по відношенню до файлу не виявляють ознаки присутності вірусу даний файл залишає сенсор S7.

Розроблена мультиагентна система здатна здійснювати виявлення бот-мереж на рівні 88-96%, проте в процесі антивірусного діагностування задіюються ресурси як комп'ютерних систем так і ресурси корпоративної мережі, що впливає на загальну ефективність антивірусного діагностування комп'ютерних систем.

Постановка проблеми. Таким чином, постає задача побудови методики визначення ефективності антивірусного діагностування комп'ютерних систем з урахуванням параметрів антивірусного діагностування, ресурсів операційної системи та комп'ютерних мереж, які використовує антивірусний засіб для діагностування.

Обчислення ефективності. Ефективність діагностування комп'ютерних систем (КС) на наявність шкідливого програмного забезпечення визначають агенти МАС. Сенсори, які містяться в агентах взаємодіють з одного боку з ШПЗ та з операційною системою - з іншого.

Таким чином, загальна ефективність антивірусного діагностування КС враховує результати антивірусного діагностування та програмно-апаратні ресурси, які використовуються в процесі антивірусного діагностування, і визначається:

$$P = \frac{E}{R}, \quad (1)$$

де E - ефективність роботи антивірусного засобу, яка включає достовірність роботи агента; R - ресурси, що залучаються для здійснення антивірусного діагностування.

Ресурсами, що залучаються для здійснення антивірусного діагностування, вважатимемо тривалість ДКС на наявність ШПЗ кожним з сенсорів $T_{S_1} \dots T_{S_7}$; середню тривалість обробки даних процесором агента T_A ; середній час обміну діагностичною інформацією між агентами T_N^A ; об'єми даних, що проходять від кожного з сенсорів V_{S_i} ; об'єми даних, що надходять процесору агента V_A ; об'єми даних, які обмінюються між агентами V_N^A . Ресурси, що залучаються для здійснення антивірусного діагностування $R = f(T_A, T_{S_1} \dots T_{S_7}, T_N^A, V_A, V_{S_1} \dots V_{S_7}, V_N^A)$, розраховуватимемо:

$$R = T_A \cdot V_A + T_{S_1} \cdot V_{S_1} + T_{S_2} \cdot V_{S_2} + T_{S_3} \cdot V_{S_3} + T_{S_4} \cdot V_{S_4} + T_{S_5} \cdot V_{S_5} + T_{S_6} \cdot V_{S_6} + T_{S_7} \cdot V_{S_7} + T_N^A \cdot V_N^A, \quad (2)$$

Для оцінки ефективності антивірусного діагностування введемо показник хибних спрацювань ДКС на наявність ШПЗ, який обчислимо за формулами:

$$X_M = \frac{\sum_{i=1}^s x_i}{X}, \quad (2)$$

де X - загальна кількість програм, x_i - кількість програм, віднесених антивірусним засобом до i -того типу шкідливого програмного забезпечення.

В процесі роботи антивірусного засобу явище хибних спрацювань також використовує ресурси, що відображається зниженням рівня ефективності антивірусного діагностування $R_X = f(X_M, T_{X_M}, V_{X_M})$ на величину:

$$R_X = X_M \cdot T_{X_M} \cdot V_{X_M}, \quad (3)$$

де X_M - показники хибних спрацювань; T_{X_M} - показники часу, затраченого на хибні спрацювання; V_{X_M} - об'єми даних, що проходять в процесі хибних спрацювань.

Отже, визначення ефективності діагностування КС на наявність шкідливого програмного забезпечення з урахуванням показника хибних спрацювань, складе:

$$P = \frac{E}{R + R_X}. \quad (4)$$

Достовірністю результатів роботи діагностування вважатимемо значення відношення виявленого ШПЗ до усієї кількості ШПЗ. Приймемо n_i , $i = \overline{1, s}$, $s \in \mathbb{N}$ як кількість ШПЗ i -того типу, k_i - кількість об'єктів ШПЗ, виявлених антивірусним засобом. Тоді достовірність результатів роботи діагностування D_M складе:

$$D_M = \frac{\sum_{i=1}^s \alpha_i \cdot k_i}{\sum_{i=1}^s \alpha_i \cdot n_i}, \quad (5)$$

де α_i - відсоток i -того класу від усього ШПЗ, $0 \leq \alpha_i \leq 1$.

Під тривалістю ДКС на наявність ШПЗ T_M розглядатимемо час t_b здійснення пошуку b_j -тої сигнатури або поведінки в базі

$$T_M = \sum_{j=1}^v t_b, \quad (6)$$

де t_p - час, необхідний для занесення $l = \overline{1, r}$ сигнатур та поведінок ШПЗ до бази.

Об'єми даних $V_{Mt_{ym}}$, що проходять в процесі ДКС на наявність ШПЗ агентом за одну умовну одиницю часу, визначимо як суму об'ємів даних, помножених на частоту запитів даних на одну умовну одиницю часу n_r :

$$V_{Mt_{ym}} = V_A^s \cdot n_r^s + V_{S1}^p \cdot n_r^p + V_{S2}^a \cdot n_r^a + V_{S3}^c \cdot n_r^c + V_{S4}^c \cdot n_r^c + V_{S5}^d \cdot n_r^d + V_{S6}^e \cdot n_r^e + V_{S7}^f \cdot n_r^f + V_N^g \cdot n_r^g, \quad (7)$$

де V_A^s - об'єм даних, що проходить при ДКС при обробці даних процесором агента; V_{S1}^p , V_{S2}^a , V_{S3}^c , V_{S4}^c , V_{S5}^d , V_{S6}^e , V_{S7}^f - об'єм даних, що використовується при роботі сенсорів агента; V_N^g - об'єм даних, що використовується агентами для обміну інформацією. Тоді з урахуванням часу необхідного для реалізації кожного етапу загальний об'єм даних, що проходять в процесі ДКС V_M обчислимо так:

$$V_M = V_{Mt_{ym}} \cdot n_{t_{ym}}. \quad (8)$$

а) ефективність роботи антивірусного засобу E , що визначається на основі максимізації значення достовірності діагностування КС на наявність ШПЗ $D_M \rightarrow \max$; цільове значення показника $E \rightarrow \max$;

б) ресурси, що залучаються для здійснення антивірусного діагностування, які визначаються на основі мінімізації значень середньої тривалість діагностування, середнього часу підготовки до діагностування та мінімізації об'ємів даних, що проходять в процесі ДКС, $T_A \rightarrow \min$, $T_N^A \rightarrow \min$, $T_{S1} \dots T_{S7} \rightarrow \min$, $V_N^A \rightarrow \min$, $V_{S1} \dots V_{S7} \rightarrow \min$, $V_A \rightarrow \min$, а також

мінімізації показників, що залучаються в процесі хибних спрацювань, $X_M \rightarrow \min, T_{X_M} \rightarrow \min, V_{X_M} \rightarrow \min$.

Дослідження характеристик та показників системи діагностування дають можливість здійснити визначення достовірності та ефективності антивірусного діагностування КС на наявність ШПЗ.

Основні результати дослідження. Для оцінки та порівняння загальної ефективності роботи антивірусної MAC у порівнянні з іншими відомими антивірусними засобами (АЗ) було проведено ряд експериментів.

Для дослідження ефективності антивірусних засобів було здійснено оцінку розробленої MAC, Panda antivirus, Eset Nod32, Kaspersky, Avira, Microsoft, Avast, Zillia, Dr. Web.

Експеримент проводився шляхом запуску 55 вірусних програм (ботів) [6-8].

Таблиця 1. Результати експериментів діагностування

Засіб антивірусного діагностування	достовірність роботи, %	тривалість ДКС на наявність ШПЗ, хв	середня тривалість ДКС, хв	середній час підготовки ДКС (обмін між агентами), хв	об'єми даних, що проходять в процесі ДКС, Мб	ефективності ДКС на наявність ШПЗ
MAC	95	29	1	1	2395	0,93
Panda	86	27	1	3	2389	0,87
Eset Nod32	89	22	1	3	2322	0,88
Kaspersky	91	30	1	2	2368	0,91
Avira	90	28	1	3	2322	0,88
Microsoft	76	17	1	2	2322	0,78
Avast	85	25	1	3	2380	0,81
Zillia	77	28	1	4	2340	0,76
Dr. Web	85	29	1	3	2322	0,81

Таблиця 2. Показники хибних спрацювань під час антивірусного діагностування

Засіб антивірусного діагностування	Показник ресурсів, використаних в процесі хибних спрацювань під час антивірусного діагностування, %
	MS Windows 7
MAC	5
Panda	3
Eset Nod32	4
Kaspersky	6
Avira	4
Microsoft	3
Avast	5
Zillia	5
Dr. Web	3

Загальна ефективність антивірусного діагностування представлена діаграмою на рис.2.

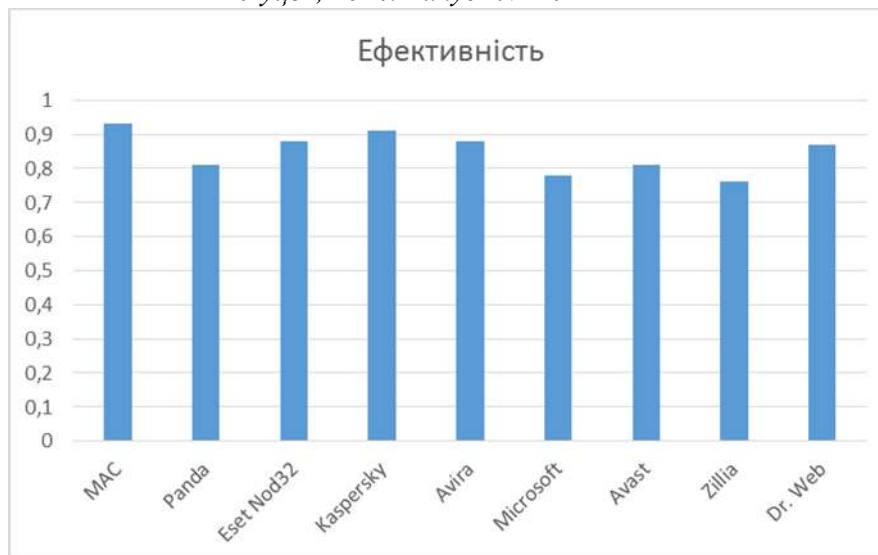


Рис. 2. Загальна ефективність антивірусного діагностування

Висновки. Для оцінки та порівняння загальної ефективності роботи запропонованої антивірусної мультиагентної системи з іншими відомими антивірусними засобами розглянуто параметри діагностування, які впливають на загальну ефективність антивірусного діагностування, а саме: достовірність роботи агента, тривалість діагностування та об'єми даних, що проходять в процесі антивірусного діагностування.

Завдяки використанню мультиагентного підходу для діагностування КС досягається високий рівень достовірності виявлення бот-мереж. Згідно запропонованої методики визначення ефективності антивірусного діагностування комп'ютерних систем з'ясовано, що розроблена мультиагентна система демонструє високу ефективність діагностування, яка склала 0,93.

1. Tim Rains Operating System Infection Rates: Application Vulnerabilities & Exploits Trend Up, Increase OS Infection Rates [Електронний ресурс] - Режим доступу : <http://blogs.technet.com/b/security/archive/2012/12/31/operating-system-infection-rates-vulnerabilities-amp-exploits-trending-up-increase-os-infection-rates.aspx>.

2. Williamson M. M. Virus throttling / M. M. Williamson, J. Twycross, J. Griffin // Virus Bulletin. – 2009.

3. VB100 Results Summary [Електронний ресурс] : Anti-Virus comparative. - <http://www.virusbtn.com/vb100/archive/summary>.

4. AV Comparatives laboratories [Електронний ресурс] – Access mode <http://www.av-comparatives.org>. – назва домашньої сторінки Інтернету.

5. Proactive/Retrospective test. [Електронний ресурс] : Anti-Virus comparative. – Режим доступу : <http://av-comparatives.org>. – назва домашньої сторінки Інтернету.

6. Савенко О.С. Процес діагностування комп'ютерних систем на наявність ботнет-мереж на основі мультиагентних технологій / О.С. Савенко, С.М. Лисенко, А.Ф. Кришук // Комп'ютерно-інтегровані технології : освіта, наука, виробництво: наук. журн. – 2013. — №11. – С.72–81.

7. Shoham Y. Multiagent Systems Algorithmic, Game-Theoretic, and Logical Foundations / Yoav Shoham, K. Leyton-Brown. - Cambridge University Press, 2009. – 504 p.

8. Alkhateeb F. Multi-Agent Systems / Faisal Alkhateeb, Eslam Al Maghayreh, Iyad Doush. - Modeling, Control, Programming, Simulations and Applications, 2011. – 522 p.

9. Savenko O. The Technique for Computer Systems Trojan Diagnosis in the Monitor Mode / Savenko O., Lysenko S. // Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications - USA, NJ 08855-1331: IEEE Operations Center, 2011 - vol.2, pp. 845-853.

10. Savenko O. Multi-agent based approach of botnet detection in computer systems / Savenko O., Lysenko S., Kryschuk A. // Computer Networks Communications in Computer and Information Science, 2012, Volume 291, pp. 171-180.

11. Smyth, B. Computing Patterns In Strings / B. Smyth // Williams. – 2006. – С. 496.

УДК 004.42

Н.А. Христинець, О.І. Міскевич

Луцький національний технічний університет

SHELL-ПРОГРАМУВАННЯ ПРОЦЕСІВ ФОНОВОГО РЕЖИМУ В ОС LINUX

Христинець Н.А., Міскевич О.І. Shell-програмування процесів фоновому режиму в ОС Linux. В роботі проведено дослідження способів побудови архітектури мережених сервісів з використанням shell-інтерпретатора, що дозволяє керувати взаємодією між системою та користувачем.

Ключові слова: процес, фоновий режим, потоки, програмування, інтерпретатор, shell, Linux

Христинець Н.А., Міскевич О.І. Shell-программирование процессов фоновому режима в ОС Linux. В работе проведено исследование способов построения архитектуры сетевых сервисов с использованием shell-интерпретатора, который позволяет управлять взаимодействием между системой и пользователем.

Ключевые слова: процесс, фоновый режим, потоки, программирование, интерпретатор, shell, Linux

Hrystynets N., Miskevych O. Shell-programming processes of the background in OS Linux. This paper studied the ways of constructing the architecture of network services using shell-interpreter which allows you to manage the interaction between the system and the user.

Keywords: process, background, streams, programming interpreter, shell, Linux

Постановка проблеми загальному вигляді і її зв'язок з важливими науковими та практичними завданнями. Як відомо, для ефективної роботи комп'ютера потрібно постійно слідкувати за станом операційної системи, встановленої на ньому. Проте, навіть у тих випадках, коли ОС функціонує належним чином, її налаштування не завжди відповідають оптимальним. Тому час від часу для деяких користувачів постає питання оптимізації або тонкого налаштування системи. Отже, питання оптимізації системи є досить актуальною в наш час. Надзвичайно велика кількість сервісів, що виконуються в системі, надає ймовірності того, що якийсь з них почне працювати невірно.

Аналіз останніх досліджень, у яких започатковано вирішення проблеми. У багатьох статтях різними авторами (Bozidar Levi, W. Richard. Stevens, С. Могильний та ін.) згадуються і навіть описуються способи побудови архітектури мережених сервісів (демонів). При цьому мало у кого з авторів є реальний досвід створення та оптимізації демонів, що працюють з десятками тисяч одночасних з'єднань, або керують, наприклад, гігабітним трафіком.

Цілі статті. Метою статті є програмна організація способу побудови архітектури мережного сервісу для ОС Linux за допомогою вбудованого інтерпретатора shell.

Виклад основного матеріалу дослідження. Крім вирішення проблем відслідковування сервісу екстенсивними методами в сучасних операційних системах (як то Linux) ця задача виконується в фоновому режимі, і при запуску системи стартує цілий пакет таких задач. Управління автоматизованими задачами, демони управління живленням і переметрами CPU, демони для друку, для ведення системних журналів – це далеко не весь перелік необхідних сервісів сучасних ОС. Так як компаніям-виробникам дистрибутивів важко дізнатись, що точно треба буде для роботи конкретного десктопу, вона намагається діяти по принципу надлишковості (тобто, зазвичай, кількість запропонованих сервісів значно перевищує необхідний мінімум). В результаті такого підходу комп'ютер може отримати демона управління живленням і частотою CPU для ноутбука або демоном для Bluetooth. Задіяність таких програм при роботі конкретного споживача може бути обмежена, а от об'єми пам'яті та ресурси процесора вони використовують.

Існує багато способів знайти і вимкнути непотрібні сервіси.

Найзручніше скористатися програмою *chkconfig*, що дозволяє редагувати необхідні сервіси. Linux, так само як і Unix, використовує різні рівні запуску для різноманітних сервісів. Для прикладу, рівень запуску 1 зазвичай використовується для аварійного завантаження в режимі однокористувацької системи. Це означає, що в ньому немає ні мережі, ні графічного робочого столу.

Перед тим як почати вимикати непотрібні сервіси, потрібно створити за допомогою нехитрої команди список сервісів, що були присутні до оптимізації (це звичайна застережливості на випадок некоректних подальших змін). Для цього слугує команда *chkconfig -A > services.save*, що перенаправить вивід команди *chkconfig* в файл з ім'ям *services.save*. Переглянувши його в текстовому редакторі, можна побачити, що він містить в точності те саме, що й вивід стандартної команди.

Отже, щоб вимкнути демон, потрібно виконати команду *chkconfig назва_демону off*, (ввімкнути – *chkconfig назва_демону on*)

Всі зміни будуть виконані лише після рестарту системи.

Якщо щось пішло не так, команда `chkconfig -s <services.save` – і вся підбірка сервісів, що була до цього, почне працювати.

Є ще один метод вимкнути непотрібне. Цей метод – програма `sysvconfig`. `Sysvconfig` – це утиліта, що має певний консольний інтерфейс і запускається з терміналу. Її по замовчуванні в дистрибутиві в основному немає, але її можна встановити командою `sudo aptitude install sysvconfig`.

Будь демон повинен приймати і обробляти мережеві з'єднання. Так як стек транспортних протоколів TCP/IP виріс із UNIX, а складовими таких ОС є файли, то можемо зробити висновок, що мережеві з'єднання є також файли, причому файли особливого типу, які можна відкривати, закривати, читати і писати стандартними функціями ОС для роботи з файлами.

Отже, насамперед, будь-який демон викликає системні функції:

`socket ()`, `bind ()`, а потім `listen ()` і в підсумку отримує файл спеціального типу. Параметри цих функцій і подальші дії демона дуже сильно залежать від застосовуваного транспортного протоколу (TCP, UDP, ICMP, RPD).

Усе, що може відбуватися з файлом спеціального типу «той, що слухає сокет» – це періодично виникаючі події типу «запит вхідного з'єднання». Демон може прийняти таке з'єднання функцією `accept ()`, яка створить новий файл, на цей раз вже типу «відкритий мережевий сокет». Імовірно, демон повинен прочитати з цього з'єднання запит, обробити його і відправити назад результат. При цьому, мережевий сокет – це вже більш-менш нормальний файл: хоч він і був створений не самим стандартним чином, принаймні з нього можна намагатися читати і писати дані. Але є й істотні відмінності від звичайних файлів, розташованих на файлової системі:

- всі події відбуваються дійсно асинхронно і з невідомої тривалістю за часом і будь-яка операція в гіршому випадку може зайняти десятки хвилин;
- з'єднання, на відміну від файлів, можуть закриватися «самі собою» в будь-який, найнесподіваніший момент;
- ОС не завжди повідомляє про що закрилося з'єднання, недіючі сокети можуть висіти по півгодини;
- з'єднання на клієнті і на сервері закриваються в різний час. Якщо клієнт спробує створити нове з'єднання і «довідправити» дані, можливо дублювання даних, а при неправильно написаному клієнті – і їх втрата. Також можлива наявність на сервері декількох відкритих з'єднань від одного клієнта;
- дані розцінюються як потік байтів і можуть буквально приходити порціями по 1 байту. Тому вважати їх, наприклад, рядками UTF-8 не можна;
- ніяких буферів крім тих, які надав сам демон, в мережі немає. Тому запис в сокет навіть 1 байта може заблокувати демон на десятки хвилин;
- будь-які помилки можуть траплятися в будь-якому місці, демон повинен коректно обробляти їх всі.

Найпростіший спосіб не дати сервісам впливати один на одного – це запустити для кожного з них окремий процес (тобто окрему копію програми). Недоліки цього методу очевидні – запуск окремого процесу це дуже ресурсномістка операція. Процес у всіх ОС є одиницею обліку системних ресурсів – пам'яті, відкритих файлів, прав доступу, квот і так далі. Якщо створюється демон віддаленого доступу до операційної системи на зразок Shell або FTP - користувач просто зобов'язаний запускати окремий процес від імені кожного зареєстрованого користувача, щоб правильно враховувати права доступу до файлів. Аналогічно, на сервері shared-хостингу одночасно, на одному «фізичному» порту крутяться сотні сайтів різних користувачів – і процеси потрібні арасхе для того, щоб сайти одних користувачів хостингу не могли потрапити до даних інших користувачів. Використання процесів не дуже впливає на продуктивність арасхе, про що свідчать результати бенчмарк з урахуванням аналізу обробки графіків запитів

Для відносно невеликої кількості одночасних з'єднань багатопотокова архітектура – найкращий вибір. Але якщо сполук дійсно багато, скажімо десять тисяч, перемикання між потоками починає займати занадто багато часу. Але навіть це – не головний недолік багатопотокової архітектури.

Головний аспект полягає в тому, що потоки не незалежні і можуть один одного блокувати. Якщо два потоки почнуть одночасно виконувати операції $a = b + c$ і $b = c + a$, вони заблокують один одного назавжди. Така ситуація називається клінчем, пошук і дозвіл клінчів – окрема тема

паралельного програмування. Але й без клінків потоки, якщо вони не знімають блокування швидко, можуть зупинити один одного на досить тривалі проміжки часу.

Крім того, атомарні операції фізично реалізуються за допомогою монопольного захоплення шини ОЗП і самого ОЗП. Робота безпосередньо з пам'яттю, а не з кешем, дуже повільна сама по собі, а в даному випадку ще й викликає інвалідацію відповідних кеш-ліній всіх ядер всіх інших процесорів сервера. Тобто, навіть у кращому випадку, за відсутності блокувань, кожна атомарна операція виконується досить довго і погіршує продуктивність інших потоків.

- Змінні трапляються за таких умов:
- Загальна черга нових сполук ;
- Загальна черга доступу до бази даних або подібних ресурсів ;
- Загальна черга запитів на виділення пам'яті (так-так, malloc () і new () можуть викликати блокування) ;
- Загальний журнал (log -файл) і загальні об'єкти підрахунку статистики.

Це тільки найочевидніші.

У деяких випадках існують способи обійтися без загальних змінних. Наприклад, від блокувань черги нових сполук можна відмовитися, якщо наділити один з потоків функцією «диспетчера», який буде таким чином роздавати завдання заздалегідь. Іноді вдається застосувати спеціальні «неблокуючі» структури даних. Але в цілому проблема взаємних блокувань в багатопотокової архітектури не вирішена.

Написання демона можливе на будь-якій мові (звичайно, мається на увазі мова високого рівня), наприклад, C, Perl (і навіть PHP). Написати демон у вигляді shell-скрипта досить проблематично, але вирішення такого роду завдання є моментом вдосконалення розуміння як архітектури ОС в цілому, так і специфіки використання інтерпретатора, а також можливостей застосування великої кількості консольних утиліт.

Визначимо загальні принципи роботи демона.

Найперше, слід пам'ятати, що демон – це звичайна програма, що виконується у фоновому режимі. Але так як демон буде запускатися з *init.d*, то на нього накладаються певні обмеження :

1. Демон має зберегти свій PID у файл, для того щоб потім можна було його коректно зупинити.

2. Необхідно виконати ряд підготовчих операцій для початку роботи у фоновому режимі.

У запропонованій у цій статті моделі демон функціонуватиме за наступним алгоритмом:

- 1) Відділення від керуючого терміналу і перехід у фоновий режим.
- 2) Поділ на дві частини: батько (моніторинг) і нащадок (функціонал демона).
- 3) Моніторинг стану процесу демона.
- 4) Обробка команди оновлення конфігу.
- 5) Обробка помилок.

Запропоноване програмне вирішення проблеми складається з скрипта, який рекомендується запуснути в фоновому режимі:

```
#!/bin/sh
set -euf
while true;do
logger -p local2.notice -f /etc/rsyslog/dmzfifo_access
done
```

Даний скрипт за допомогою програми *Logger* слухає *fifo* пайп *dmzfifo_access* і відправляє вміст з пайпа в *syslog*. Даний скрипт повинен знаходитись в фоновому режимі і постійно слухати пайп.

Наступним кроком є написаний скрипт для *init.d*:

```
#!/bin/sh
#
# processname: dmzfifo_accessd
DMZFIFO_BIN=/etc/rsyslog/dmzfifo_access.sh
KIND="DMZFIFO_ACCESSD"
```

```
. /etc/rc.d/init.d/functions
    start() {
        echo -n "$Starting $KIND services: "
        daemon /etc/rsyslog/dmzfifo_access.sh
        echo
    }
    stop() {
        echo -n "$Shutting down $KIND services: "
        killproc /etc/rsyslog/dmzfifo_access.sh
        echo
    }
    restart() {
        echo -n "$Restarting $KIND services: "
        killproc /etc/rsyslog/dmzfifo_access.sh
        daemon /etc/rsyslog/dmzfifo_access.sh
        echo
    }
    case "$1" in
        start)
            start
            ;;
        stop)
            stop
            ;;
        restart)
            restart
            ;;
        *)
            echo $"Usage: $0 {start|stop|restart}"
            exit 1
    esac
    exit $?

```

Дане вирішення проблеми організації роботи процесів в фоновому режимі може бути вдосконалене, залежно від потреб користувача.

Висновки. Навіть якщо користувачі запрограмованих процесів фоновому режиму не зареєстровані в операційній системі, але до програмного сервісу висуваються підвищені вимоги з безпеки, то в загальному виділення окремого процесу під кожного користувача є дуже розумним архітектурним рішенням. Це не дозволить користувачам отримувати або блокувати доступ до даних інших користувачів, навіть якщо вони знайдуть баг у написаному демоні, що дозволяє читати чужі дані або просто руйнуючий процес. Нарешті, у кожного процесу маєтись власний адресний простір, і різні процеси не заважають один одному користуватися пам'яттю. Тобто, особливості написання Shell-додатків для процесів фоновому режиму в ОС Linux, є сучасним вирішенням проблем організації роботи процесів.

1. Томас Р., Йейтс Дж. Операционная система UNIX. Руководство для пользователей. – М.: Радио и связь, 1986. -352 с.
2. Банахан М., Раттер Э. Введение в операционную систему UNIX. -М.: Радио и связь, 1986. - 341 с.
3. Тихомиров В.П., Давидов М.И. Операционная система UNIX: Инструментальные средства программирования. -М.: Финансы и статистика, 1988. -206 с.
4. Баурн С. Операционная система UNIX. -М.: Мир, 1986. -462 с.
5. <http://www.citforum.ru/ftp/pub/os/shell-win.zip> - Інтерпрітатор командного мови Shell

УДК 004.414

П.С. Шолом, С.В. Гринюк, С.Р. Семчук
Луцький національний технічний університет

СИСТЕМА КОНТРОЛЮ ПРОСТОРОВИМ ПОЛОЖЕННЯМ МОНИТОРА

Шолом П.С., Гринюк С.В., Семчук С.Р. Система контролю просторовим положенням монітора. Встановлено можливість поліпшення роботи з комп'ютерним монітором шляхом розробки системи контролю за його просторовим положенням; розроблено апаратне та програмне забезпечення системи.

Ключові поняття: система, двигун, драйвер, LPT-порт, Delphi
Рис. 6, Літ. 12

Шолом П.С., Гринюк С.В., Семчук С.Р. Система контролю просторовим положенням монітора. Установлена можливість удешевлення роботи з комп'ютерним монітором шляхом розробки системи контролю за його просторовим положенням; розроблено апаратне та програмне забезпечення системи.

Ключевые понятия: система, мотор, драйвер, LPT-порт, Delphi
Рис. 6, Лит. 12

Sholom P.S., Grynyuk S.V., Semchuk S.R. Spatial Position Control System of the Monitor. The possibility of improving of the work with a computer monitor by developing a spatial position control system was established; system hardware and software was developed.

Keywords: system, motor, driver, LPT, Delphi
Fig. 6, Ref. 12

Постановка проблеми. Перші комп'ютери створювалися виключно для обчислень (що відображено в назвах «комп'ютер» і «ЕОМ»). Навіть найпримітивніші комп'ютери в цій галузі у багато разів перевершують людей. Іншою сферою застосування комп'ютерів стали бази даних. Перш за все вони були потрібні урядам і банкам, які вимагають вже складніших комп'ютерів з розвиненими системами вводу / виводу та зберігання інформації. Ще одним застосуванням було управління різноманітними пристроями. Тут розвиток йшов від вузькоспеціалізованих пристроїв (часто аналогових) до поступового впровадження стандартних комп'ютерних систем, на яких запускаються керуючі програми. Крім того, все більша частина техніки починає включати в себе керуючий комп'ютер. Комп'ютери розвинулися настільки, що стали головним інформаційним інструментом в багатьох галузях освіти, науки і виробництва. Тепер майже будь-яка робота з інформацією найчастіше здійснюється саме через комп'ютер. Персональний комп'ютер (ПК) – електронна обчислювальна машина, призначена для особистого використання, ціна, розміри та можливості якого задовольняють потреби багатьох людей.

Ускладнення функціональної структури діяльності у зв'язку із застосуванням електронно-обчислювальних систем, персональних комп'ютерів (ПК), пред'являє нові, іноді підвищені вимоги до організму людини. Недооцінка ролі людського фактора при проектуванні і створенні ПК неминуче відбивається на якісних і кількісних показниках діяльності користувачів. Широке впровадження комп'ютерної техніки, в т.ч. ПЕОМ (ПК), зумовило значну зміну характеру праці, його змісту та умов, в яких він здійснюється.

Ще у 1968 році дослідник Дуглас Енгельбарт показав те, що стало звичним на початку 21 століття – електронну пошту, гіпертекст, текстовий процесор, відеоконференції та маніпулятор «мишу». Разом з тим у той час використання комп'ютера (ЕОМ) було занадто дорогими для індивідуального користування (у бізнесі чи освіті).

У 1970 році компанією Hewlett-Packard було презентовано розробку обчислювальної системи (комп'ютера), яка повністю розміщувалась на верхній частині столу і мала клавіатуру, маленький дисплей(монітор) і принтер. Першим персональним комп'ютером вважається Xerox Alto, розроблений у 1973 році Науково-дослідним центром компанії Xerox (PARC) у Пало-Альто (Каліфорнія, США). Ранні персональні комп'ютери (мікрокомп'ютери) були цікавими головним чином для фахівців і аматорів радіоелектронних пристроїв.

З початку 1980-х років комп'ютери широко застосовуються для персонального (у тому числі і домашнього) використання, розробляється програмне забезпечення широкого спектру. Згідно [1] в 2009 році світові поставки персональних комп'ютерів склали 308,3 млн штук.

Аналіз останніх досліджень і публікацій. У зв'язку зі стрімким розвитком комп'ютерів, прискоренням їх серійного виробництва та розширенням сфер застосування активізувалась наукова робота, яка здійснює комплексне вивчення особливостей виробничої діяльності людини в системі «людина – техніка» з метою забезпечення її ефективності, безпеки та комфорту. Наука

вивчає допустимі фізичні, нервові та психічні навантаження на людину в процесі праці, проблеми оптимального пристосування навколишніх умов виробництва для ефективної праці. Існують цікаві просторові аспекти цієї науки: планування робочого місця передбачає раціональне розміщення у просторі матеріальних елементів виробництва, зокрема устаткування, технологічного та організаційного оснащення, а також робітника. Вивченням цих питань займались А.А. Абрамов, В.В. Адамчук, В.І., Варгунін, В.Г. Макушин та ін. Зокрема, у роботі І.Й. Літвача [2] висвітлені питання, пов'язані із зоровою роботою з комп'ютером та її наслідками, вимогами до візуальних параметрів. У праці В.М. Муніпова та В.П. Зінченка [3] детально розглянуто ергономіку апаратних та програмних засобів обчислювальної техніки, роботу з дисплеями та вимогами до них, організацію комп'ютеризованих робочих місць. А.П. Широков [4] досліджує особливості взаємодії людини та техніки.

Виділення невирішених раніше частин загальної проблеми. Робота із застосуванням персональних ПЕОМ (ПК) пов'язана зі значними зоровими і нервово-психологічними навантаженнями, що підвищує вимоги до організації праці користувачів ПК. Тому доцільним є створення систем, які спрощують виконання цих вимог.

Метою дослідження є спрощення забезпечення ергономічних параметрів положення людини відносно монітора персонального комп'ютера; розробка апаратного та програмного забезпечення для автоматизації зміни просторового положення дисплею.

Виклад основного матеріалу дослідження. Система дистанційного контролю просторовим положенням монітора включає в себе апаратне та програмне забезпечення.

До складу апаратного забезпечення входить:

- дерев'яна підставка (два дерев'яних круги однакового розміру);
- ковзаючий механізм;
- кроковий двигун.

Діаметр платформи визначено в результаті аналізу стійок моніторів різних моделей та виробників і становить 20 см.

Ковзаючий механізм розташований всередині дерев'яних кругів і має радіус 8 см. відносно центру платформи.

У конструкції крокового двигуна використовується блок електромагнітних котушок (обмоток збудження) на статорі і певна кількість постійних магнітів, що знаходяться на роторі. При подачі електричного струму котушки генерують електромагнітне поле, яке притягує або відштовхує найближчий магніт на валу двигуна. Це призводить до обертання вала двигуна, а також дає можливість точно регулювати кут його повороту. Однією з особливостей конструкції крокових двигунів також є збереження положення вала після відключення живлення.

Найбільш поширені два типи крокових двигунів – уніполярні та біполярні. На відміну від біполярних, уніполярні крокові двигуни мають 5 або 6 контактів, через які здійснюється управління чотирма котушками. Живлення (плюс) на кожен з котушок подається по загальній шині живлення, що і визначає назву типу двигуна – уніполярний. Через інший вивід кожна з котушок підключається до шини із від'ємним знаком. Через цей вивід також подаються імпульси, що керують обертанням.

Для забезпечення руху дерев'яної підставки через ковзаючий механізм було використано двигун 28BYJ-48 (рис. 1). Це п'ятивольтовий кроковий двигун. Він є типовим представником класу уніполярних двигунів. Часто застосовується в робототехніці, DIY-пристроях, для управління жалюзі, кондиціонерів, невеликих вентиляторів і т.п. Всі технічні параметри відповідають електронному стандарту SJ/T10689-95.



Рис. 1. Кроковий двигун 28BYJ-48

Технічні характеристики двигуна[5]:

- робоча напруга: 5 В постійного струму;
- кількість полюсів: 4;
- кількість кроків на оборот (на вихідному валу): 4096;
- коефіцієнт зміни швидкості: 1/64;
- кут кроку: $5.625^\circ / 64$
- опір постійному струму: $50 \text{ Ом} \pm 7\%$ (25 °C);
- номінальна частота: 100 Гц;
- номінальний опір обмоток (25 °C): 50 Ом;
- частота холостого ходу (по часовій стрілці): 600 Гц;
- частота холостого ходу (проти часової стрілки): 1000 Гц;
- обертальний момент (по часовій стрілці при частоті 120 Гц): 34,3 мН·м;
- обертальний момент: 34,3 мН·м;
- момент тертя (опір обертанню): 600-1200 гс·см;
- номінальна тяга: 300 гс·см;
- опір ізоляції: 10МОм (500 В);
- клас ізоляції: А;
- рівень шуму: менше 35 Дб.

Драйвер на мікросхемі ULN2003 DARLINGTON (рис. 2) призначений для керування кроковим двигуном [6, 7]. Побудований на основі транзисторів Дарлінгтона. Складений транзистор (або транзистор Дарлінгтона) – це об'єднання двох чи більше біполярних транзисторів з метою збільшення коефіцієнта підсилення по струму. Такий транзистор використовується в схемах, що працюють з великими струмами (наприклад, в схемах стабілізаторів напруги, вихідних каскадів підсилювачів потужності) і у вхідних каскадах підсилювачів, якщо необхідно забезпечити великий вхідний імпульс. Складений транзистор має три виводи (база, емітер і колектор), які еквівалентні виводам звичайного одиночного транзистора.

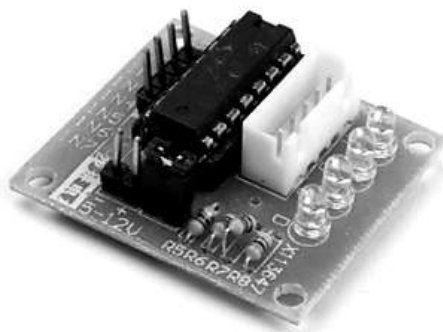


Рис. 2. Драйвер крокового двигуна ULN2003

Драйвер може здійснювати управління 4-фазним 5-провідним кроковим двигуном з напругою від 5 до 12 В. Має чотири світодіода для індикації роботи (по світодіоду на кожен канал). Струм – 500 мА. Дуже простий у використанні з Arduino сумісними мікроконтролерами або іншими платформами розробки. Керуючий сигнал подається на нього з комп'ютера через LPT кабель. Схема підключення зображена на рисунку 3.

Живлення двигуна здійснюється за допомогою блоку живлення на 5В від мережі 220 В.

Програмне забезпечення системи розроблене у середовищі Delphi 2010. Основними елементами програми є дві керуючі кнопки, які здійснюють повороти праворуч та ліворуч із наперед заданою швидкістю (рис. 4).

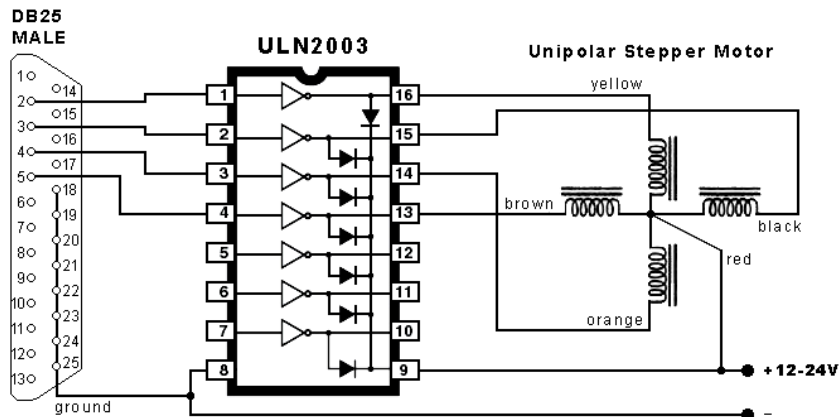


Рис. 3. Схема підключення LPT кабелю та крокового двигуна до драйвера ULN2003

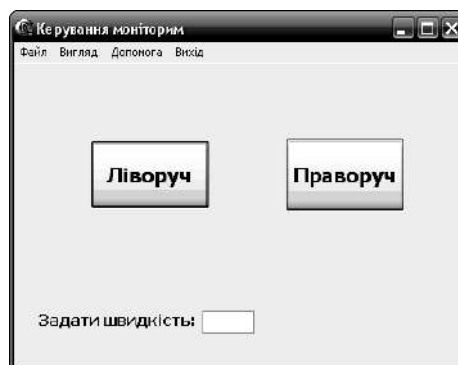


Рис. 4. Інтерфейс програмного забезпечення для управління кроковим двигуном

Для роботи з паралельним LPT-інтерфейсом (рис. 5) у середовищі Delphi використано динамічну бібліотеку `inpout32.dll`. Її потрібно скопіювати в каталог з проектом. Бібліотека працює тільки в 32-розрядних версіях ОС Windows. Для забезпечення правильної роботи бібліотеки потрібно оголосити такі функції:

```
function Inp32(PortAdr: word): byte; stdcall; external 'inpout32.dll';  
function Out32(PortAdr: word; Data: byte): byte; stdcall; external 'inpout32.dll';
```

LPT-порт має 8 двонаправлених ланцюгів (з 2-го по 9-й вивід) з адресою \$378 (888 – у десятковій системі числення); 5 однонаправлених ланцюгів, що призначені тільки для вводу (з 10-го по 13-й і 15 виводи) з адресою порту \$379 (889); 4 однонаправлених ланцюга, призначених тільки для виводу (1, 14, 16 та 17-й виводи) з адресою \$37A (890), де всі виводи, крім 16-го, інверсні (рис. 6).

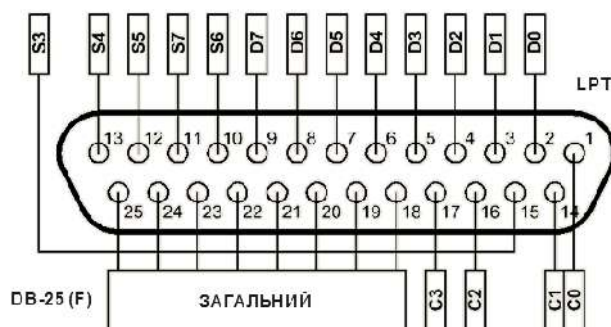


Рис. 5. Інтерфейс роз'єму LPT-порту

\$798 (888) data				\$379 (889) status				\$37A (890) control									
№ контакту роз'єму	Назва ланцюга	Номер біта	Розрядність	Двонаправлений вхід / вихід	№ контакту роз'єму	Назва ланцюга	Номер біта	Розрядність	Тільки вхід	№ контакту роз'єму	Назва ланцюга	Номер біта	Розрядність	Тільки вихід			
2	D0	0	1									1	Strobe		0	1	
3	D1	1	2									14	AutoLF		1	2	
4	D2	2	4									16	Init		2	4	
5	D3	3	8			15	Error	3		8		17	SelIn		3	8	
6	D4	4	16			13	Sel	4		16							
7	D5	5	32			12	PE	5		32							
8	D6	6	64			10	ASK	6		64							
9	D7	7	128			11	Busv	7		128							

Рис. 6. Інтерфейс програмного забезпечення для управління кроковим двигуном

Висновки. У роботі було досліджено ергономіку роботи з персональними комп'ютерами та можливість поліпшення роботи з монітором шляхом розробки апаратного забезпечення у вигляді підставки. У середовищі Delphi 2010 було розроблено прототип програми, яка здійснює управління підставкою для забезпечення зміни положення монітора в просторі.

1. Поставки комп'ютерів в мирі виростуть в 2010 году на 22% [Електронний ресурс]. – Режим доступу: <http://www.finmarket.ru/news/1508954/>
2. Ергономіка – заботливая наука [Електронний ресурс]. – Режим доступу: <http://www.osp.ru/school/1999/05/13031884/>
3. Мунипов В.М., Зинченко В.П. Ергономіка: человекоориентированное проектирование техники, программных средств и среды: Учебник. – М.: Логос, 2001. – 356 с.
4. Широков А.П. Основы эргономики: учеб. пособие / А.П. Широков. – Хабаровск: Изд-во ДВГУПС, 2006. – 117 с.
5. 28BYJ-48 – 5V Stepper Motor [Електронний ресурс]: Kiatronics Ltd. – Режим доступу: <http://www.sensors.co.nz/datasheet/28BYJ-48%20Stepper%20Motor.pdf>
6. 4 Phase ULN2003 Stepper Motor Driver PCB [Електронний ресурс]: Kiatronics Ltd. – Режим доступу: <http://www.sensors.co.nz/datasheet/942102570285.pdf>
7. ULN2002A, ULN2003A, ULN2003AI, ULN2004A, ULQ2003A, ULQ2004A. HIGH-VOLTAGE, HIGH-CURRENT DARLINGTON TRANSISTOR ARRAYS [Електронний ресурс]: Texas Instruments. – Режим доступу: ftp://imall.iteadstudio.com/Modules/IM120723012/DS_IM120723012.pdf
8. Коноваленко І.В., Федоріє І.С. Системне програмування у Windows з прикладами на Delphi. Навчальний посібник для технічних спеціальностей вищих навчальних закладів. – Тернопіль: ТНТУ ім. І.Пулужя, 2012. – 320с.
9. А.Н.Вальвачев, К.А.Сурков, Д.А.Сурков, Ю.М.Четырько. Программирование на языке Delphi. Учебное пособие. – Режим доступа: <http://www.rsdn.ru/?summary/3165.xml>, 2005.
10. Платт Ч. Электроника для начинающих: Пер. с англ. – СПб.: БХВ-Петербург, 2012. – 480 с.: ил. – (Электроника).
11. Максимихин Б.А. Технологические процессы пайки электромонтажных соединений. – Л.: Энергия. Ленингр. отд-ние, 1980. – 80 с.
12. Брайант Р., О'Халларон Д. Компьютерные системы: архитектура и программирование. Пер. с англ. – СПб.: БХВ-Петербург, 2005. – 1104 с.

УДК 621.3.078.4

К.Я.Бортник

Луцький національний технічний університет

ІННОВАЦІЙНІ ТЕХНОЛОГІЇ В ЕНЕРГОЗБЕРЕЖЕННІ ПО КОМПЕНСАЦІЇ РЕАКТИВНОЇ ПОТУЖНОСТІ

Бортник К.Я. Інноваційні технології в енергозбереженні по компенсації реактивної потужності. Стаття присвячена питанням, пов'язаним з компенсацією реактивної потужності в електричних розподільних мережах. Проведено аналіз різних способів компенсації, розглянуті оптимальні схеми приєднання пристроїв компенсації. Система компенсації реактивної потужності розглянута як комплексна задача, для рішення якої необхідно врахувати різні параметри, пов'язані з експлуатаційними режимами роботи, наявністю джерел гармонійних перекручувань, терміном служби і ступенем надійності устаткування, а також цілий ряд інших факторів та їхній взаємний вплив.

Для вирішення даної задачі пропонується багаторівнева системи обробки інформації з поділом функцій, виконуваних кожним рівнем, яка враховує змінну безліч параметрів.

Ключові слова: реактивна потужність, конденсаторні батареї, резонансна частота.

Бортник Е.Я. Иновационные технологии в энергосохранении по компенсации реактивной мощности. Статья посвящена вопросам, связанным с компенсацией реактивной мощности в электрических распределительных сетях. Проведен анализ разных способов компенсации, рассмотрены оптимальные схемы присоединения устройств компенсации. Система компенсации реактивной мощности рассмотрена как комплексная задача, для решения которой необходимо учесть разные параметры, связанные с эксплуатационными режимами работы, наличия источников гармоничных искажений, сроком службы и степенью надежности оборудования, а также целый ряд других факторов и их взаимное влияние.

Для решения данной задачи предлагается многоуровневая системы обработки информации с разделением функций, исполняемых каждым уровнем, которая учитывает изменение большого количества параметров

Ключевые слова: реактивная мощность, конденсаторные батареи, резонансная частота.

Bortnyk K. Innovative technologies in energy efficiency for reactive power compensation. The Article is dedicated to questions, connected with compensation of the reactive power in electric network. The Organized analysis of the miscellaneous media coverage, are considered optimum schemes of the joining device to compensations. The System to compensations to reactive power is considered as complex problem, for decision which necessary to take into account the different parameters, connected with working state of working, presence of the sources of the harmonious distortion, lifetime and degree to reliability of the equipment, as well as variety of the other factor and their mutual influence.

Layered system information handling is offered for decision problems with division function, performed each level, which takes into account change big amount parameter.

Key words: reactive power, capacitor banks, the resonant frequency.

Актуальність даної проблеми полягає в тому, що енергозабезпечення різних галузей промисловості потребує скорочення розходу споживання електроенергії. Система компенсації реактивної потужності розглядається як комплексна задача, для рішення якої необхідно враховувати різні параметри, які пов'язані з експлуатаційними режимами, роботи різних джерел. Пошук найбільш ефективних засобів економії електроенергії є важливим чинником по удосконаленню технологічного процесу.

Проблема полягає в тому, що при збільшенні вживаної енергії, збільшується і реактивна складова, яка йде на нагрів електрообладнання і не являється корисною, а разом з цим росте і ціна на витрачену енергію.

Збільшення витрат на реактивну енергію несе разом з тим і негативні наслідки, такі як:

- збільшення вживаної потужності;
- збільшення падіння напруги в кабельних мережах;
- збільшення втрат на нагрів струмопровідних мереж;
- скорочення строку праці електрообладнання та інше.

Одним із ефективних способів, який ми будемо розглядати в даній статті, є компенсація реактивної потужності з застосування конденсаторних батарей з регульованою або нерегульованою місткістю.

В нерегульованих конденсаторних установках немає можливості точно оцінювати витрати на споживану електроенергію. Це є негативний фактор по вживанню нерегульованих конденсаторних установок.

Для вирішення даної проблеми, одним із ефективних способів, який ми будемо розглядати в даній статті, є компенсація реактивної потужності з регульованими конденсаторними установками.

Принцип роботи регульованих конденсаторних батарей полягає в зміні коефіцієнта потужності споживачів відповідно до заданого шляху ступінчастого регулювання місткості батареї конденсаторів.

Регулятор компенсації реактивної потужності є параметричним мікропроцесорним контролером, оснащеним необхідними пристроями введення/виведення інформації і команд, який здійснює регулювання коефіцієнта потужності відповідно до заданої установки.

Однак, відповідні методи мають недоліки:

-небезпека виникнення резонансних коливань в контурі "конденсатор - трансформатор" внаслідок наявності гармонійних складових;

-зниження точності налаштування компенсації, пов'язане з погрішностями вимірювальних трансформаторів;

-складність комутаційної апаратури при підключенні конденсаторних батарей на високій стороні;

-недоліки, пов'язані з урахуванням характеру навантаження, експлуатаційних режимів роботи, конфігурації схеми та інших параметрів при розрахунку місткості конденсаторних батарей.

Як показує досвід експлуатації конденсаторних батарей, який проводився представниками "Волиньобленерго", повна компенсація реактивного навантаження недоцільна, оскільки підвищення потужності компенсуючого пристрою призводить до зниження його ефективності.

На підставі розрахункових формул був побудований графік зміни величини зниження втрат активної потужності в залежності від $Q_{бк}$ (рис.1).

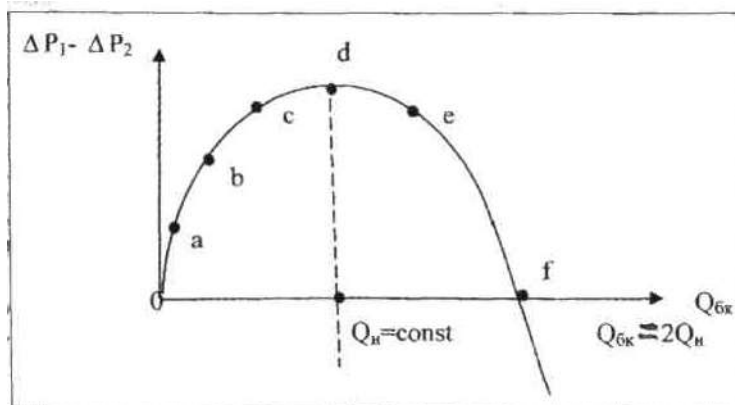


Рис 1. Графік зміни величини зниження втрат активної потужності в залежності від потужності конденсаторної батареї $Q_{бк}$

З графіка видно, що найбільша ефективність можлива на ділянці 0 - b. У точці d настає момент перекомпенсації з подальшим зростанням потужності конденсаторної батареї і зниження ефективності. Ефект компенсації значно збільшується при аварійному живленні лінії, на якій встановлені конденсатори, до того ж, наявність конденсаторів дозволяє підтримувати напругу на номінальному рівні, що сприятливо позначається на експлуатації електрообладнання.

Проаналізуємо вплив величини втрат в системі електропостачання, при використанні компенсуючих пристроїв.

Втрати в системі електропостачання обумовлені наступними параметрами:

- для активної потужності

$$\Delta P = \frac{P^2 + Q^2}{U^2} * R, \quad (1)$$

де P - активна потужність, що передається по лінії;

Q - реактивна потужність;

U - напруга лінії;

R - активний опір лінії;

- для реактивної потужності

$$\Delta Q = \frac{P^2 + O^2}{U^2} * X, \quad (2)$$

U - реактивний опір лінії.

При використанні компенсації реактивної потужності за допомогою компенсуючих пристроїв, наприклад, конденсаторних батарей, формули (1) і (2) перетворюються в наступний вигляд:

$$\Delta P = \frac{P^2 + (Q - Q_{\text{бк}})^2}{U^2} * R, \quad (3)$$

$$\Delta Q = \frac{P^2 + (Q - Q_{\text{бк}})^2}{U^2} * X, \quad (4)$$

де $Q_{\text{бк}}$ - потужність встановлених конденсаторних батарей.

Як показують формули (3) і (4) ефективність компенсації реактивної потужності залежить від величини, яка в свою чергу визначається точністю настройки регулятора.

Розрахунок потужності регульованої конденсаторної установки вимагає обліку ряду чинників, а також їх взаємного впливу.

Важливим чинником є схема підключення розподільних пристроїв.

Як правило, конденсаторні батареї підключаються по одній до кожної секції РП, використовуючи для вхідного сигналу по струму трансформатори струму на введенні в секцію.

Підключення конденсаторних батарей на низькій стороні трансформатора напруги має наступні переваги:

- виключається залишковий заряд на конденсаторах, що у свою чергу підвищує рівень безпеки технічного персоналу при проведенні регламентних робіт;
- немає необхідності використовувати розрядні опори;
- відсутня комутаційна апаратура, що підвищує надійність експлуатації електротехнічного устаткування.

Наступним чинником, що впливає на ефективність роботи компенсуючих пристроїв, є конфігурація схеми підключення конденсаторних батарей. Найбільш ефективними є установки з несиметричною конфігурацією, оскільки можуть забезпечити як точне регулювання, так і мінімальну частоту комутації східців при меншій вартості, в порівнянні з симетричними.

Компенсація реактивної потужності призводить до збільшення коефіцієнта потужності (КП) $\cos\varphi$, і при величинах КП системи, що перевищують 0,97, і наявності джерел гармонійних складових, може виникнути явище резонансу, що призводить до збільшення струму компенсації, перегрівання і виходу з ладу устаткування.

Якщо резонансна частота контура "конденсатори - трансформатор" близька до гармоніки викликаній одним з навантажень, струм цієї гармоніки може циркулювати в контурі, викликаючи високу напругу в лінії. Струм компенсації в цьому випадку може перевищити номінальне значення більш ніж в 2 або 3 рази від його номінальної величини. Резонанс може виникнути на будь-якій частоті, хоча у більшості випадків джерела синусоїдального струму існують на 5-ій, 7-ій, 11-ій і 13 гармоніках [2].

Як правило, запобігти цьому дозволяє регулятор, що вимірює гармоніки і відключає частину потужності конденсаторної батареї. Проте, в цьому випадку досягнення необхідного КП вже неможливе - а значить, ефективність установки істотно знижується.

Реальним захистом можуть бути тільки спеціальні реактори, що встановлюються в кожному ступені і забезпечують зміщення резонансної частоти контура "конденсатор-трансформатор" нижче домінуючої гармоніки або фільтрацію гармонік.

Фільтрація гармонік певної частоти (чи декількох частот відразу), виключає проникнення цих гармонік в живлячу мережу. Як правило, застосовуються пасивні фільтри 5, 7, 11, 13 гармонік і гармонік вище 15-й [2].

Недоліками пасивних фільтрів гармонік є дорожнеча, пригнічення невеликої кількості гармонік вимагають, трудомісткість робіт, пов'язаних з аналізом розподільної електричної мережі.

Для усунення вказаних недоліків пропонується використовувати активні фільтри.

Крім того, до переваг активних фільтрів відносяться зниження витрат на технічне обслуговування і управління системою фільтрів гармонік, простий монтаж і налаштування, динамічну компенсацію широкого спектру гармонік.

Таким чином, вдосконалення системи компенсації зводиться до забезпечення комплексного інженерного підходу, що враховує усі особливості системи електропостачання.

Враховуючи велику кількість параметрів, що описують систему компенсації реактивної потужності, побудова схеми регулювання компенсуючих пристроїв передбачає обробку значної кількості сигналів і даних. Введення цих сигналів повинне забезпечуватися в режимі реального часу і шляхом введення даних як результатів розрахункових і інших операцій (наприклад, з використанням реляційних або ієрархічних баз даних).

Складність і трудомісткість поставленого завдання не дозволяє використовувати традиційні схеми збору і обробки інформації. Одним з рішень цієї задачі може бути розробка багаторівневої системи обробки інформації з розподілом функцій, що виконуються кожним рівнем. Крім того, ця система не повинна використовувати строго обмежену кількість сигналів (даних), а необхідно передбачити можливість гнучкого управління і обліку змінної безлічч параметрів, що описують зовнішні дії, взаємний їх вплив один на одного. При цьому основним критерієм ефективності роботи цієї системи є економічна доцільність, тому ця система має бути досить універсальна і призначена для вирішення широкого спектру завдань управління.

Висновки.

1. Ефективність компенсації реактивної потужності залежить від величини місткості конденсаторної батареї, яка у свою чергу визначається точністю налаштування регулятора.
2. Налаштування регульованої конденсаторної установки визначається наступними чинниками: реальна споживана потужність і характер навантаження, визначуваними техніко-економічними і статистичними показниками роботи мережі; схема підключення розподільних пристроїв; конфігурація схеми підключення конденсаторних батарей; розрахунок і використання фільтрів гармонік електромережі; погрішності вимірювальних трансформаторів струму і напруги.
3. Вдосконалення системи компенсації зводиться до забезпечення комплексного інженерного підходу, що враховує усі особливості системи електропостачання
4. Одним з рішень цієї задачі може бути розробка багаторівневої системи обробки інформації з розподілом функцій, що виконуються кожним рівнем. Крім того, ця система не повинна використовувати строго обмежену кількість сигналів (даних), а необхідно передбачити можливість гнучкого управління і обліку змінної безлічч параметрів, що описують зовнішні дії і взаємний їх вплив один на одного.

Для вирішення даної задачі пропонується багаторівнева системи обробки інформації з поділом функцій, виконуваних кожним рівнем, яка враховує змінну безліч параметрів.

1. А. М. Василянський, Р. Р. Мамошин, Г. Би. Якимов. Совершенствование системы тягового электроснабжения железных дорог, электрифицированных на переменном токе 27,5 кВ, 50Гц.// Железные дороги мира, 08-2002. - С. 73-75.
2. Ю. С. Железко. О нормативных документах в области качества электроэнергии и условий потребления реактивной мощности./ Электрика, 2003. № 1.- С. 9-16.
3. Галыгина О. С., Заугольников В. Ф. О некоторых аспектах учета и потерь электроэнергии в предприятиях электросетей./ Энергетик, 2004, №5.- С. 18-22.

УДК 654.92

В.А. Брей, Н.В. Здолбіцька, А.П. Здолбіцький
Луцький НТУ

ЗВУКОВА СИГНАЛІЗАЦІЯ НА PIC16F84 МІКРОКОНТРОЛЕРІ

В.А. Брей, Н.В. Здолбіцька, А.П. Здолбіцький. Звукова сигналізація на PIC16F84 контролері. У даній статті подано опис звукової сигналізації на PIC мікроконтролері. Здійснено огляд використаного контролера та інших елементів для створення повноцінного охоронного пристрою. Розглянуто питання доцільності та необхідності пристрою.

Ключові слова: сигналізація, PIC, мікроконтролер, давач.

В.А. Брей, Н.В. Здолбицкая, А.П. Здолбицкий. Звуковая сигнализация на PIC16F84 контроллере. В этой статье представлено описание звуковой сигнализации на PIC микроконтроллере. Осуществлен обзор использованного контроллера и других элементов для создания полноценного охранного устройства. Рассмотрены вопросы целесообразности и необходимости устройства.

Ключевые слова: сигнализация, PIC, микроконтроллер, датчик.

V.A. Bray, N.V. Zdolbitska, A.P. Zdolbitskyi. Audible alarm on PIC16F84 controller. This article contains a description of the sound alarm on PIC microcontroller. Made the survey used by the controller and other elements to create a full circuit of the device. The question of the appropriateness and necessity of the device is offered.

Keywords: alarm, PIC, microcontroller, sensor.

Рис. 2. Літ. 11.

Постановка проблеми. Сучасні системи безпеки (охоронна сигналізація, відео і аудіодомофони, відеоокуляри) відносно недавно увійшли до нашого життя. Але із-за нестабільності економічної обстановки в стані, яка веде до загострення соціальних відносин, із-за підвищення рівня злочинності вони стали набувати великої актуальності. Спочатку ці системи були більше «іграшками» для забезпечених людей, але поступово вони стають нормою і речами, що увійшли до вжитку і необхідними для масових верств населення. Не дивлячись на все більшу популярність систем безпеки, в місті не так багато підприємств, що займаються їх продажем, установкою і подальшим сервісним обслуговуванням. Існуючі підприємства вважають за краще працювати з великими замовниками (в основному, юридичними особами) тому для невеликих офісів, квартир, дач пропонується даний охоронний пристрій.

Аналіз останніх досліджень і публікацій. Необхідно зазначити, що розробці подібних пристроїв приділялась і приділяється значна увага, вони проектуються та розробляються кожного разу все більш досконалішими та модифікованими.

Невирішені частини проблеми. Незважаючи на досить великий термін проектування подібних пристроїв ще й досі найбільшою проблемою є вибір контролера.

Метою даної роботи є розробка охоронного пристрою на мікроконтролері PIC16F84.

Основні результати дослідження. Результатом дослідження є спроектований та описаний охоронний пристрій.

ВСТУП

Людство вступило в 21-е століття. Технології в багатьох галузях розвиваються в геометричній прогресії. Особливо це стосується електроніки. Великий прорив відбувся у області передачі, обробки і перетворенні даних. Це досягнуто завдяки новим відкриттям і прогамованим мікросхемам. Застосування поверхневого монтажу дискретних елементів і «мініатюрної» друкарської плати з мікроконтролерами дозволило отримати прилад, який поміщається на «долоні». Все це в даний час широко використовується в областях охоронної сигналізації, контролю доступу, телефонії. Тепер, в порівнянні з приладами, які широко застосовувалися всього лише 25 років тому, що мають розмір значного «ящика», застосовуються малогабаритні пристрої. Можливості цих пристроїв збільшилися мінімум на порядок, а зовнішній вигляд їх став «ювелірним», який не зіпсує дизайн офісу або квартири.

РОЗВ'ЯЗАННЯ ЗАДАЧІ

Охоронний пристрій на мікроконтролері PIC16F84 (надалі пристрій) призначений для цілодобової охорони об'єктів різного призначення: офісів, дач, квартир, гаражів, сховищ і т.п. Він забезпечує ввімкнення сирени та світлового покажчика у випадку проникнення об'єкт, що охороняється, сторонніх осіб. До пристрою можна підключити декілька охоронних ліній з магнітоконтактними давачами, що працюють на розмикання та паралельно з ними ІЧ давачі руху.

Постановка на режим охорони та зняття з нього відбувається з використанням електронних ключів.

Запропонований пристрій може виконувати функції охоронної сигналізації або просто вмикати освітлення при русі людини в приміщенні і при відкриванні входних дверей. Його схема показана на рисунку 1. Основою служить мікроконтролер PIC16F84. У його програмі передбачено автоматичне відновлення нормальної роботи після випадкових збоїв.

Включення режиму охорони і зняття з нього робиться за допомогою електронних ключів-таблеток iButton DS1990. Ці дуже прості в експлуатації і надійні вироби не бояться вологи і агресивного зовнішнього середовища, не вимагають зміни або підзарядки елементів живлення, так як не містять їх. Код, записаний в таблетці, неможливо змінити. Він має 2800000000000000 варіантів, що практично виключає можливість добірки ключа.

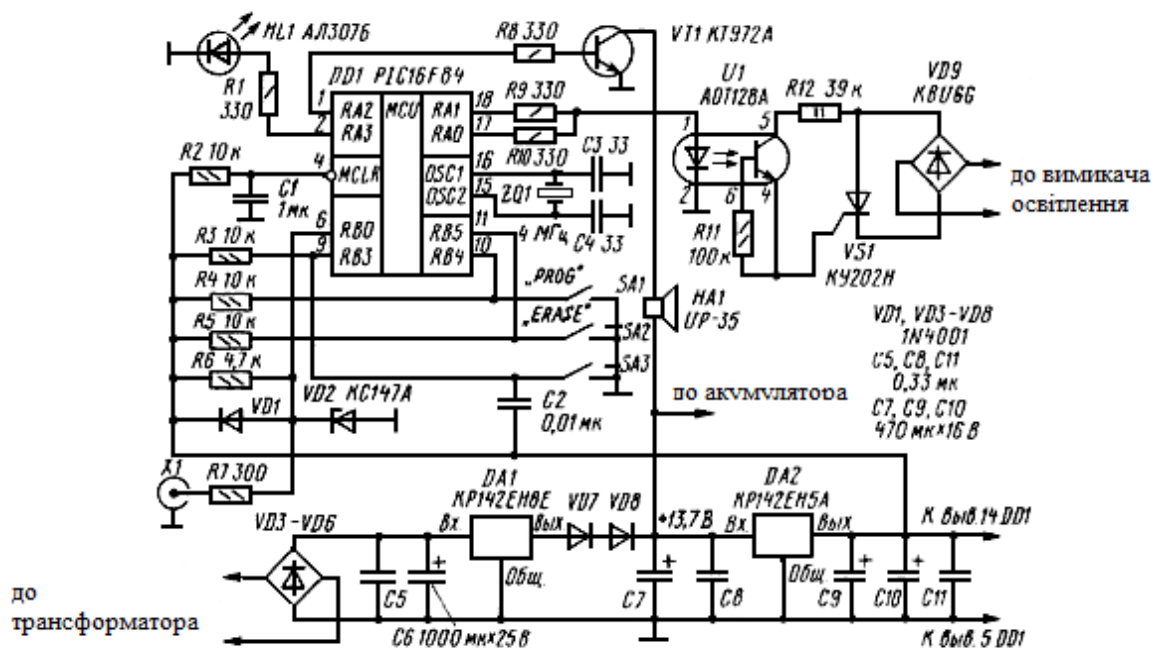


Рис. 1. Схема сигналізації

Контактний пристрій X1, якого потрібно торкнутися ключем, щоб включити або виключити охорону, розташовують у будь-якому зручному місці, наприклад, на косяку дверей. Однак сам охоронний прилад повинен бути недоступний стороннім. Проводку від нього до контактному пристрою виконують витю парою проводів довжиною не більше декількох метрів.

Світлодіод HL1 служить для контролю стану приладу. По ньому можна визначити, чи включений режим охорони, чи спрацювала сигналізація з моменту її включення.

Охоронний шлейф, умовно показаний на схемі як вимикач SA3, – послідовно з'єднані вихідні контакти ІЧ давача руху і встановленого на двері герконового давача СМК. Якщо давача руху немає, прилад буде реагувати тільки на відкривання дверей.

При включенні живлення пристрій починає роботу в режимі, в якому він при розмиканні шлейфу лише включає освітлення. Приблизно через хвилину світло буде автоматично вимкнене. Щоб перейти в режим охорони, потрібно торкнутися зареєстрованим ключем контактному пристрою X1. Прочитавши код ключа, мікроконтролер на Іс. забороняє повторне зчитування, запобігаючи непередбачуваній зміні режимів при занадто тривалому утриманні ключа в контактному пристрої.

Ввимкнення режиму охорони буде підтвержене коротким звуком сирени HA1 і спалахом світлодіода HL1. Але фактично охорона буде включена тільки після відновлення замкнутого стану шлейфу (якщо останній був розімкнутий). Це дає можливість «господареві» піти з приміщення і закрити за собою двері, не викликавши помилкової тривоги.

У режимі охорони світлодіод HL1 світиться, освітлення вимкнене. Порушення шлейфу призводить до включення на 3 хв. сирени HA1 і мигання світлодіода HL1. Потім мікроконтролер ще раз перевіряє стан шлейфу. Якщо він виявиться знову замкнутим, сирена буде вимкнена, але

мигання світлодіода продовжиться. Повторне торкання зареєстрованим ключем контактного пристрою зніме відключення режиму охорони, що буде підтверджено дворазовим короткочасним включенням сирени і згасанням світлодіода. А ось будь-яка спроба підібрати ключ не змінить режиму роботи, але викличе включення сирени на 1с.

Вузол живлення приладу складається з випрямляча на діодному мосту VD3 - VD6 (понижуючий трансформатор живлення на схемі не показаний) і двох інтегральних стабілізаторів DA1 і DA2. Діоди VD3 - VD8 можна замінювати будь-якими з напругою не менше 50В і струмом 1А.

Напруга 13,7 В призначена для живлення сирени HA1 і підзарядки акумуляторної батареї, що підтримує роботу пристрою при відключенні мережі. Якщо в резервному живленні немає необхідності, замість KP142EH8 можна встановити стабілізатор на 12В, а діоди VD7 і VD8 виключити.

Сирена HA1 – автомобільна UP-35, але можна застосувати і будь-які інші п'єзосирени, наприклад, використовувані в охоронних системах (ООПЗ-12, «Сопілка»). Акумуляторна батарея – кислотна, призначена для приладів охоронно-пожежної сигналізації.

Пристрій зібрано на друкованій платі з односторонньо фольгованого склотекстоліту. Для мікроконтролера DD1 краще всього передбачити панель.

Це дозволить, при необхідності, замінити або перепрограмувати мікроконтролер.

Щоб зареєструвати ключ, короткочасно замкніть контакти вимикача SA1 «PROG». Як тільки засвітиться світлодіод HL1, доторкніться ключем контактів X1. Успішну реєстрацію підтвердять згасання світлодіода і короткий звук сирени.

Через деякий час світлодіод знову засвітиться – контролер готовий до реєстрації наступного ключа. Всього їх може бути сім. Восьмий зітре дані про перші, дев'ятий – про другий і так далі по колу. Через 40с. після останньої реєстрації прилад автоматично повернеться в робочий режим.

Щоб скасувати реєстрацію всіх ключів, достатньо короткочасно замкнути вимикач SA2 «ERASE». Пам'ять буде очищена і включений описаний вище режим реєстрації. Ця процедура необхідна при першому включенні пристрою, а також у випадку втрати одного з ключів.

РОЗРОБКА СТРУКТУРНОЇ СХЕМИ ПРИСТРОЮ

Основним функціональним елементом пристрою є мікроконтролер.

Основні функції, здійснювані мікроконтролером:

- відлік часу (для даного пристрою час на вхід, на вихід і на звучання сирени);
- приймання команд, що надходять від перемикачів.

Проаналізувавши функції, виконувані мікроконтролером, до нього можна висунути такі вимоги:

- тактова частота мікроконтролера повинна бути достатньою для нормальної роботи всієї системи;
- вихідні струми й напруги на виводах мікроконтролера повинні бути достатніми для керування індикатором;
- загальна кількість виводів мікроконтролера повинна бути достатньою, для забезпечення керування всіма елементами, що входять до складу пристрою, робота яких керується мікроконтролером [1].

Також мікроконтролер повинен відповідати низці другорядних вимог:

- мікроконтролер повинен мати мале енергоспоживання;
- об'єм пам'яті програм повинен вміщувати програмний код;
- обсяг оперативної пам'яті має бути достатнім для коректної і швидкої роботи пристрою.

Враховуючи всі перераховані вимоги, в нашому пристрої можна використати мікроконтролер – це PIC16F84 [2].

Основною приладу є мікроконтролер, як ядро, де відбувається обробка вхідної інформації, а також індикація, на якій відображається інформація в зручному для оператора вигляді.

Живиться сигналізація як від загальної мережі, так і від акумулятора.

До приладу підключаються давачі, від яких на МК подається інформація, якщо якийсь з давачів спрацював, то контролер обробивши інформацію, подає сигнал на сирену та на вимикач освітлення, який вмикає його. Структурна схема приладу зображена на рисунку 2.

Охоронну сигналізацію можна розділити на такі структурні блоки, які й будуть складати структурну схему даного пристрою:

- мікроконтролер PIC16F84;
- давачі;
- зчитувач ключів;
- сирена, зовнішній маяк;
- блок живлення, акумулятор.

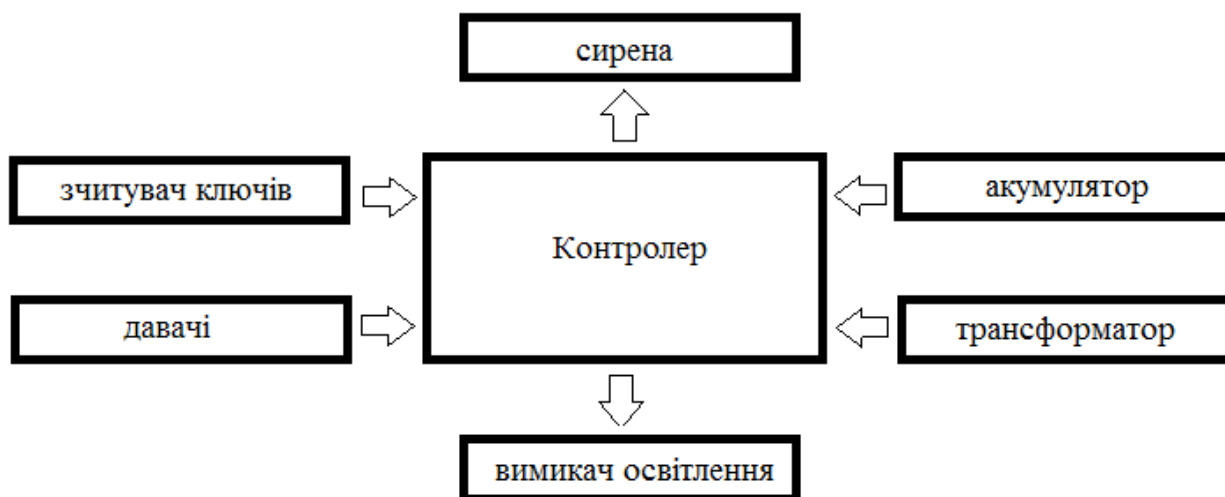


Рис. 2. Структурна схема пристрою

Давачі руху і герконові давачі встановлені в кімнаті і входних дверях приміщення відповідно, що охороняється. При порушенні цілісності одного або кількох шлейфів сигналізації (наприклад, розмикання герконового давача) приводить до вмикання низькочастотного звукового сигналу і активує освітлюваний пристрій.

Сирена (звуковий оповісник) НА1 забезпечує тривалу подачу голосного звукового сигналу (100...105 дБ). Зовнішній світлодіод сигналізує про режими роботи охоронного пристрою.

ВИБІР МАТЕРІАЛУ ДРУКОВАНОЇ ПЛАТИ

Як основу для виготовлення ДП, використовують шаруваті діелектрики, фольговані електролітичною міддю.

Вимоги до параметрів матеріалу основи ДП:

- висока термостійкість і мала вологопроникність;
- поверхневий опір при 40°C має бути не меншим 10^4 МОм;
- чистота міді 99,5%;
- шорсткість не гірше 0,4 мкм.

Таким параметрам відповідають гетинакс і склотекстоліт.

Гетинакс – шаруватий пластик на основі паперу й синтетичних смол. Сполучною речовиною найчастіше є феноло-формальдегідні смоли, рідше – меламіно-формальдегідні, епоксидно-феноло-аніліно-формальдегідні. Вміст смоли в гетинаксі складає 40–55%. Іноді гетинакс фольгують червоно-мідною електролітичною фольгою, облицьовують бавовняними, скляними або азбестовими тканинами, армують металевою сіткою. Залежно від призначення гетинакс випускають кількох марок.

Гетинакс має високу механічну міцність та хороші електроізоляційні властивості. Наведемо деякі його характеристики: щільність $1,25 \text{ г/см}^3$; теплостійкість по Мартенсу 150...160 °C; міцність при розтягуванні $70...100 \text{ МН/м}^2$ ($700...1000 \text{ кг/см}^2$), міцність при статичному вигині (по основі) $80...140 \text{ Мн/м}^2$ ($800...1400 \text{ кг/см}^2$); питома ударна в'язкість $1,3...1,5 \text{ кДж/м}^2$ ($13...15 \text{ кг/см}^2$); водопоглинання за 24 години $0,3-0,6 \text{ г/дм}^2$; питомий поверхневий електричний опір $10^{10}...10^{12} \text{ Ом}$; тангенс кута діелектричних втрат при 10^3 кГц $0,07-0,10$ [11].

Склотекстоліт – шаруватий пластик, що складається з шарів склотканини (наповнювач), просякнутих синтетичною смолою (зв'язувальна речовина). Склотканини, що використовуються

для виготовлення склотекстоліту, можуть бути одношаровими й багатшаровими, різними за видом плетива (наприклад, кордне, полотняне, сатинове) і складом волокон. При продукуванні склотекстоліту зазвичай використовують кілька шарів склотканин (головно одношарової).

Склотекстоліт застосовують як конструкційний матеріал для виготовлення листів і великогабаритних виробів складної конфігурації, а також як електроізоляційний матеріал в електро- і радіотехніці.

Недоліком цього матеріалу є його низька теплопровідність порівняно з металами, але він дешевий і виробництво ДП на його основі є економічно вигідним, тому зупиняємо вибір на ньому.

Вибираємо фольгований склотекстоліт.

Висновки. Аналізуючи вище викладене, можна констатувати, що на даний час важливою є питання безпеки та збереження матеріальних цінностей.

У статті описаний принцип роботи сигналізації, показана її схема, наведені основні деталі необхідні для повноцінної роботи..

1. Бирюков С.А. Цифровые устройства на МОП-интегральных микросхемах. – М.: Радио и связь, 1998.
2. Журнал «Беспроводные технологии» №1. – 2007 г.
3. А.Л. Ланцов, Л.Н. Зворыкин, И.Ф. Осипов. Цифровые устройства на комплементарных МДП интегральных микросхемах. – М.: Радио и связь, 1983. – 272 с.
4. Пухальский Г.И., Новосельцева Т.Я. Проектирование дискретных устройств на интегральных микросхемах: Справочник. – М.: Радио и связь, 2002. – с. 254.
5. Савельев А.Я., Овчинников В.А. Конструирование ЭВМ и систем. – М.: Высшая школа, – 1989. – 312 с.
6. У. Тилл и Дж. Лаксон. Интегральные схемы. Материалы – Приборы Изготовление, Мир, Москва, – 1985 г., 183 с.
7. Ушаков Н.Н. Технология производства СВМ. – М.: Высш. Шк., 1991. – 416 с.
8. Шевкопляс Б. В. Микропроцессорные структуры. Инженерные решения: Справочник. Дополнение первое. – М.: Радио и связь, 1993.
9. Яковлев, О.И. Распространение радиоволн / О.И. Яковлев, В.П. Якубов, В.П. Урядов, А.Г. Павельев. – СПб. Ленанд, – 2009. – 496 с.
10. http://www.atmel.com/dyn/resources/prod_documents/doc0368.pdf.
11. <http://www.eldigi.ru>.

УДК 614.8

О.О. Вісин

Луцький національний технічний університет

АНАЛІЗ СУЧАСНИХ КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ УПРАВЛІННЯ ОХОРОНОЮ ПРАЦІ

Вісин О.О. Аналіз сучасних комп'ютеризованих систем управління охороною праці. В статті наведено аналіз сучасних комп'ютеризованих систем для комплексного управління охороною праці. Детально розглянуто використання багатофункціональних інтегрованих систем комплексної безпеки.

Ключові слова: управління охороною праці, комп'ютеризовані системи, інтегровані системи комплексної безпеки, ризик, система менеджменту.

Висын Е.А. Анализ современных компьютеризированных систем управления охраной труда. В статье приведен анализ современных компьютеризированных систем для комплексного управления охраной труда. Подробно рассмотрено использование многофункциональных интегрированных систем комплексной безопасности.

Ключевые слова: управление охраной труда, компьютеризированные системы, интегрированные системы комплексной безопасности, риск, система менеджмента.

Visyn O.O. Analysis of modern computerized safety management system. This paper presents an analysis of modern computerized systems for integrated safety management. Considered in detail using integrated multi-function integrated security systems.

Keywords: safety management, computerized system, an integrated system of comprehensive security risk management system.

Постановка проблеми. В даний час охорона праці перебуває на етапі значних змін. Нові комп'ютеризовані моделі і принципи управління, на ряду з відомими формами і методами управління систем управління охороною праці, отримують широкого поширення.

Аналіз останніх досліджень і публікацій. Щодо останніх досліджень зазначеної проблематики, то більш ґрунтовно це питання було вперше висвітлено Ткачуком К.Н., Мольчаком Я.О., Каштановим С.Ф., Полукаровим О.І., Ткачуком К.К., Полукаровим Ю.О. та Стасюком В.М. [1]. Щодо комп'ютеризації служб охорони праці, то ці питання вперше були розглянуті Зацарним В.В., Сабарно Р.В. Гогіташвілі Г.Г., Карчевські Є.-Т., Лапіним В.М. [2, 8].

Виділення невирішених раніше частин загальної проблеми. Використання комп'ютеризованих систем управління охороною праці дасть можливість забезпечити максимальну оперативність та максимально можливу оптимізацію щодо прийняття рішень при використанні ризик орієнтованого підходу. Складові елементи – оцінка ризику, тобто аналіз виникнення і масштабів ризику в конкретній ситуації та управління ризиком, тобто аналіз ситуації і розробка рішень, які спрямовані на зведення ризику до прийняттого мінімуму.

Впровадження національної концепції розвитку в галузі управління охороною праці вимагає, в першу чергу, подальшого ефективного розвитку в галузі управління охороною праці, а саме, вирішення проблеми щодо створення сучасних комп'ютеризованих систем управління охороною праці з метою максимального підвищення ефективності СУОП як на державному, так і на регіональному, галузевому та виробничому рівнях. Це дасть змогу максимально зменшити вплив "людського фактору" на прийняття рішень в діючих системах управління охороною праці. Саме така система призведе до зменшення виробничих ризиків.

Формулювання мети дослідження. Завдання дослідження полягає у висвітленні основних проблем при управлінні охороною праці та їх вирішення шляхом запровадження сучасних комп'ютеризованих СУОП.

На зміну сформованих традиційних форм і методів управління охороною праці в Україні (і не тільки) приходить концепція менеджменту гігієни та безпеки праці, що базується на міжнародно визнаних стандартах і рекомендаціях. При чому цей процес відбувається одночасно не тільки в охороні праці, а й у суміжних сферах діяльності, таких як промислова та екологічна безпека. Враховуючи це, на деяких підприємствах України і Росії на базі локальних систем управління формуються, так звані синтезовані системи, що включають в себе одночасно питання забезпечення професійної, промислової і екологічної безпеки. В Україні - Інтегровані Системи Менеджменту Охорони Праці та Промислової Безпеки (ІСМОП і ПБ) [1-2].

Згідно з ДСТУ-П ОHSAS 18001 "Системи управління безпекою та гігієною праці", ОHSAS 18001 "Системи менеджменту охорони професійного здоров'я та безпеки праці", ІLO-OSH 2001 (МОП-СУОП 2001) "Керівництво з систем управління охороною праці" та "Рекомендаціям щодо побудови, впровадження та удосконалення системи управління охороною праці" [3-6], затверджених спеціально уповноваженим центральним органом виконавчої влади з нагляду за охороною праці, СУОП повинна передбачати:

- планування заходів з охорони праці;
- контроль виконання поточного та оперативних планів;
- можливість здійснення корегувальних та попереджувальних дій;
- можливість адаптації до обставин, що змінилися;
- можливість інтеграції в загальну систему управління.

Безумовно, що найбільш ефективно усі ці функції можуть бути реалізовані саме із застосуванням сучасних високоефективних комп'ютеризованих систем управління охороною праці (СУОП).

Одним з найбільш перспективних напрямків щодо подальшого розвитку таких комп'ютеризованих СУОП є все більш широке застосування в них багатофункціональних інтегрованих систем комплексної безпеки та їх складових, що підтверджується загальними тенденціями подальшого розвитку СУОП у найбільш розвинутих країнах світу.

Сучасний рівень розвитку інтегрованих систем комплексної безпеки, в тому числі і в Україні, наприклад, "Дунай-XXI" – "ВЕНБЕСТ", "Legos" – "Укрспецмонтаж 1. Системи безпеки", "Інспектор +" - ISS "Укр-Інвест-Консалтинг", "ForSec" – "Формула безпеки", "Кодос" – "Союзспецавтоматика", "FortNet" тощо, дозволяє вже зараз використовувати їх в СУОП як для підвищення рівня охоронної та пожежної безпеки промислових об'єктів, так і одночасно для покращення умов і безпеки праці на виробництві та вирішення питань гігієни праці, виробничої санітарії, зменшення рівня виробничого травматизму і професійних захворювань, а також для покращення соціальної захищеності працюючих [7].

В комп'ютеризованих СУОП можуть ефективно використовуватися майже усі складові багатофункціональних інтегрованих систем комплексної безпеки – це системи пожежної сигналізації та пожежегасіння, системи мовного оповіщення, системи охоронної сигналізації, відеонагляду (сctv) та відеореєстрації, контролю та управління доступом, системи управління зовнішнім обладнанням та пристроями безпеки тощо.

Такі складові багатофункціональних інтегрованих систем комплексної безпеки, як системи відеонагляду (сctv) та відеореєстрації, контролю та управління доступом вже зараз можуть значно підвищити ефективність роботи СУОП у сфері нагляду та контролю за станом охорони праці, у сфері виробничої безпеки, а також при проведенні внутрішнього і зовнішнього аудиту з питань охорони праці, при розслідуванні та профілактиці травматизму тощо.

Системи відеонагляду та відеореєстрації можуть бути використанні в СУОП для:

- контролю за станом охорони праці на підприємстві в реальному часі;
- контролю та аналізу стану охорони праці на підприємстві за певний період часу по відео матеріалам з бази даних;
- спостереження за технологічними процесами на декількох виробничих дільницях, як одночасно мультиплексний режим роботи монітору, так і роздільно-послідовне перемикання на моніторі відеосигналів з різних відеокамер;
- блокування роботи виробничого обладнання при попаданні в його небезпечну зону людини за рахунок використання програмного детектору руху (на одному кадрі може використовуватися одночасно більш ніж 10 запрограмованих зон з незалежними програмними детекторами руху в кожній з них);
- ведення журналу подій в небезпечних зонах обладнання при порушеннях безпеки праці з реєстрацією часу та осіб, які знаходяться в небезпечній зоні;
- запису в базу даних інтегрованої системи комплексної безпеки відео матеріалів стосовно нещасних випадків та аварій на виробництві із заданими параметрами (часом відео запису як до, так і після часу події аварії чи нещасного випадку; швидкістю запису; роздільною здатністю і т. ін.) і передачі цих відео матеріалів за необхідною адресою (ФССНВ; Держгірпромнагляд; галузеве Міністерство; і т. д.) з використанням телефонних чи мобільних ліній зв'язку, або Internet;
- контролю виконання вимог щодо пожежної безпеки на виробництві;
- контролю за використанням засобів індивідуального захисту.

Що стосується систем контролю та управління доступом, то вони можуть бути використанні в системі управління охороною, наприклад, для:

- контролю за відповідністю часу проведення регламентно-ремонтних робіт технологічним нормам;
- контролю за відповідністю кількості працюючих нормам безпеки при проведенні робіт з підвищеною небезпекою;

- допуску до роботи на технологічному обладнанні лише працівників, які мають відповідну кваліфікацію та пройшли всі необхідні види інструктажів;
- недопущення працівників, які не мають необхідної кваліфікації та відповідного наряду-допуску, до виконання робіт з підвищеною небезпекою;
- виключення можливості попадання працівників у виробничі приміщення з підвищеною вибухопожежонебезпекою, які не мають права там знаходитись;
- контролю режиму праці та відпочинку тощо.

Також в СУОП можливе більш ефективне використання і такої складової інтегрованих систем комплексної безпеки, як охоронні системи, які можуть не тільки виконувати свої основні охоронні функції, що теж дуже важливо, особливо на об'єктах з підвищеною небезпекою, а у разі застосування відповідних датчиків, одночасно виконувати і додаткові функції - наприклад, контролю відповідності санітарним нормам рівнів освітлення, шуму і вібрацій на робочих місцях, наявності в повітрі робочої зони шкідливих та небезпечних речовин або підвищеного рівня електромагнітного та іонізуючого випромінювання тощо.

Існуюча елементарна база багатofункціональних інтегрованих систем комплексної безпеки дозволяє все це з успіхом робити. Також, в разі необхідності, вона дозволяє здійснювати і управління виробничим обладнанням та пристроями безпеки, наприклад, системами освітлення виробничих приміщень, системами електропостачання, водопостачання, газопостачання, аварійними системами вентиляції, димовидалення, системами оповіщення тощо. Як бачимо, вже цих перелічених функцій більш ніж досить, щоб значно підвищити ефективність роботи систем управління охороною праці.

Крім того, на базі існуючих багатofункціональних інтегрованих систем комплексної безпеки можливо створення також і мобільних автоматизованих систем у сфері виробничої безпеки з елементами централізованого управління, і зробити це можливо без суттєвих додаткових витрат.

Слід підкреслити, що і економічні методи управління охороною праці також можуть бути реалізовані більш ефективно при застосуванні вищезгаданих складових інтегрованих систем комплексної безпеки.

Ще однією важливою особливістю сучасних інтегрованих систем комплексної безпеки є те, що модульний принцип їх побудови з використанням локальних і регіональних комп'ютерних мереж дозволяє і в подальшому практично нескінченно нарощувати їх як у часі, так і в просторі, а це, в свою чергу, дає можливість реалізовувати на їх базі вискоефективні комп'ютеризовані СУОП розосередженого типу. Причому це можливо не тільки у межах окремих промислових об'єктів, але і на регіональному і, навіть, на державному рівнях, що значно полегшує задачу створення єдиної вискоефективної загальнодержавної комп'ютеризованої СУОП.

Використання багатofункціональних інтегрованих систем комплексної безпеки у складі СУОП дозволяє максимально зменшити вплив "людського фактору" на прийняття рішень в діючих СУОП, а також значно підвищує ефективність роботи останніх, так як дає можливість не тільки фіксувати і вести відповідний облік та аналіз нещасних випадків, профзахворювань та аварій на виробництві і здійснювати УОП за рахунок електронної розсилки відповідних вказівок та рекомендацій суб'єктам СУОП, але і, що дуже важливо, забезпечує максимально високій рівень оперативності та динамізму щодо реагування СУОП на будь-які небезпечні та аварійні ситуації, а також дає можливість безпосередньо впливати на стан охорони праці на виробництві у реальному часі за рахунок постійно діючого моніторингу як на рівні окремого підприємства, установи чи організації, так і на регіональному та загальнодержавному рівнях.

У теперішній час на багатьох підприємствах служби охорони праці оснащені комп'ютерами, які дозволяють здійснювати низку завдань:

- складання графіків перевірки знань і медоглядів персоналу та контроль за їх здійсненням;
- контроль виконання заходів з охорони праці;
- контроль своєчасності технічних оглядів та оглядів посудин, що працюють під тиском, підйомних машин і механізмів, електрообладнання та ін.;
- підготовка звітів за формами 1-ПВ, 1-УБ, 7-ТНВ та ін.;
- оформлення актів Н-1 і наслідків травматизму;
- аналіз інформації про травматизм;
- нагадування про важливі планові заходи;
- надання довідок про порушення правил охорони праці;

- стеження за виконанням розпорядчих документів;
- розроблення різних документів та ін.

Київська фірма "Протек" створила відповідне програмне забезпечення для автоматизованого робочого місця "Охорона праці". На львівському заводі "Рема" була розроблена автоматизована система аналізу травматизму і захворювань (АСАТІЗ). Кожний місяць керівництво одержує дані про травматизм і захворювання на підприємстві, що дає можливість оперативно впливати й усувати причини тих явищ. Досить дієва є також автоматизована система контролю виконання документації (АСКД). За кожний місяць керівництво одержує: перелік невиконаних запланованих заходів з охорони праці із зазначенням прізвищ відповідальних; кількості прострочених днів; переліку працівників, які потребують переатестації з охорони праці; переліку осіб, відповідальних за безпеку об'єктів підвищеної небезпеки, що звільнилися, з нагадуванням про необхідність їх заміни. Усі відповідальні особи одержують повідомлення про закінчення терміну виконання ними запланованих заходів з охорони праці. За допомогою комп'ютерної техніки вирішується і багато інших питань охорони праці [8].

При встановленні на робочому місці АРМ необхідно провести попередню підготовку до роботи. Потрібно заповнити довідники, які використовує АРМ, відповідно до існуючих на підприємстві штатного розкладу та нормативних документів, які регламентують періодичність медоглядів, перевірок знань працівників, періодичність огляду обладнання тощо. У довідниках, що поділяються на групи, відтворюється певна інформація.

1. Структурно-адміністративні:
 - 1.1. Підприємство.
 - 1.2. Форма власності.
 - 1.3. Тип підприємства (промисловість, зв'язок, будівництво та ін.).
 - 1.4. Навчальні заклади з охорони праці, де планується проводити підготовку, атестацію та переатестацію працівників.
 - 1.5. Органи управління (міністерства, відомства).
2. Події, де зазначається:
 - 2.1. Характер події.
 - 2.2. Тип нещасного випадку (нешасний випадок, зі смертельним наслідком, груповий нещасний випадок, аварії першої категорії, аварії другої категорії).
 - 2.3. Вид події, що призвела до нещасного випадку (ДТП, ураження струмом і т. ін.).
 - 2.4. Наслідки нещасних випадків (інвалідність першої групи, інвалідність другої, третьої груп, переведення на легшу роботу, одужання, смерть).
 - 2.5. Стягнення (попередження, штраф, догана).
3. Причини:
 - 3.1. Причини нещасного випадку (технічні, організаційні, психофізіологічні, які, своєю чергою, поділяються на більш конкретні причини).
 - 3.2. Устаткування" яке стало причиною травми (готовий список з можливістю розширення).
4. Персонал:
 - 4.1. Стать особи (чол., жін.).
 - 4.2. Професія, посада (заноситься згідно з штатним розкладом).
 - 4.3. Відповідальність за безпечне ведення робіт (готовий список з можливістю розширення).
 - 4.4. Право на роботи з підвищеною небезпекою.
 - 4.5. Теми навчання та перевірки знань.
 - 4.6. Види медоглядів.
 - 4.7. Діагнози захворювань.
 - 4.8. Стан потерпілого щодо алкоголю.
5. Документи:
 - 5.1. Нормативні документи (правила, інструкції і т. ін.).
 - 5.2. Потоки документів (внутрішні, вхідні, вихідні).
 - 5.3. Види документів (положення, рішення, лист, акт, вказівка і т. ін.).
 - 5.4. Кореспонденти.
6. Виробництво:
 - 6.1. Тип виробництва.
 - 6.2. Підрозділи підприємства.
 - 6.8. Групи та підгрупи устаткування.
 - 6.4. Належність устаткування.

- 6.5. Види технічного обстеження.
- 6.6. Виконавці технічного обстеження.
- 6.7. Ознака реєстрації устаткування.
- 6.8. Види умов праці (чинники виробничого середовища і трудового процесу).
- 6.9. Засоби індивідуального захисту.
7. Норми:
 - 7.1. Нормативна періодичність технічних обстежень.
 - 7.2. Періодичність навчання.
 - 7.3. Періодичність медогляду.
 - 7.4. Норми видачі засобів індивідуального захисту.

Після первинного заповнення довідників робота спеціаліста з охорони праці значно спрощується, за умови постійного стеження за відповідністю довідників реальному стану речей.

У результаті заповнення довідників створюється повна картотека працюючих із зазначенням прізвища працівника, місця його роботи, підрозділу, в якому він працює, професії (посади), статі й дати народження та картотека устаткування. У будь-який момент можна переглянути по кожному працюючому перелік тем навчання, коли відбулася і коли планується наступна перевірка знань, перелік видів медогляду, дати проходження останнього й наступного медоглядів, коли і яких порушень вимог охорони праці припустився працюючий з підставою для обліку порушення, датою порушення і номером документа. Можна також переглянути або вивести на друк графік проведення навчання та перевірки знань по підприємству в цілому, по окремому підрозділу чи виду роботи або по конкретній особі. Такі ж графіки можна отримати по медоглядах та порушеннях вимог охорони праці [8].

Звіти 1-У Б, 7-ТНВ, 1-ПВ заповнюються автоматично, з можливістю корегування.

Створена картотека устаткування буде мати такі графіки: найменування обладнання, тип, марка, реєстраційний номер, обліковий номер, підприємство, підрозділ.

За допомогою фільтрів можна обрати устаткування за такими ознаками: назва ознаки реєстрації, підприємство, підрозділ, група устаткування, термін експлуатації (відпрацьований / не відпрацьований). Можна вивести графік технічних обстежень обладнання.

Найявні можливості дозволяють вести журнал реєстрації нещасних випадків і в будь-який момент отримати аналіз виробничого травматизму за певний визначений період за такими критеріями: кварталами, місяцями, роками, видом події та віком потерпілого, загальним стажем роботи та наслідками нещасного випадку, основною причиною нещасного випадку, перебування потерпілого в стані сп'яніння, підрозділами, стажем роботи, фахом, супутніми причинами нещасного випадку, а також за типами нещасних випадків, устаткуванням — джерелом нещасних випадків та за шкідливими чинниками.

У будь-який момент працюючому доступні довідки, які стосуються того блоку АРМ, з яким проводиться робота або довідка щодо АРМ загалом [8].

Використання АРМ дозволяє значно спростити роботу спеціаліста з охорони праці, передбачає більше часу на контроль за станом охорони праці безпосередньо на виробництві.

Було б доцільно на кожному підприємстві розробити систему автоматизованого аналізу нещасних випадків та планування заходів запобігання виробничого травматизму. Це дало б змогу здійснювати ефективне, оперативне та планомірне системне проектування планових та позапланових технічних і навчальних заходів з охорони праці для конкретного підрозділу з урахуванням статистичного аналізу та чинній нормативній базі діяльності кожного конкретного підрозділу.

Таким чином, використання складових багатофункціональних інтегрованих систем комплексної безпеки, як сучасні комп'ютеризовані системи відео нагляду та відео реєстрації і системи контролю та управління доступом, дійсно може дозволити значно підвищити ефективність роботи СУОП. Насамперед, це стосується можливості підвищення рівня організаційно-технічних заходів щодо безпеки праці на виробництві, а також вирішення питань гігієни праці, виробничої санітарії, зменшення рівня виробничого травматизму і професійних захворювань. Також досягається покращення соціальної захищеності працюючих, бо при проведенні розслідування нещасних випадків і аварій та при визначенні причин їх виникнення ця робота, безумовно, може бути виконана більш кваліфіковано при використанні матеріалів із відеоархівів систем відеонагляду та відеореєстрації, особливо враховуючи той факт, що у відеоархів може записуватися як відео, так і аудіоінформація з місця подій, і це дуже важливо для

підвищення ефективності роботи системи соціального страхування від нещасних випадків на виробництві, професійних захворювань та аварій.

Список використаних джерел

1. Управління охороною праці: Навч. посібник. / Ткачук К.Н., Мольчак Я.О., Каштанов С.Ф., Полукаров О.І., Ткачук К.К., Полукаров Ю.О., Стасюк В.М. - Луцьк: РВВ ЛНТУ, 2011. – 288 с.
2. Охорона праці та промислова безпека: Навчальний посібник. / Ткачук К.Н., Зацарний В.В., Сабарно Р.В. – К: Лібра, 2010. – 560 С.
3. ДСТУ-П ОHSAS 18001 «Системи управління безпекою та гігієною праці».
4. ОHSAS 18001 «Система менеджменту охорони здоров'я та безпеки персоналу».
5. Керівництво з систем управління охороною праці - МОП-СУОП 2001, ILO-OSH 2001 // Міжнародне бюро праці, м.Женева – 2001.
6. Рекомендації щодо побудови, впровадження та удосконалення системи управління охороною праці // Держгірпромнагляд - 7 лютого 2008 р.
7. Штацкий П.А. Системы комплексной безопасности объектов. Вопросы классификации по уровням и схемам интеграции / Штацкий П.А., Барышников А.Е. // Системы безопасности – 2003. – №2. – С. 26-30.
8. Управління охороною праці та ризиком за міжнародними стандартами: Навч. посіб. / Гогіташвілі Г.Г., Карчевські Є.-Т., Лапін В.М.– К: 2007. – 367 с.

УДК 004.9

О.К. Жигаревич

Луцький національний технічний університет

ДОСЛІДЖЕННЯ СКЛАДНИХ СИСТЕМ ЗАСОБАМИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ

Жигаревич О.К. Дослідження складних систем засобами імітаційного моделювання. В статті розглядаються питання становлення імітаційного моделювання в Україні. Досліджуються напрямки в області імітаційного моделювання. Описуються переваги використання імітаційного моделювання при побудові моделей.

Ключові слова: моделювання, аналітичні методи, імітаційне моделювання, моделювання системи, концептуальна модель.

Жигаревич О.К. Исследование сложных систем средствами имитационного моделирования. В статье рассматриваются вопросы становления имитационного моделирования в Украине. Исследуются направления в области имитационного моделирования. Описываются преимущества использования имитационного моделирования при построении моделей.

Ключевые слова: моделирование, аналитические методы, имитационное моделирование, моделирование системы, концептуальная модель.

Zhyharevych O.K. The study of complex systems by means of simulation. This paper deals with the establishment of simulation modeling in Ukraine. Researched areas in the field of simulation. Describe the benefits of using simulation modeling for building models.

Keywords: modeling, analytical methods, simulation, modeling systems, conceptual model.

Постановка проблеми. Управління в сучасному світі стає все більш важкою справою, оскільки організаційна структура суспільства постійно ускладнюється. Ця складність пояснюється характером взаємних відносин між різними елементами організацій і фізичними системами, з якими вони взаємодіють. Хоча ця складність існувала давно, лише тільки зараз ми розуміємо її значення. Очевидно те, що зміна однієї з характеристик системи може легко привести до змін, або навіть створити проблему в інших частинах системи.

У зв'язку з цим одним з найбільш важливих і корисних засобів аналізу складних процесів і систем стало імітаційне моделювання. Імітувати, згідно словнику Вебстера, означає "уявити, проаналізувати зміст явища, не проводячи експериментів на реальному об'єкті". [1]

"Імітаційне моделювання", як інструментарій дослідження складних систем, наведене в монографії відомого американського фахівця Р. Шеннона "Імітаційне моделювання систем - мистецтво і наука": "імітаційне моделювання є експериментальною і прикладною методологією, яка передбачає наступні етапи становлення:

- опис поведінки системи;
- побудова теорії і гіпотези, які можуть пояснити спостережувану поведінку;
- використання цієї теорії для прогнозу майбутньої поведінки і оцінки (в рамках обмежень, що накладаються деяким критерієм або сукупністю критеріїв) різних стратегій, що забезпечують функціонування даної системи".

З погляду комп'ютерної реалізації імітаційне моделювання - це комплексний метод дослідження складних систем на ЕОМ, що включає побудову концептуальних, математичних і програмних моделей, виконання широкого спектру цілеспрямованих імітаційних експериментів, обробку і інтерпретацію результатів цих експериментів.

Аналіз попередніх досліджень і публікацій. Методологічною основою для розвитку імітаційного моделювання на Україні є роботи Бусленко Н.П., Глушкова В.М., Моїсеєва Н.Н., Марчука Т.И., Коваленко І.Н. Слід відмітити, що в минулому склалися чотири школи в області імітаційного моделювання.

Московська – Моїсеєв Н.Н., Емельянов С.В., Калашников В.В., Андріанов А.Н., Бичков С.П., Хорошилов А.І., Черненко В.М., Лутков В.І., Бусленко В.Н., Рівес Н.Я., - на чолі з Н.П. Бусленко.

Ленінградська - Фомін Б.Ф., Аврамчук Е.Ф. - на чолі з член-кореспондентом Вавіловим А.А..

Новосибірська – Нечипоренко М.И., Чинін Г.Д., Окольнішніков В.В., очолювана академіком Марчуком Г.І.

Київська - Т.П. Мар'янович, В.В. Літвінов, И.Н. Коваленко, Л.А. Калініченко, В.В. Гусев, Жук К.Д., Яровицький Н.В., Бакаєв А.А., А.И. Нікітін на чолі з академіком СРСР Глушковим В.М.

Вказані школи не мали офіційного статусу, але їх керівники мали високий науковий авторитет і публікації основоположних робіт з питань комп'ютерного імітаційного моделювання, а їх учні очолили наукові колективи, які виконали і продовжують виконувати великий об'єм досліджень по наступних традиційних напрямках в області імітаційного моделювання:

- розвиток методології, методів і технологій моделювання;
- розробка засобів і систем моделювання на базі універсальних алгоритмічних мов моделювання;
- розробка пакетів моделювання широкого призначення;
- розробка проблемно-орієнтованих пакетів моделювання.

Представники Київської школи імітаційного моделювання на чолі з академіком Глушковим В.М. вели розробки і дослідження по всіх вказаних напрямках в області імітаційного моделювання. Інститут кібернетики є першим в області розробки методів і засобів імітаційного моделювання на основі мов моделювання високого рівня для вітчизняних ЕОМ всіх поколінь.

Природно, що творці засобів і систем імітаційного моделювання в Україні використовували в міру можливості зарубіжний досвід розробки таких систем.

Історія становлення і розвитку імітаційного моделювання в Україні пов'язана з відповідними етапами в світовій практиці в даній області.

Визначальними чинниками в історії імітаційного моделювання були генерації мов моделювання. Проте впродовж більш, ніж 30-річній історії змінювалися концепції, парадигми програмування, платформи, що вплинуло на специфіку відповідних етапів.

Відомі фахівці в області імітаційного моделювання Р. Ненсі і Ф.Ківіат в своїх роботах визначали декілька етапів в практиці розвитку імітаційного моделювання.[1]

Імітаційне моделювання є дуже широким і недостатньо чітко визначеним поняттям, що має дуже велике значення для людей, відповідальних за створення і функціонування практично будь-яких систем. Разом з тим, сама ідея імітаційного моделювання проста і в той же час складна. Вона дає можливість експериментувати з системами (що існують або запропонованими) в тих випадках, коли робити це на реальному об'єкті практично неможливо або недоцільно. Кожен сучасний керівник, якщо хоче отримати максимальну ефективність роботи свого підприємства, повинен періодично звертатися до методів імітаційного моделювання, тому що вони є найбільш універсальними методами дослідження систем.

При імітаційному моделюванні динамічні процеси - системи-оригіналу замінюються процесами, імітованими в абстрактній моделі, але з дотриманням основних правил (режимів, алгоритмів) функціонування оригіналу. В процесі імітації фіксуються певні події і стани або вимірюються вихідні дії, по яких обчислюються характеристики якості функціонування системи.

Імітаційне моделювання дозволяє розглядати процеси, що відбуваються в системі, практично на будь-якому рівні деталізації. При цьому в імітаційній моделі можна реалізувати практично будь-який алгоритм управлінської діяльності або поведінки системи. Крім того, моделі, які допускають дослідження аналітичними методами, також можуть бути проаналізовані імітаційними методами. Отже імітаційні методи моделювання в даний час є основними методами дослідження складних систем.

Переваги імітаційного моделювання.[2]

Всі імітаційні моделі є моделі типу так званого чорного ящика. Це означає, що вони забезпечують видачу вихідних параметрів системи, якщо на її взаємодіючі підсистеми поступають вхідні дії. Тому для отримання необхідної інформації або результатів слід здійснити "прогін" (реалізацію, "репетицію") моделей, а не "вирішувати" їх. Імітаційні моделі не здатні формувати своє власне рішення в тому вигляді, в якому це має місце в аналітичних моделях, а можуть лише служити як засіб для аналізу поведінки системи в умовах, які визначаються експериментатором. Цей недолік, насправді є головною перевагою імітаційного моделювання, адже доцільність застосування імітаційного моделювання стає очевидним за наявності будь-якої з наступних умов:

- не існує закінченої математичної постановки завдання, або ще не розроблені аналітичні методи рішення сформульованої математичної моделі;
- аналітичні методи є, але математичні процедури такі складні і трудомісткі, що імітаційне моделювання дає більш простий спосіб рішення задачі;
- крім оцінки певних параметрів, бажано здійснити на імітаційній моделі спостереження за процесом протягом деякого часу;
- імітаційне моделювання може виявитися єдиною можливістю внаслідок труднощів постановки експериментів і спостереження явищ в реальних умовах;
- для довготривалої дії систем або процесів може бути доцільно стиснення тимчасової шкали. Імітаційне моделювання дає можливість повністю контролювати час вивчення системи, оскільки явище може бути сповільнене або прискорене за бажанням.

У одній з своїх статей Маккіні пише: "Коли керівник досягає справжнього розуміння проблеми і починає вільно управляти своєю моделлю, він знаходить здатність бачити зміст своєї роботи з інших точок зору. Він хоче перевірити на моделі безліч альтернативних варіантів, щоб оцінити що відкрили йому нові можливості. Отже, він використовує модель для підвищення своєї майстерності управління, що дозволяє йому на новому рівні чітко встановити всі істотні наслідки змін, що вносяться до системи. Можливо, він міг би виконати це і на реальній системі, але через її складність це було б важко і практично нереально. Саме тому він звертається до моделі як до засобу оцінки своїх нових інтуїтивних припущень і висновків." [2]

Ідею імітаційного моделювання можуть використовувати, як керівники так і дослідники систем. Тому метод імітаційного моделювання в даний час прагнуть застосувати для вирішення практично кожного завдання, з яким доводиться зустрічатися на практиці. За результатами досліджень 1000 найбільших фірм США (їх перелік постійно приводиться в журналі Fortune) встановлено, що методами, які найчастіше використовуються при аналізі їх діяльності, є методи імітаційного моделювання (табл.1).

Таблиця 1. Методи, які найчастіше використовуються при аналізі діяльності підприємств

МЕТОДИ	Частота використання	Відсоток
Імітаційне моделювання	60	29
Лінійне програмування	43	21
Мережеві методи планування і управління	28	14
Теорія управління запасами	24	12
Нелінійне програмування	16	8
Динамічне програмування	8	4
Цілочисельне програмування	7	3
Теорія масового обслуговування	7	3
Інші	12	6

Дані табл.1 говорять про те, що не дивлячись на недостатню математичну точність, імітаційне моделювання є одним з найкращих кількісних методів, які використовують при вирішенні проблем управління.[3]

Вибір засобів моделювання. Програмні і технічні засоби моделювання вибираємо за наступними критеріями. Неодмінна умова при цьому - достатність і повнота засобів для реалізації концептуальної моделі. Серед інших критеріїв можна назвати доступність, простоту і легкість освоєння, швидкість і коректність створення програмної моделі. Якщо вибір технічних засобів не викликає особливих труднощів, то вибір програмних продуктів є складним.

В даний час відомо більше 500 мов моделювання. Така кількість мов частково обумовлена різноманітністю класів модельованих систем, цілей і методів моделювання. Проте бажання спростити і прискорити процес створення моделей приводить до реалізації ідеї автоматизації програмування імітаційних моделей. Створений ряд систем, відокремлює дослідника від програмування. Програма створюється автоматично по одній з формалізованих схем на підставі параметрів системи, зовнішніх дій і особливостей функціонування, що задаються дослідником. Це найбільш перспективний напрям розвитку засобів імітаційного моделювання. Успішність проведення робіт на перших трьох етапах моделювання створює умови для успішного виконання всіх подальших заходів.

Процес створення загальної схеми імітаційного моделювання.

Для моделювання необхідно створити модель і провести її дослідження. Перед створенням моделі потрібно конкретизувати мету моделювання. Після дослідження потрібно виконати обробку і аналіз результатів моделювання.

Процес створення моделей проходить декілька стадій, починається із вивчення (обстеження) реальної системи, її внутрішньої структури і змісту взаємозв'язків між її елементами, а також зовнішніх дій і закінчується розробкою моделі. У збільшеному плані імітаційного моделювання припускаємо наявність наступних етапів:

1. Розробка концептуальної моделі;
2. Підготовка початкових даних;
3. Вибір засобів моделювання;
4. Розробка програмної моделі;
5. Перевірка адекватності і коректування моделі;
6. Планування машинних експериментів;
7. Власне моделювання;
8. Аналіз результатів моделювання і ухвалення рішення.

Для однієї і тієї ж системи можна скласти безліч моделей. Вони можуть відрізнятися ступенем деталізації і обліку тих або інших особливостей і режимів функціонування, відображати певну сторону системи, орієнтуватися на дослідження певних її властивостей. Тому всі етапи імітаційного моделювання викладені і сформульовані у меті дослідження. Важливу роль відіграють перші три етапи. Концептуальна модель системи. Після визначення мети моделювання будується концептуальна модель досліджуваної системи.[4]

Концептуальна модель (змістовна модель) - це абстрактна модель, що визначає склад і структуру системи, властивості елементів і причинно-наслідкові зв'язки, властиві аналізованій системі, характерні для досягнення мети моделювання. У концептуальній моделі зазвичай в словесній формі приводяться відомості про природу і параметри (характеристиках) елементарних явищ досліджуваної системи, про вигляд і ступінь взаємодії між ними, про місце і значення кожного елементарного явища в загальному процесі функціонування системи.

Наступним кроком на шляху створення концептуальної моделі служить вибір рівня деталізації моделі (стратифікація). Модель системи представляється у вигляді сукупності частин (підсистем, елементів). В сукупність включаються всі частини, які забезпечують збереження цілісності системи, з одного боку, а з іншої - досягнення поставленої мети моделювання.

Надалі проводиться остаточна деталізація, локалізація (виділення системи з навколишнього середовища), структуризація (вказівка і загальний опис зв'язків між виділеними елементами системи), укрупнений опис динаміки функціонування системи і її можливих станів.

При створенні концептуальної моделі практично паралельно формується область початкових даних (інформаційний простір системи). На даному етапі виявляються кількісні характеристики (параметри) функціонування системи і її елементів, чисельні значення яких складуть початкові дані для моделювання.

Очевидно, що значна частина параметрів системи - це випадкові величини. Тому особливе значення при формуванні початкових даних мають вибір законів розподілу випадкових величин, апроксимація функцій і т.д.

Серед методів прикладного системного аналізу імітаційне моделювання є сучасним і ефективним інструментом дослідження складних систем, управління якими пов'язане з ухваленням рішень в умовах невизначеності. В порівнянні з іншими методами таке моделювання дозволяє розглядати велику кількість альтернатив, покращувати якість управлінських рішень і точніше прогнозувати їх наслідки.

Цими обставинами по суті і визначається та область людської діяльності, в якій імітаційне моделювання по праву займає гідне місце.

Література:

1. Томашевський В.М. Імітаційне моделювання систем і процесів. – К.: ІСДО, 350 с. – 2005.
2. McKenney J. L., Clinical Study of the Use of a Simulation Model, The Journal of Industrial Engineering, N1, Jan. 1987.
3. Jacobson I. Object-Oriented Software Engineering. – s.l.:ASM press., 2004. – 528с.
4. Mental Model Musings. <http://www.outsights.com/systems/welcome.htm>
5. "CyberBest is a fast growing privately held company that focuses on Software Development and Technical Staffing Services" <http://www.simulationinformation.com/>
6. http://www.tbm.tudelft.nl/webstaf/edwinv/SimulationSoftware/Review_Simple.htm

УДК 004.023

Крестьянполь Л.Ю., ас.

Луцький національний технічний університет

РОЗРОБКА КОМП'ЮТЕРНОЇ МОДЕЛІ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ ОПТИМІЗАЦІЇ СИСТЕМИ ЗАХИСТУ ПАКОВАНЬ АЛКОГОЛЬНОЇ ПРОДУКЦІЇ

Крестьянполь Л.Ю. Розробка комп'ютерної моделі для вирішення задачі оптимізації системи захисту пакування алкогольної продукції. У роботі показано принципи побудови системної моделі, яка описує систему захисту пакування алкогольної продукції, а також показано використання інформаційних технологій для побудови моделі пошуку оптимальної системи захисту алкогольної продукції. Модель включає загальний алгоритм оптимізації та комп'ютерну програму пошуку

Ключові слова: комп'ютерна модель, інформаційна технологія, система захисту, алгоритм оптимізації.

Крестьянполь Л.Ю. Разработка компьютерной модели для решения задачи оптимизации системы защиты пакования алкогольной продукции. В работе показано принципы построения системной модели, описывающей систему защиты упаковки алкогольной продукции, а также показано использование информационных технологий для построения модели поиска оптимальной системы защиты алкогольной продукции. Модель включает общий алгоритм оптимизации и компьютерную программу поиска.

Ключевые слова: компьютерная модель, информационная технология, система защиты, алгоритм оптимизации.

Krestyanpol L.Y. The development of computer model to solve optimization problem for protection packing of alcoholic beverages. This work shows the principles of the system model, which describes the system of protection packaging of alcoholic beverages, as well use of information technologies to constructing a model for the optimal protection of alcoholic beverages. The model includes optimization algorithm and computer program search

Keywords: computer model, information technology, system protection, optimization algorithm.

1. Постановка задачі

На сьогоднішній день на ринку алкогольної продукції спостерігається тенденція до збільшення попиту на алкогольні напої якості яких виробник певною мірою захищає. З метою забезпечення алкогольної продукції належним рівнем захисту, виробники вводять у пакування різноманітні елементи захисту. Сукупність усіх цих елементів захисту та їх зв'язків із елементами конструкції пакування утворює систему захисту пакування. Кожен виробник знаходить для себе найбільш доцільний відносно ціни та ефективності набір захисних елементів. В свою чергу використання комбінації різних засобів захисту, створило велику кількість можливих варіантів захисту, які різняться між собою кількістю та видами захисних елементів, однак є далекими від оптимальних [1].

Існування великої кількості елементів системи захисту породжує комбінаторний вибух кількості можливих варіантів системи захисту, що, в свою чергу, висуває проблему пошуку оптимального варіанту системи захисту.

2. Принцип побудови системи захисту пакування алкогольної продукції

Для вирішення цієї задачі нами застосовано системний підхід, суть якого полягає в тому, що загальний рівень захисту формується як система. Система захисту пакування в загальному випадку при її оцінюванні на основі системного підходу повинна включати опис множини захисних елементів та множини їх зв'язків із елементами конструкції пакування. Об'єднання окремих елементів захисту за допомогою певних зв'язків із елементами пакування та між собою утворює системний захист пакування. Кількість варіантів системи захисту N , визначається як кількість комбінацій захисних елементів з яких можливо скласти систему захисту.

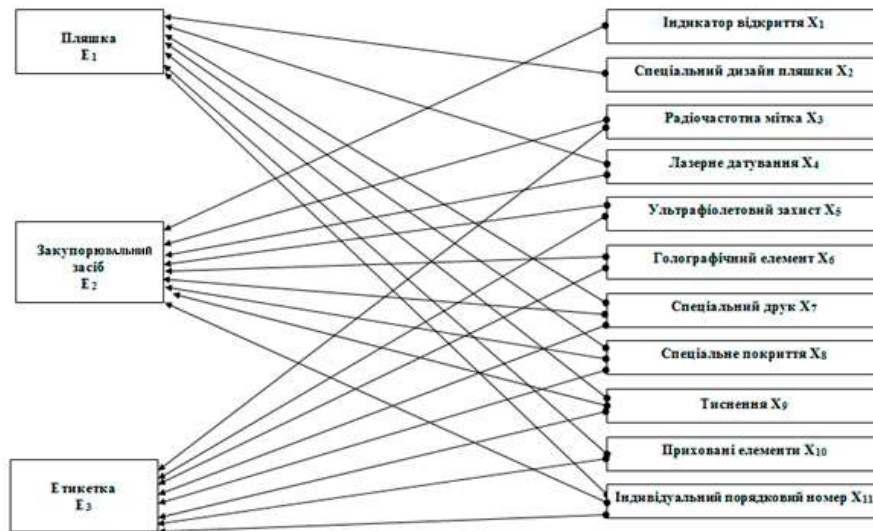


Рис. 1. Схема утворення системи захисту пакування за допомогою захисних елементів та їх зв'язків

Опишемо таку систему захисту пакування за допомогою графа зв'язків із трьома елементами конструкції пакування (пляшка – E_1 , закупорювальний засіб – E_2 , етикетка – E_3), як це показано на рис. 1.

Оскільки захисні елементи можуть розміщуватись на різних елементах пакування, то їх поєднання з конструктивними елементами пакування утворює значну кількість захисних систем.

За допомогою наступного виразу нами розрахована загальна кількість варіантів системи захисту:

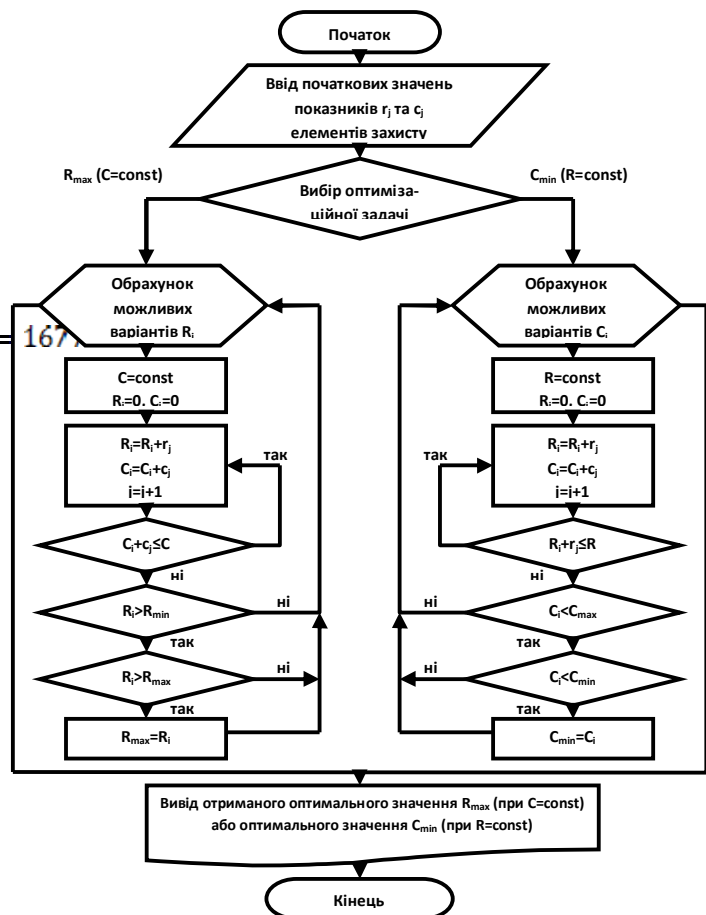
$$N = 24 + 1 + C_{24}^2 + C_{24}^3 + C_{24}^4 + \dots + C_{24}^{23} = 1677$$

де: N – кількість варіантів системи захисту,
 C – кількість сполучень.

Виходячи з отриманих результатів можна зробити висновок про існування значної кількості комбінацій захисних елементів з яких можливо скласти значну кількість систем захисту. Тому виникає питання пошуку оптимальної системи захисту серед усіх існуючих

3. Математична постановка задачі оптимізації системи захисту пакувань розробка алгоритмів розв'язання задачі оптимізації

Нами було використано метод перебирання для вирішення оптимізаційної задачі, а також вибрано параметри оптимізації серед яких ми вибрали вартість системи захисту C та



рівень захисту системи R . У такому випадку задача вирішується у двох варіантах постановки, а саме:

Задача 1 – знаходження найвищого рівня захисту R_{\max} при заданих економічних витратах на елементи захисту $C = \text{const}$.

Задача 2 – знаходження найдешевшого варіанту захисту C_{\min} при заданому рівні захисту $R = \text{const}$.

Інформаційні процедури оптимізації дозволяють здійснити пошук такого варіанту ступеню захисту R , який найбільшою мірою задовольняє умови задачі.

Формулювання постановки задачі оптимізації у разі використання одного з технічних параметрів R або одного з економічних параметрів C як критерію оптимізації можна представити у вигляді:

$$\begin{cases} F_1(X) = R \rightarrow \max; \\ C \leq C_{\text{зад}}. \end{cases}$$

або

$$\begin{cases} F_2(X) = C \rightarrow \min; \\ R \geq R_{\text{зад}}. \end{cases}$$

Тобто, в першому варіанті функцією мети є технічний параметр R , який максимізується, а економічний параметр

C задається у вигляді граничної умови, а **Рис. 2.** Алгоритм розв'язання у другій постановці навпаки - економічний параметр C – є функцією мети, а технічний параметр введений як гранична умова [1].

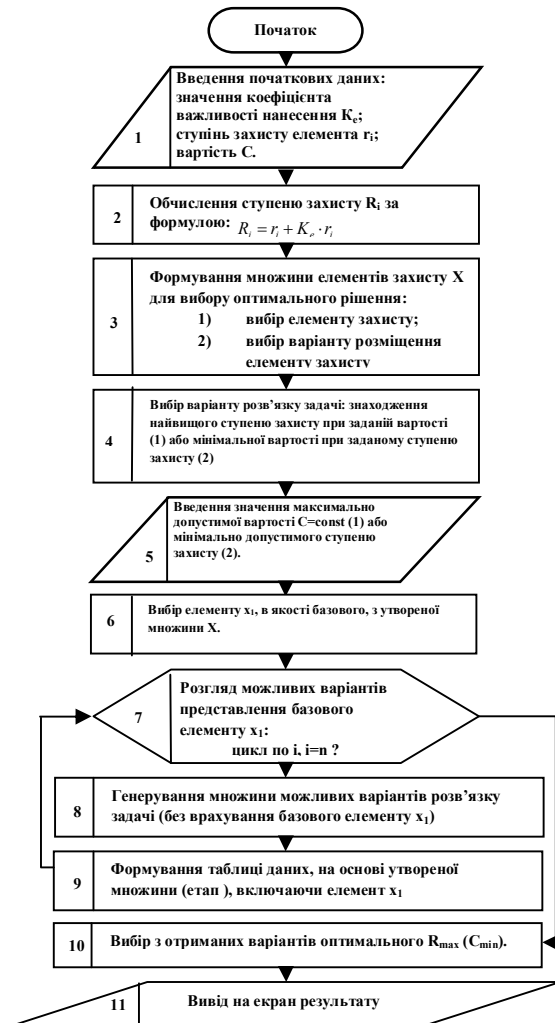
На основі поставленої задачі оптимізації нами було розроблено алгоритм для розв'язання оптимізаційної задачі (рис. 2).

На початковому етапі виконання алгоритму обчислення оптимальної системи захисту з таблиці беруться вихідні дані початкових значень r_i та c_i елементів захисту. На наступному етапі здійснюється вибір оптимізаційної задачі, яка може вирішуватися в одному із двох варіантів, а саме: пошуку найвищого рівня захисту R_{\max} , чи пошуку найдешевшого варіанту захисту C_{\min} . Наступним кроком є розв'язання вибраної задачі, яке здійснюється таким чином:

1. При сталому параметрі загальної вартості ($C = \text{const}$) визначається можливі варіанти ступеню захисту R_i , які не повинні бути меншими за мінімальне граничне значення R_{\min} .

2. При сталому параметрі ступеня захисту ($R = \text{const}$) визначаються можливі варіанти загальної вартості C_i , які не повинні перевищувати максимальне граничне значення вартості C_{\max} .

3. З множини усіх отриманих варіантів вибирається максимальне значення ступеня захисту (R_{\max}) при заданій вартості, або мінімальне значення вартості (C_{\min}) при заданому рівні захисту.



Наступним кроком є написання комп'ютерної програми основним завданням якої є пошук оптимальної системи захисту за заданими параметрами.

Дана програма дозволить виробнику алкогольної продукції обґрунтовано підбирати захисні елементи та створювати системи захисту підходящі відносно ціни, або ступеня захисту, та затратити якомога менше часу [2].

4. Оптимізаційна програма «ProТес»

Інформаційні процедури оптимізації дозволяють здійснити пошук такого варіанту ступіню захисту R , який

найбільшою мірою задовольняє умови задачі.

На рисунку 3 наведена функціональна блок-схема роботи програми «ProТес».

Рис. 3. Функціональна блок-схема роботи програми «ProТес»

Розглянемо детальніше роботу з програмою. Процедуру визначення оптимального варіанту системи захисту можна умовно поділити на 3 етапи:

1. Введення даних;
2. Вибір елементів захисту;
3. Визначення оптимального варіанту:
 - а) знаходження найвищого рівня захисту;
 - б) знаходження найдешевшого варіанту захисту.

На першому етапі формується вихідна таблиця з даними: коефіцієнт важливості розміщення (K_e), ступінь захисту елемента (r_i) та вартість (C).

В утворену таблицю вводяться назви елементів захисту та вибираються варіанти їх розміщення (рис. 4).

Залишилось заповнити дані в таблиці, що сформувалась на головному вікні. Показник ступеню захисту R_i розраховується автоматично, при введенні коефіцієнта важливості розміщення K_e і ступеню захисту окремого елемента r_i .

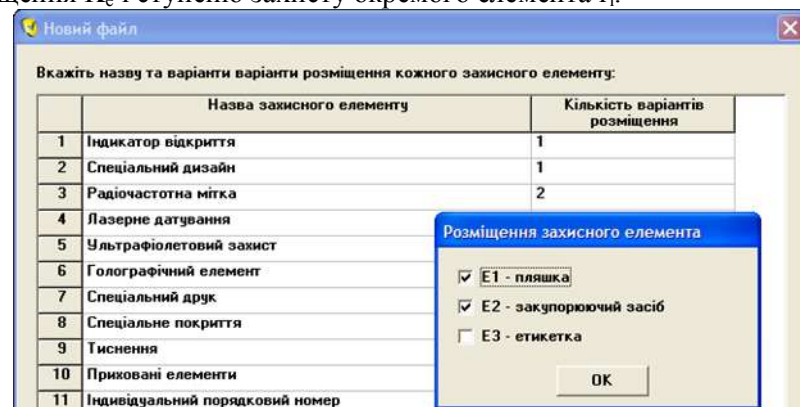


Рис. 4. Вибір варіантів розміщення захисного елемента

На наступному етапі вибираємо елементи захисту, які найбільше задовольняють поставленим вимогам і на основі яких далі буде визначатись оптимальний варіант системи захисту.

При додаванні елемента до списку вибраних, з'являється діалогове вікно, що дозволяє вказати варіант розміщення даного елемента.

Останній етап можна поділити на 3 умовні частини:

1. Вибір оптимізаційної задачі.
2. Генерування множини варіантів структури.
3. Оцінка варіантів та вибір оптимальної структури

Спочатку визначаємось з оптимізаційною задачею:

- пошук найвищого рівня захисту R_{max} при заданих економічних витратах на елементи захисту $C=const$;
- пошук найдешевшого варіанту захисту C_{min} при заданому рівні захисту $R=const$.

Натиснувши на кнопку «Пошук оптимального рішення», переходимо до генерування множини варіантів структури на основі вибраних елементів захисту та введених вихідних даних. Розглядаються всі можливі варіанти комбінування елементів захисту [3].

Визначившись із множиною всіх можливих варіантів системи захисту програма переходить до обрахунку показників рівня захисту R та економічних витрат C .

На основі обраної, на початку даного етапу, задачі переходимо до вибору оптимальної системи захисту:

1. Для першої задачі програма відкидає всі варіанти, що не задовольняють умові задачі – сумарна вартість не повинна перевищувати заданого раніше значення $C = \text{const}$.

З варіантів, що залишилися обирається структура, яка має найвищий рівень захисту R_{max} .

2. Для другої – всі варіанти, в яких рівень захисту нижчий заданого $R = \text{const}$. З варіантів, що залишилися обирається структура, яка має найменшу вартість.

Таким чином, програма дозволяє автоматизувати процес пошуку оптимального рішення під конкретні вимоги користувача.

Отриманий результат відображається в новому вікні (рис.5), де вказуються складові елементи системи захисту, варіант їх розміщення, а також рівень захисту та вартість.

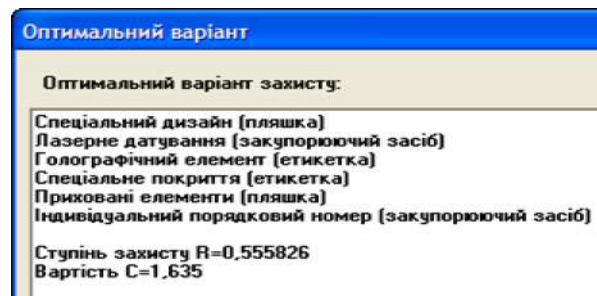


Рис. 5. Вивід на екран оптимального варіанту захисту

Дана програма дозволяє в стислі терміни підібрати варіант системи захисту пакування прийнятний для виробника відносно ступеню захисту чи вартості системи.

Висновки:

1. Розроблено метод побудови системної моделі захисту пакування алкогольної продукції, що описує структуру захисту як поєднання елементів і їх зв'язків.
2. Здійснено розрахунок загальної кількості варіантів систем захисту.
3. Вибрано критерії оптимізації та здійснено математичну постановку задачі оптимізації системи захисту пакувань алкогольної продукції від фальсифікації.
4. Розроблено алгоритми оптимізації та алгоритми роботи комп'ютерної програми для пошуку оптимального варіанту.

Список використаної літератури

1. Palchevskiy V., Krestianpol L. Synteza optymalizacyjna systemu ochrony napojów alkoholowych przed podrobaniem//Miesięcznik techniczno-ekonomiczny. Opakowanie NR 10. – Warszawa, 2013.-р.-46-50.- ISSN 0030-3348.
2. Пальчевський Б.О. Інформаційні технології проектування технологічного устаткування: Монографія /Б.О. Пальчевський. - Луцьк: Луцький НТУ, 2012.-572 с.
3. Крестьянполь Л.Ю. Інформаційні технології в проектуванні системи захисту алкогольної продукції / Л.Ю. Крестьянполь // Наукові нотатки. Міжвузівський збірник. Вип. № 42.- Луцьк: ЛНТУ, 2013. - С.134-140.
4. Крестьянполь Л.Ю. Аналіз засобів захисту товарів від фальсифікації/ Л.Ю. Крестьянполь, О.А. Крестьянполь // «Наукові нотатки». Міжвузівський збірник. Вип. 23.- Луцьк: ЛНТУ, 2008. - С.146-151.
5. Курицкий Б.Я. Оптимизация вокруг нас.-Л.: Машиностроение. Ленингр. отдние, 1989.-144с.: ил.
6. Майник Э. Алгоритмы оптимизации на сетях и графах. - М.: Мир, 1981. – 323 с.

УДК 665.511(047)

Г.В. Кулінченко, П.В. Леонт'єв, О.О. Ляпощенко
Сумський державний університет

ІДЕНТИФІКАЦІЯ МОДЕЛІ ПРОЦЕСУ НИЗЬКОТЕМПЕРАТУРНОЇ СЕПАРАЦІЇ ПРИРОДНОГО ГАЗУ

Кулінченко Г.В., Леонт'єв П.В., Ляпощенко О.О. Ідентифікація моделі процесу низькотемпературної сепарації природного газу. У статті аналізуються відомі моделі низькотемпературної сепарації газу. На основі сформованого критерію керування процесом ідентифікуються параметри моделі об'єкта. Проводиться візуалізація результатів моделювання в середовищах Flowvision і Matlab

Ключові слова: критерій керування, стала часу, математична модель, природний газ, сепарація
Форм. 7 Рис.7. Лит.10

Кулинченко Г.В., Леонтьев П.В., Ляпощенко А.А. Идентификация модели процесса низкотемпературной сепарации природного газа. В статье анализируются известные модели низкотемпературной сепарации газа. На основании сформированного критерия управления процессом идентифицируются параметры модели объекта. Проводится визуализация результатов моделирования в средах Flowvision и Matlab.

Ключевые слова: критерий управления, постоянная времени, математическая модель, природный газ, сепарация
Форм. 7 Рис.7. Лит.10

Kulinchenko G.V., Leontev P.V., Laposchenko A.A. Identification of the model low temperature separation process of natural gas. The paper analyzes the known models of low-temperature gas separation. Formed on the basis of criteria identified by the process control parameters of the model object. Visualization of simulation results carried out in environments Flowvision and Matlab.

Keywords: control criterion, the time constant, the mathematical model, natural gas, separation

Постановка проблеми. Розглядаючи завдання керування установкою комплексної переробки природного газу[1], можна констатувати, що найбільш відповідальними об'єктами автоматизації установки є процес теплообміну та низькотемпературної сепарації (НТС). В результаті НТС відбувається остаточна очистка газу від конденсату та вологи. Логікою розвитку енергетичних об'єктів - підготовки природного газу до транспортування є підвищення ефективності функціонування процесу НТС. Вирішення завдань підвищення ефективності зазвичай відбувається на базі автоматизованих систем керування, які дозволяють не тільки оптимізувати режими роботи системи, але й реалізувати завдання дистанційного керування об'єктом. Та розбудова будь якої системи керування передбачає наявність моделі об'єкту керування. Відомі на цей час моделі процесів НТС достатньо громіздкі та потребують багато програмно-апаратних ресурсів в разі їх використання. Тому такі моделі важко реалізувати в мікропроцесорному виконанні. Слід відзначити проплинність у часі параметрів процесу НТС, що викликає додаткові складнощі в проектуванні системи керування єдиного процесу підготовки природного газу до транспортування.

Таким чином актуальне завдання досліджень полягає в параметризації моделі процесу НТС, придатної для її мікропроцесорної реалізації.

Аналіз публікацій. Застосування технологічних моделюючих систем [2] для моделювання процесів промислової підготовки газових конденсатів (до складу якої входить НТС) дозволяє оцінювати оптимальні технологічні режими роботи і визначати кількість води і метанолу у всіх потоках промислової установки. Для цього використовуються модулі розрахунку процесів сепарації, краплеутворення, поділу рідин, дроселювання і теплообміну.

Зважаючи на технологічні проблеми процесу НТС, з метою забезпечення матеріального балансу рідкої і газової фаз в сепараторі, методом аналізу матеріальних потоків розроблена лінеаризована математична модель статички і динаміки процесу[3]. Завдання функціонування моделі полягає в підтримці матеріального балансу рідкої і газової фаз в сепараторі. У разі використання сучасних технологій модель НТС передбачає регулювання складу вихідної суміші газу при заданих температурі і тиску.

Подальшим розвитком процесів моделювання [2] є розробка інформаційно-моделюючої системи [4], яка адаптується до умов та особливостей роботи конкретної промислової установки. Розрахунки, на основі яких можуть бути обрані оптимальні технологічні режими установки, дозволяють оцінити показники товарної продукції, та виробити рекомендації відносно підвищення ефективності режимів роботи.

В процесі розвитку нових низькотемпературних газодинамічних технологій сепарації природного газу вдається застосувати математичні моделі [5], які дозволяють візуалізувати та оцінювати рух газокраплинних потоків в газосепараторах в режимі реального часу.

Та зважаючи на сформовані завдання досліджень, розглянуті підходи до моделювання процесів НТС не дають відповіді на питання про можливості розбудови каналів керування процесом. Аналіз сучасних класичних систем автоматизації газопромисловими об'єктами[6,7] дозволяє виробити стратегію розробки апаратно-програмного комплексу, який зможе забезпечити підвищити ефективність процесу НТС

Невирішені проблеми. Сучасний рівень автоматизації установок комплексної підготовки газу(УКПГ) обумовлює використання програмованих логічних контролерів(ПЛК), які на базі програмного комплексу, реалізованого на мовах МЕК 61131-3, (наприклад CoDeSys) забезпечують функціонування завдань керування польового рівня. ПЛК успішно забезпечують стабілізацію і програмне керування локальних регуляторів. Програмне забезпечення локальних систем регулювання складається відповідно до функціональних залежностей, які відпрацьовані спеціалістами на етапах моделювання об'єкту [2-4]. Проте, в разі зміни поточних некерованих параметрів процесу корекція його режимів здійснюється операторами верхнього рівня системи керування, в результаті чого на процес керування впливову складову накладає «людський фактор». Крім нестационарності параметрів цього процесу, певні труднощі в організації ефективного керування процесами НТС проявляються в нелінійному характері моделі. Останні складності традиційно долаються з допомогою лінеаризації моделі в вибраній точці[3], втім зрозуміло, що при зміні характеристик процесу, тобто відходу від цієї точки, необхідна корекція параметрів оптимального регулятора.

Тактика адаптації параметрів регулятора нелінійної моделі зазвичай залишалась невирішеною, оскільки вважалось, що автоматичний регулятор на базі нелінійної моделі виявляється складним для його реалізації. Дійсно, якщо формувати закон керування на основі многокритеріальної оптимізації, то труднощі реалізації виявляються ще на етапі ідентифікації моделі. Тому, приймаючи до уваги складність всього процесу НТС, на цьому етапі досліджень роботу розглянемо процес НТС без теплообмінника.

Якщо формувати критерій керування на базі одного з параметрів, то визначити ефективність роботи сепаратора достатньо складно. Це обумовлено суперечливим впливом параметрів процесу на показник вологості газу, що сепарується. Логічно створити критерій керування, спираючись на комплекс взаємопов'язаних параметрів, від яких залежить кількість відібраних продуктів. Враховуючи принципи визначення критерію керування та можливості впливу керуючих параметрів на досліджуваній об'єкт, можна констатувати, що математична модель процесу НТС повинна враховувати як вхідні параметри – витрати газу, тиск, температуру, так і змінні параметри (канали керування), які впливають на вихідні параметри - вологість газу і його температуру.

Труднощі в формалізації параметрів моделі сепарації пов'язані з характером течії газового потоку, оскільки на різних ділянках характер течії газу нерівномірно змінюється. Тому передумовою опису досліджуваного об'єкту є моделювання течії газу на різних ділянках установки сепарації.

Метою дослідження є вивчення характеристик течії газу на різних ділянках установки сепарації; визначення та ідентифікація параметрів моделі процесу сепарації; оцінка ступені впливу змінних параметрів на вихідні; отримання критерію керування, який дозволить синтезувати мікропроцесорний регулятор процесу НТС.

Основні результати дослідження. Для проведення аналізу характеристик течії газу в сепараторі скористайтесь спрощеною схемою процесу НТС, яка представлена на Рис.1. Вхідним потоком сепараторної установки є газова суміш, що підлягає очищенню, а вихідні потоки складаються з очищеного газу, конденсату та сконденсованої води.

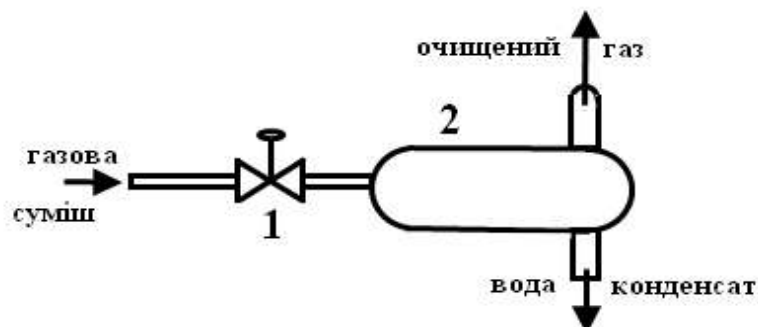


Рис. 1 Схеми процесу НТС

1- Дросель; 2-Сепаратор

Процес НТС відбувається завдяки ефекту дроселювання (ефект Джоуля - Томсона). В залежності від ступеня відкриття дросельного штуцера змінюється його гідравлічний опір, який і визначає глибину дросельного ефекту.

Опис процесів течії природного газу формується на базі системи рівнянь – нерозривності, руху та енергії:

$$\frac{\partial \rho}{\partial t} + \nabla(\rho V) = 0 \quad (1)$$

$$\frac{\partial V}{\partial t} + (V * \nabla) * V + \frac{1}{\rho} \nabla p = 0 \quad (2)$$

$$\left(\frac{\partial}{\partial t} + V * \nabla \right) * \left(c_v T + \frac{|V|^2}{2} \right) + \frac{1}{\rho} \nabla(pV) = 0 \quad (3)$$

де V - швидкість течії газу; ρ - питома щільність газу, p - тиск газу; c_v - теплоємність при постійній щільності; T – температура

Виходячи з системи рівнянь (1)...(3), для аналізу розподілу поля течій газу в процесі НТС можна скористатись програмним середовищем *FlowVision* [8], яка дозволяє чисельно розв'язати наведені рівняння.

Необхідною умовою моделювання є отримання та імпортування геометрії сепаратора та дроселя з допомогою системи автоматизованого проектування (САПР) *Autodesk Inventor*. В результаті створюється розрахункова область, в якій визначені рівняння (1)...(3) та межі об'єму, де сформовані граничні умови.

Достатньою умовою моделювання є створення початкової розрахункової сітки. В нашому випадку вибрана сітка, яка утворює 15x15x15 (3375) точок. Для точних розрахунків їх кількість збільшується, але при цьому значно зростає час розрахунків.

В моделі враховано, що природний газ має властивість повністю стискуватись. Початкові значення параметрів задаються для всієї розрахункової області: температура, концентрація, тиск та швидкість.

У постпроцесорі, що входить до системи моделювання, налаштовуються і задаються параметри візуалізації отриманих результатів, які стосуються необхідних прошарків моделі.

Необхідні початкові дані моделювання отримаємо з умови забезпечення необхідної продуктивності сепарації газу $Q=19,5\text{ м}^3/\text{с}$ при значенні вхідного тиску $p_{\text{вх}}=9,0\text{ МПа}$:

$$Q = \mu_d S \sqrt{\frac{2 * \Delta p}{\rho}} \quad (4)$$

де μ_d - коефіцієнт витрат дроселя; S - площа прохідного отвору дроселя;

Δp – перепад тиску на дроселі.

Скориставшись довідковими даними про конструктивні параметри дроселя (S), $\mu_d=0,7$, отримуємо перепад тиску Δp для двох різних значень прохідного отвору дроселя: $\Delta p_a=2,0\text{ МПа}$ і $\Delta p_b=4,0\text{ МПа}$.

Додаткові параметри моделювання - швидкість всередині розрахункового простору $U=38\text{ м/с}$ і початкова температура газу $T=283\text{ К}$, дають змогу отримати розподіл поля тисків процесу НТС.

На Рис.2 представлені розподіл поля тисків в трубі з дроселем при $\Delta p_a=2,0\text{ МПа}$ (Рис.2а) і $\Delta p_b=4,0\text{ МПа}$ (Рис.2б)

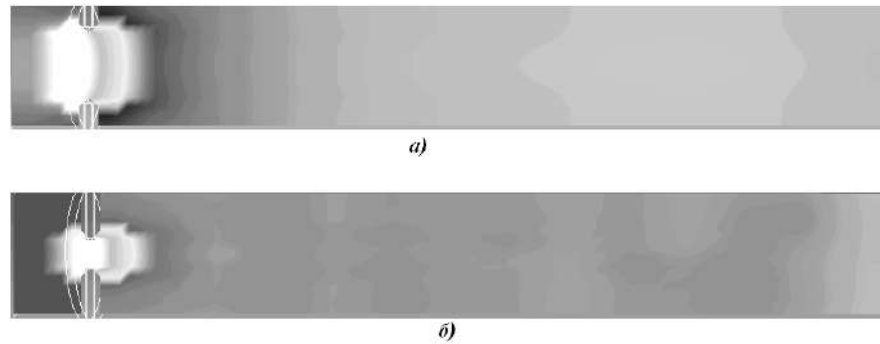


Рис.2. Розподіл поля тиску в трубі з дроселем

На Рис.3 відображено розподіл поля тисків в сепараторі при входному тиску $p_a=7,0$ МПа (Рис. 3а) і $p_a=5,0$ МПа (Рис. 3б) відповідно.

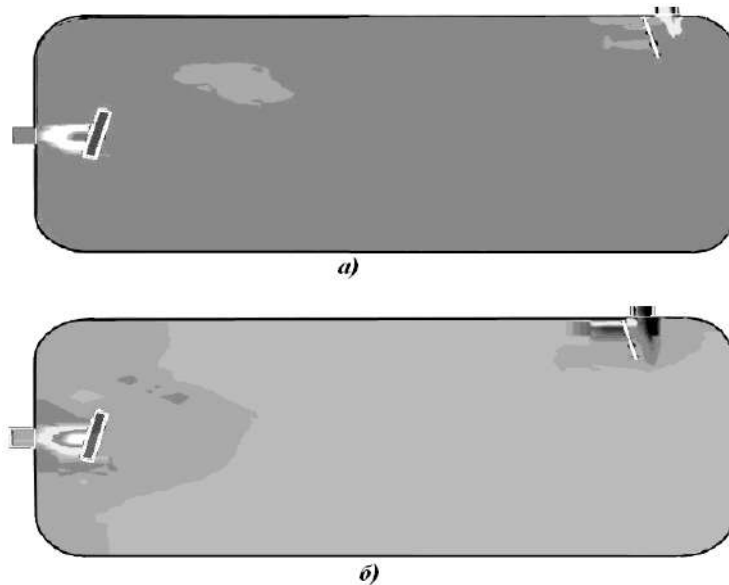


Рис.3. Розподіл поля тиску в низькотемпературному сепараторі

Взагалі практичний інтерес представляє розподіл поля «дросель-сепаратор», оскільки в сенсі керування об'єктом необхідно оцінювати вплив зміни керуючого параметру на поле течії газу. Проте процес розрахунків такою моделлю займає багато часу та ускладнює візуалізацію результатів. Тому модель процесу НТС була поділяється на функціональні частини, при цьому потік газу з труби приймається за вхідний потік для сепаратора.

Аналіз розподілу течії газу процесу НТС показує, що основні зміни процесу сепарації відбувається в дросельному пристрої, а поле тисків сепаратора характеризується достатньою рівномірністю.

Не меншої уваги з точки зору керування процесом заслуговує розподіл поля температур процесу НТС, оскільки місткість вологи в природному газі залежить від співвідношення значень його температури та температури точки роси (ТТР).

Моделювання температурного поля процесу дроселювання газу в середовищі *FlowVision* дозволяє отримати розподіл температур, наведений на Рис. 4.



Рис.4. Розподіл поля температур в трубі з дроселем

Характер розподілу поля температур при дроселюванні газу, що втікає з труби в сепаратор, дає підставу застосувати оцінку усередненої температури сепарації t_c [9], оскільки в подальшому температура суттєво не змінюється:

$$t_c = \frac{\sum_i G_i t_i}{G_c}$$

де G_i – витрати слою перетину; t_i – локальна температура слою;
 G_c – витрати через прохідний перетин

Таким чином результати оцінки вологомісткості газу ψ можна представити сукупністю двох параметрів – тиску і температури T °С [10]:

$$\psi(p, T) = \left[\frac{749}{p} + \exp(0.00007 T^2 - 0.0195 T) \right] * p_{np},$$

де $p_{np} = 10^{-6} * \exp(0.0735 T - 0.00027 T^2)$ пружність водяної пари в насиченому стані.

Якщо в практиці керування процесом НТС використовується ТТР, то вологомісткість газу можна розраховувати по цим же співвідношенням.

Повертаючись до формулювання критерію керування об'єктом, слід підкреслити, що розбудова його моделі повинна враховувати не тільки оцінки ефективності сепарації, а і варіанти апаратно-програмної реалізації.

Протиріччя оцінки ефективності сепарації обумовлені тим, що зменшення вологості газу пов'язано із зменшенням швидкості його течії, відповідно і з продуктивністю установки сепарації. Це обумовлює формування мінімаксного, комплексного критерію керування, який відслідковує мінімальну місткість вологи та конденсату на вході в магістраль при стабільних витратах газу Q , що транспортується:

$$\min_{\psi} \max_Q \{ \Phi(\psi_0, T, \Delta p, V) \} \quad (5)$$

де ψ_0 – початкова вологість газу.

Відповідно до сформованого критерію (5), розглянемо структуру моделі керованого об'єкту, до найважливіших параметрів якої відносяться динамічні характеристики, які входять до рівняння передатної функції по Лапласу $W(s)$.

Для ідентифікації моделі процесу НТС скористайтесь рівнянням Джоуля – Томсона, яке встановлює зв'язок між температурою та тиском газу:

$$T_c = T_{BX} - \chi (p_{BX} - p_c),$$

де T_{BX} , T_c – температура до і після дроселя; p_{BX} , p_c – тиск до і після дроселя; χ – коефіцієнт Джоуля – Томсона.

Взагалі коефіцієнт Джоуля-Томсона для природного газу змінюється від 2 до 4°К/МПа, в залежності від складу газу, падіння тиску та початкової температури газу. З достатньою для моделювання точністю можна прийняти $\chi=3$.

В турбулентному режимі зміна тиску газу зв'язана з витратою газу нелінійним рівнянням (4), тому при переході до відповідної передатної функції отримують трансцендентні рівняння, які складно ефективно апроксимувати, та майже неможливо синтезувати необхідний регулятор.

Досвід показує, що входом з цієї ситуації є апроксимація трансцендентного рівняння передатною функцією аперіодичної ланки першого порядку з запізненням. В результаті передатна функція дроселя після переходу до прирощень має вигляд:

$$W_d(s) = \frac{K}{\tau_d s + 1}$$

де $K = \left(\frac{dp}{dQ} \right)_{Q_0}$ $\tau_d = \frac{V}{RT \alpha}$

s - оператор Лапласа; V - об'єм дроселю; R – стала газова константа; T – температура газу;
 α - провідність турбулентного дроселю.

З достатньою точністю передатну функцію сепаратора можна представити ланкою з запізненням[11]:

$$W_c(s) = \exp(-\tau_c s)$$

де $\tau_c = V_c / Q$ - стала часу затримки ; V_c - об'єм сепаратора.

Коефіцієнти та параметри передатних функцій змінюються в часі та залежать від витрат газу і інших зовнішніх факторів. Ці обставини необхідно враховувати в процесі синтезу системи керування установкою підготовки газу до транспортування та розглядати її як об'єкт керування, що описується рівняннями із змінними коефіцієнтами.

Для візуалізації перехідної характеристики процесу НТС використано середовище MatLab Simulink. Схема моделювання представлена на Рис.5.

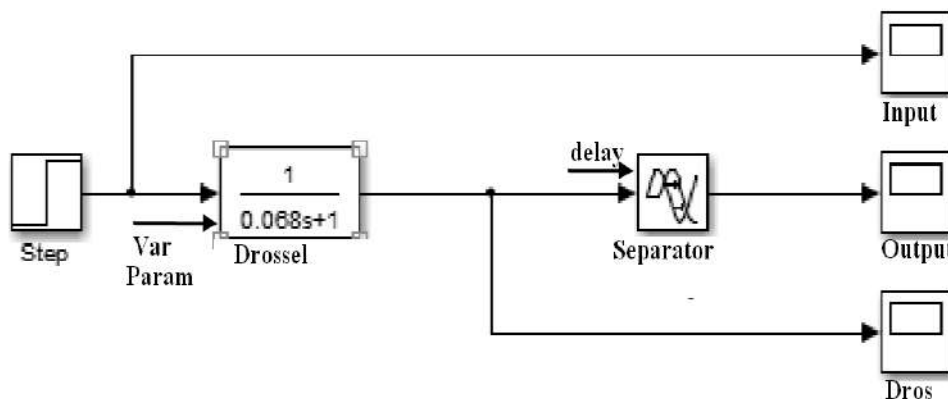


Рис.5. Схема оцінки перехідної характеристики процесу НТС

Для подальших досліджень в схемі передбачена імітація зміни параметрів в часі, що дозволить досліджувати ступінь впливу завод на якість процесу керування.

Висновки. В результаті аналізу процесів НТС проведено ідентифікацію параметрів моделі. Візуалізація процесів дроселювання та сепарації дозволила отримати передатні функції ланок процесу.

Отримана передатна функція представляється як об'єкт з затримкою.

Аналіз вигляду моделі процесу НТС показує, що розроблена модель може бути основою для синтезу регулятора процесу

1. Кулінченко Г. В. Завдання оптимального керування установкою комплексної переробки природного газу/ Г. В. Кулінченко, П. В. Леонт'єв//XII Міжнародна науково - технічна конференція «Фізичні процеси та поля технічних і біологічних об'єктів».- Кременчук: КрНУ.-2013.-С.23-24
2. Кравцов А.В. Математическое моделирование процессов отделения воды и метанола при промышленной подготовке газового конденсата/А.В. Кравцов, Н.В. Ушева, О.Е. Мойзес, Е.А. Кузьменко//Известия Томского политехнического университета.-2007.- Т. 311, № 3. С.60-63
3. Горбійчук М.І. Математична модель процесу низькотемпературної сепарації газу/ М.І. Горбійчук, Н.Л.Кулинин// Науковий вісник національного технічного університету нафти і газу.- Івано-Франківськ.-2003.-№1(13).-С.88-92.
4. Кравцов А.В. Информационно-моделирующая система процессов промышленной подготовки газа и газового конденсата/ А.В. Кравцов, Н.В. Ушева, О.Е. Мойзес, Е.А. Кузьменко, В.И. Рейзлин, А.А. Гавриков// Известия Томского политехнического университета.-2011.- Т. 318, № 5. С.132-137
5. Роммахи М. Разработка физической модели движения газоконденсатных потоков в рабочей полости ИФ – сепараторов /Мустафа Аль Роммахи, А.В.Логвин, О.О. Ляпощенко// Хімічна промисловість України. -2012.- №6.- С.18-21.
6. Масагутов Р. АСУ ТП установки подготовки газа с расширенной функциональностью системы ПАЗ./ Р. Масагутов// Современные технологии автоматизации.-2012.-№2.-С.20-25.
7. Камский П. Автоматизация установки комплексной подготовки газа на базе отказоустойчивой системы S7-400Н/ П. Камский// Современные технологии автоматизации.-2012.-№2.-С.30-37.
8. Аксенов А.А. Сертификация системы моделирования движения жидкости и газа FlowVision/ А.А.Аксенов, В.В.Шмелев, М.Л.Смирнова, В.В.Банкрутенко, И.В.Нетронин, А.В.Будников, С.А.Рогожкин//САПР и графика. Инструменты АРМ-2006.- №4.- С.80-85.
9. Ахметов Ю. М. Исследование влияния тормозного устройства на структуру потока и параметры изотермического вихрового регулятора давления./ Ю. М. Ахметов, А. Ю. Пархимович, А. В. Сви-стунув, А. А. Соловьев, В. А. Целищев, А. А. Чиндина // Вестник УГАТУ.-2011.-Т. 15, №4 (44). С. 149–153.
10. Гухман Л.М. Подготовка газа северных газовых месторождений к дальнему транспорту./ Л.М Гухман //Л.: Недра.-1980.-161с.
11. Тараненко Б.Ф. Автоматическое управление газопромысловыми объектами/ Б.Ф. Тараненко //М.: Недра.-1976.-217с.

УДК 629.113(071):004.01:004.04

А.Н. Полетайкин, О.А. Гузь

Донецкая академия автомобильного транспорта, г. Донецк

А.К. Каганюк

Луцкий государственный технический университет, г. Луцк

ИНТЕЛЛЕКТУАЛЬНЫЕ СРЕДСТВА И ТЕХНОЛОГИИ ИНФРАСТРУКТУРЫ ГОРОДСКОГО ДОРОЖНОГО ДВИЖЕНИЯ

Полетайкин А.Н., Гузь О.А., Каганюк А.К. Интеллектуальные средства и технологии инфраструктуры городского дорожного движения. В статье выполнен обзор технологий автоматизированного управления дорожным движением в крупных городах Украины. Определены приоритетные направления исследований для создания современных и перспективных технологий в данной отрасли. Рассматриваются инфраструктурные средства мониторинга и интеллектуального управления дорожным движением на основе программируемого логического контроллера с применением аппарата нечеткой логики.

Ключевые слова: обеспечение безопасности дорожного движения, инфраструктура дорожного движения, облачные технологии, радиочастотная идентификация транспорта, интеллектуальный дорожный контроллер, модель нечеткого управления.

Форм. 0. Табл. 0. Рис. 6. Лит. 11.

Полетайкін О.М., Гузь О.О., Каганюк О.К. Інтелектуальні засоби й технології інфраструктури міського дорожнього руху. У статті виконано огляд технологій автоматизованого управління дорожнім рухом у великих містах України. Визначені пріоритетні напрямки досліджень для створення сучасних і перспективних технологій у даній галузі. Розглядаються інфраструктурні засоби моніторингу й інтелектуального управління дорожнім рухом на основі програмувального логічного контролера із застосуванням апарата нечіткої логіки.

Ключові слова: забезпечення безпеки дорожнього руху, інфраструктура дорожнього руху, хмарні технології, радіочастотна ідентифікація транспорту, інтелектуальний дорожній контролер, модель нечіткого управління.

Форм. 0. Табл. 0. Рис. 6. Літ. 11.

Polietaikin A.N., Guz O.A., Kaganjuk A.K. Intelligence facilities and technologies of town road motion infrastructure. In this article review of automatic management road motion technologies is executed in large cities of Ukraine. Priority directions of studies are determined for making modern and perspective technologies in givened branch. They are considered the facility of monitoring and intelligence management road motion on base programmed logical controller with using the device of fuzzy logic.

The Keywords: provision to safety of road motion, infrastructure of road motion, cloud technologies, radiofrequency identification of transport, intelligence road controller, fuzzy control model.

Form. 0. Tabl. 0. Fig. 6. Src. 11.

Постановка проблемы. Актуальность данного исследования обусловлена чрезвычайной серьезностью проблемы обеспечения безопасности дорожного движения в крупных городах, где по всему миру ежегодно более одного миллиона человек погибает в ДТП, и более чем 50 миллионов получают травмы разной степени тяжести. По оценкам ВОЗ [11] международные денежные убытки из-за связанного с дорожным движением травматизма и материального ущерба достигают €500 миллионов в год. Почти 700 000 смертельных случаев ежегодно происходят в результате загрязнения воздуха, обусловленного движением транспорта. Без принятия эффективных контрмер, эти числа, согласно мировому отчету ВОЗ «О профилактике травм, полученных при ДТП», возрастут на 65 процентов в течение следующих 20 лет. Эту проблему усугубляет неуклонно возрастающая автомобилизация и технологическая отсталость дорожно-транспортных сетей. В целом нынешняя ситуация в области безопасности дорожного движения представляет собой кризис, выход из которого возможен только при комплексном подходе к решению указанной проблемы обеспечения БДД.

Мероприятия по обеспечению БДД предполагают в том числе эффективное управление дорожным движением. Под эффективностью, помимо безопасности, в данном случае понимается:

1) оперативность прохождения маршрута всеми участниками дорожного движения, во многом определяющая материальные затраты (также один из показателей эффективности управления дорожным движением), которые несут участники дорожного движения и владельцы транспортных средств;

2) точность следования транспортных средств, движущихся по установленным маршрутам – показатель, демонстрирующий наибольшую чувствительность к изменению состояния инфраструктуры дорожного движения;

3) экологичность дорожно-транспортной инфраструктуры, находящаяся в прямой зависимости от величины задержек транспортных средств внутри сети (например, при заторах, или в штатном режиме на пересечениях), и, как следствие, коррелирующая с показателем 1) оперативность;

4) надежность инфраструктуры дорожного движения в целом, которая определяется надежностью всех ее компонент: людей, дорог, транспорта, а также технических средств организации дорожного движения.

Анализ последних исследований и публикаций. Работы в области организации и обеспечения безопасности дорожного движения (Клинковштейн Г.И., Кременец Ю.А., Дерех З.Д., Рейцен Е.А., Ткачук С.П., Душник В.Ф.) подтверждают, что наиболее перспективным направлением решения подобных задач является применение автоматизированных систем управления дорожным движением, которые определяют и задают оптимальные управления светофорным объектам [1]. В частности, для сложных городских сетей необходимо определять оптимальное управление светофорными объектами, которое должно обеспечить наибольшую пропускную способность при минимальных потерях времени с учетом оперативных данных о дорожном движении, а также множества влияющих на дорожное движение факторов, таких как характеристики улично-дорожной сети (УДС), погодные условия, и другие [2]. При этом управляющая система должна быть обеспечена специальными техническими средствами для оперативной регистрации параметров транспортных потоков – детекторами присутствия либо прохождения транспорта [5].

Цель данного исследования – повышение эффективности управления городским дорожным движением за счет создания новых интеллектуальных средств мониторинга и управления дорожным движением в реальном масштабе времени на основе использования современных и перспективных компьютерных технологий.

Объектом исследования являются современные технологии мониторинга и управления дорожным движением в условиях крупного города.

Предмет исследования — современные программно-аппаратные средства и облачные сервисы мониторинга и управления дорожным движением.

Для достижения поставленной цели предлагается создание интеллектуальной инфраструктуры дорожного движения (ИИДД), которая отличается от существующих АСУДД структурной интеграцией трех взаимосвязанных интерактивных компонентов [9]:

- 1) существующие сервисы электронной картографии со средствами радиолокации и радионавигации;
- 2) новый облачный сервис мониторинга и управления дорожным движением на основе дорожных контроллеров;
- 3) усовершенствованные средства радиочастотной идентификации автомобиля и доступа к облачным сервисам.

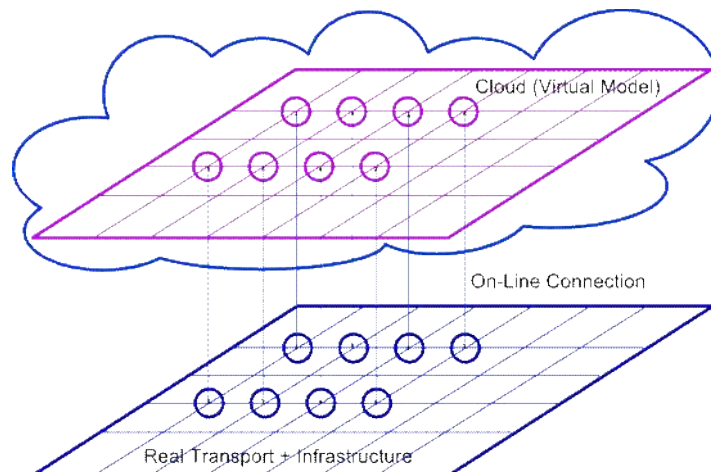


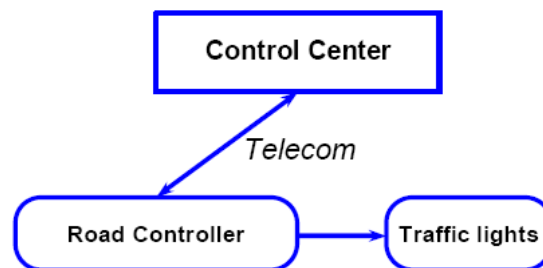
Рис. 1. Отображение инфраструктуры и транспорта в облаке

Научная новизна проекта определяется системной интеграцией облака мониторинга и управления, блоков радиочастотной идентификации транспорта, а также средств управления дорожной инфраструктурой, что дает возможность автоматизировать процессы оптимального управления транспортными средствами и дорожным движением в режиме реального времени для решения социальных, гуманитарных, экономических и экологических проблем, и создания таким образом ИИДД на уровне города, региона, страны, Планеты.

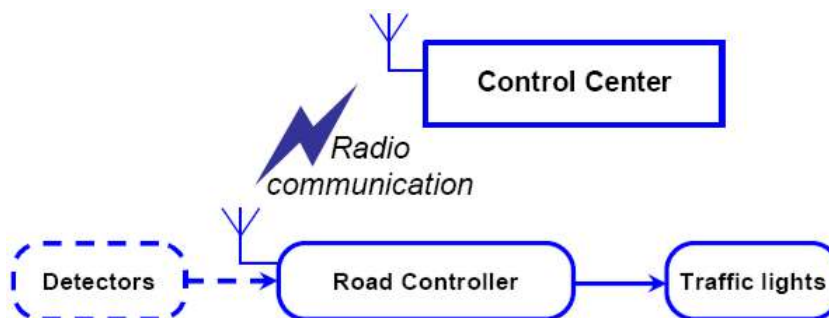
Основные результаты исследования. Анализ существующих АСУДД показал, что при проектировании подобных систем в основном используется двухуровневая архитектура типа "клиент-сервер" (рис. 2), где в качестве клиентов выступают дорожные контроллеры (ДК), а

центральный сервер обеспечивает централизованное управление дорожным движением на базе центрального управляющего пункта (ЦУП), координируя работу ДК и принимая информацию о его состоянии для отображения диспетчеру. Такая структура имеет ряд недостатков, среди которых основным является переход на локальное управление всех светофорных объектов в случае отказа сервера или другого компонента ЦУП. Можно выделить 2 поколения таких систем.

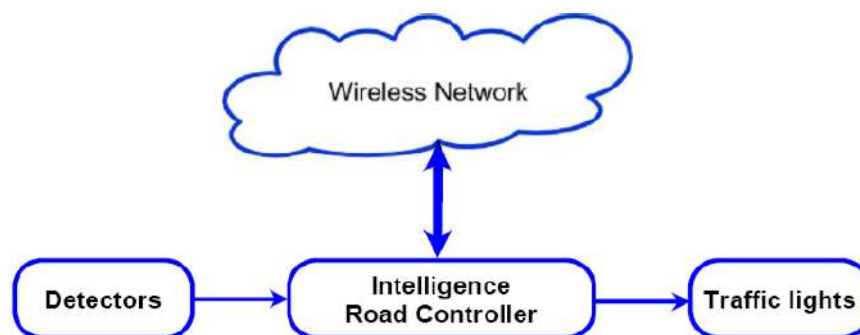
АСУДД первого поколения (рис. 2, а) к тому же состоят из устаревших контроллеров, таких как УК-2, ДКМ, Приор-05, среди которых только ДКМ продолжает уверенно функционировать, проработав несколько десятков лет. Также оставляет желать лучшего проводная телефонная сеть связи между центром и ДК, эффективность которой сомнительна. Отсутствие полноценной обратной связи с ЦУП в контуре управления этой системы приводит к тому, что неисправно работающий светофорный объект можно выявить только путем его внешнего осмотра. Осуществлять оперативное отслеживание параметров транспортных потоков с использованием детекторов транспорта в таких условиях очень проблематично [5]. Все это говорит о низкой управляемости и надежности таких систем.



а). Структура АСУДД первого поколения (устаревшая)



б). Структура АСУДД второго поколения (современная)



в). Структура АСУДД третьего поколения (перспективная)

Рис. 2. Эволюция технологий управления дорожным движением

Иначе обстоит дело с современными АСУДД второго поколения (рис. 2, б), где в качестве ДК используются поставляемые на украинский рынок контроллеры КДК ООО «Система-Сервис», г. Харьков [7], а также контроллеры фирмы Росток-Элеком, г. Киев [3], которые также предлагают современное оборудование ЦУП и средства проводной и (чаще) радио связи с дорожными контроллерами на периферии, а также спектр детекторов транспорта для оперативного отслеживания параметров транспортных потоков. Такие системы более надежны и устойчивы в

эксплуатации, демонстрируют высокую прогрессивность и наблюдаемость состояния объекта управления, и вместе с тем высокую стоимость. Однако они не лишены характерного для двухуровневых структур недостатка: остановки координированного управления в случае выхода из строя оборудования ЦУП.

Указанного недостатка лишена АСУДД третьего (рис. 2, в), перспективного, поколения. Она имеет структуру, схожую с АСУДД второго поколения, за исключением того, что управляющий центр интегрирован в "облако", то есть функционирует посредством сервисов, доступных через беспроводную глобальную коммуникационную сеть, имеющую распределенную структуру и доступную в пределах всей планеты. Последнее обстоятельство гарантирует устойчивое функционирование АСУДД в режиме координированного управления даже при физическом отсутствии ЦУП, так как множество облачных серверов, координирующих отдельные группы ДК, выделенные по географическому признаку, в состоянии обеспечить координированное управление, взаимодействуя друг с другом через вычислительную сеть. Диспетчерские же функции оказываются доступны посредством защищенных облачных сервисов в любой момент времени из любой точки планеты. Следует отметить, что такая система обязана иметь на нижнем уровне управления комплекс технических средств для мониторинга состояния дорожно-транспортной сети, а в облаке — вычислительные ресурсы, поддерживающие этот мониторинг — так называемые *мониторы инфраструктуры*. В противном случае об эффективном управлении дорожным движением не может быть и речи.

Однако инновационность этой системы заключается прежде всего в интеллектуализации основных процессов организации дорожного движения, синхронизирующей процессы развития реальной дорожно-транспортной инфраструктуры и виртуального киберпространства, и синтезирующей структурированные и взаимноинтегрированные компоненты этой интеллектуальной системы путем создания в киберпространстве облачных сервисов точного мониторинга и оптимального управления указанными реальными процессами. Такая интеллектуальная облачная инфраструктура становится самодостаточной координирующей надсистемой, радикально решающей проблему безопасности дорожного движения путем полного исключения аварийных и предаварийных ситуаций на дорогах за счет тотального контроля и интеллектуального управления.

Рассмотрим данную архитектуру более детально с точки зрения инфообмена между четырьмя ее компонентами (рис. 3) [9]. Cloud Servers – серверы, создающие облако долговременного хранения распределенных данных и сервисов; Buffer Computers – буферные компьютеры, обеспечивающие сбор данных от мониторов инфраструктуры и доставку сервисов управления дорожным контроллерам; C-RFID – компьютерные блоки радиочастотной идентификации транспортных средств; I-CMC – инфраструктурные контроллеры мониторинга и управления дорожным движением на основе радиочастотной идентификации транспортных средств.

Структура коммуникационной интеграции четырех компонентов АСУДД представлена транзакциями: $(R1 * R2) = (SC, BC, C-RFID)$ – доставка облачных сервисов к потребителю; $(R1 * R3) = (SC, BC, I-CMC)$ – инфообмен с дорожными контроллерами. Маршрут первого типа использует традиционные технологии GPRS, HSPA, Wi-Fi, WiMAX на основе сети Internet. Для второго типа транзакций, ввиду их чрезвычайной важности, а также высоких требований к надежности, помехоустойчивости и защищенности, необходимы дополнительные научно-технические исследования в процессе создания масштабируемого прототипа.

Предполагается, что в блоке C-RFID будут записаны индивидуальный код транспортного средства, электронный код регистрации по месту проживания, а также код водителя, выполняющего эксплуатацию данного автомобиля в текущий момент времени. Считывать триаду кодов должны радиоустройства, которыми будут оснащены все светофорные объекты, мосты, тоннели, железнодорожные переезды и другие пункты УДС, существенные с позиции управления дорожным движением, в том числе некоторые критические контрольные точки [9].

С точки зрения данного исследования наиболее интересен компонент I-CMC. Он представляется в виде матрицы, элементами которой являются дорожные контроллеры (R-PLC), а столбцы соответствуют сегментам УДС, контролируемым серверами сегмента (RSS), которые управляются буферным компьютером АСУДД. Модуль RSS представляет надежный компьютер промышленного исполнения, а компонент R-PLC построен на базе компактного и достаточно мощного программируемого логического контроллера SIMATIC S7-1200 фирмы SIEMENS для

программирования технологических процессов, в том числе для решения задач автоматического регулирования дорожного движения.

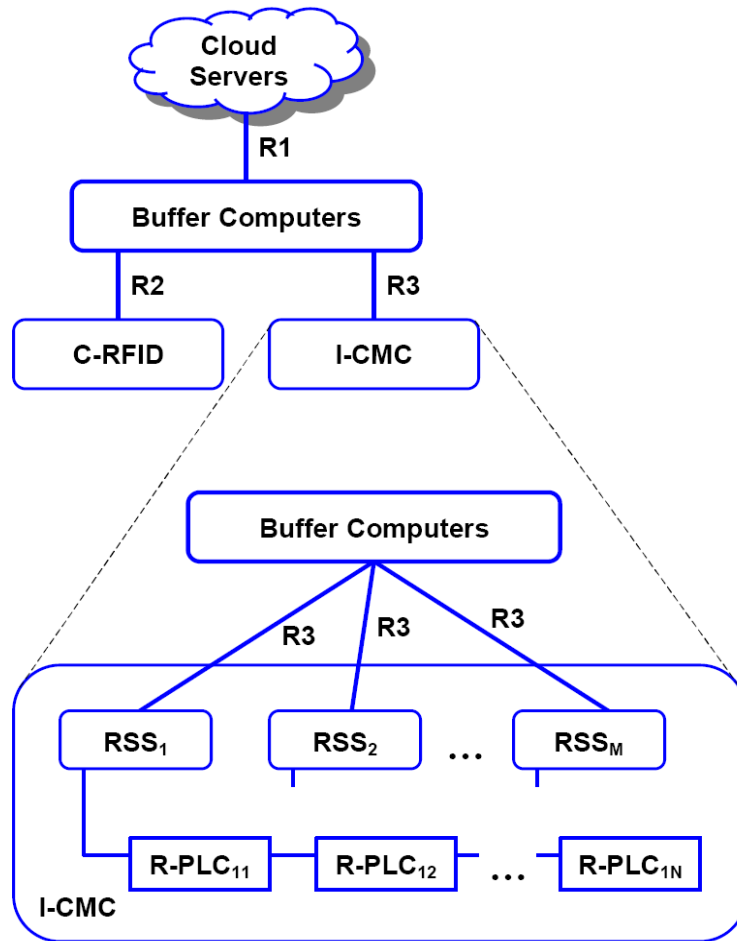


Рис. 3. Архитектура перспективной АСУ ДД с тремя уровнями управления

На рис. 4 представлена схема технической реализации инфраструктуры городского дорожного движения (ИГДД) на основе АСУДД третьего поколения. В данной схеме просматривается три уровня управления [4]. На уровне 1 реализуется технологическое управление дорожным движением в автоматическом и ручном режиме посредством специальных компьютерных и периферийных устройств. В качестве вычислительного средства используется ПЛК S7-1200. Источниками данных для управления являются информационно-регистрирующие устройства двух типов: 1) RFID-считыватель, собирающий данные с транспондеров участников дорожного движения в пределах светофорного объекта, и 2) детекторы транспорта, регистрирующие параметры входящих потоков транспорта. Исполнительные устройства представлены электромагнитными реле, обеспечивающие срабатывание светофоров. Коммуникация с центром управления осуществляется через уровень 2.

Уровень 2 является промежуточным (коммуникационным). Он выполняет две функции: 1) обеспечивает коммуникацию между центральной частью ИГДД и локальными контроллерами сегмента, и 2) реализует центральное управление контроллерами сегмента в случае отсутствия связи с уровнем 3. Технически реализован на базе промышленных компьютеров средней мощности, каждый из которых выполняет функцию буферного компьютера сегмента (RSS). Связан каналом Ethernet с ведущим контроллером R-PLC-M (Master), который в свою очередь связан промышленной сетью PROFINET с ведомыми контроллерами R-PLC-S (Slave). RSS имеет выход в Интернет и таким образом интегрируется в облачную инфраструктуру.

Уровень 3 – уровень центрального управления. Осуществляет координированное управление светофорным объектом на уровне 1 через буферные компьютеры уровня 2. Содержит серверную часть, построенную на базе мощных серверов, оснащенных средствами облачных вычислений (cloud servers), а также человеко-машинный интерфейс, включающий наряду с рабочими станциями аппаратно-программные средства визуализации и оперативного управления.

Поскольку данный проект посвящен интеллектуализации управления дорожным движением в условиях отдельного светофорного объекта, остановимся подробно на технической реализации этого аспекта ИГДД.

Lev. III

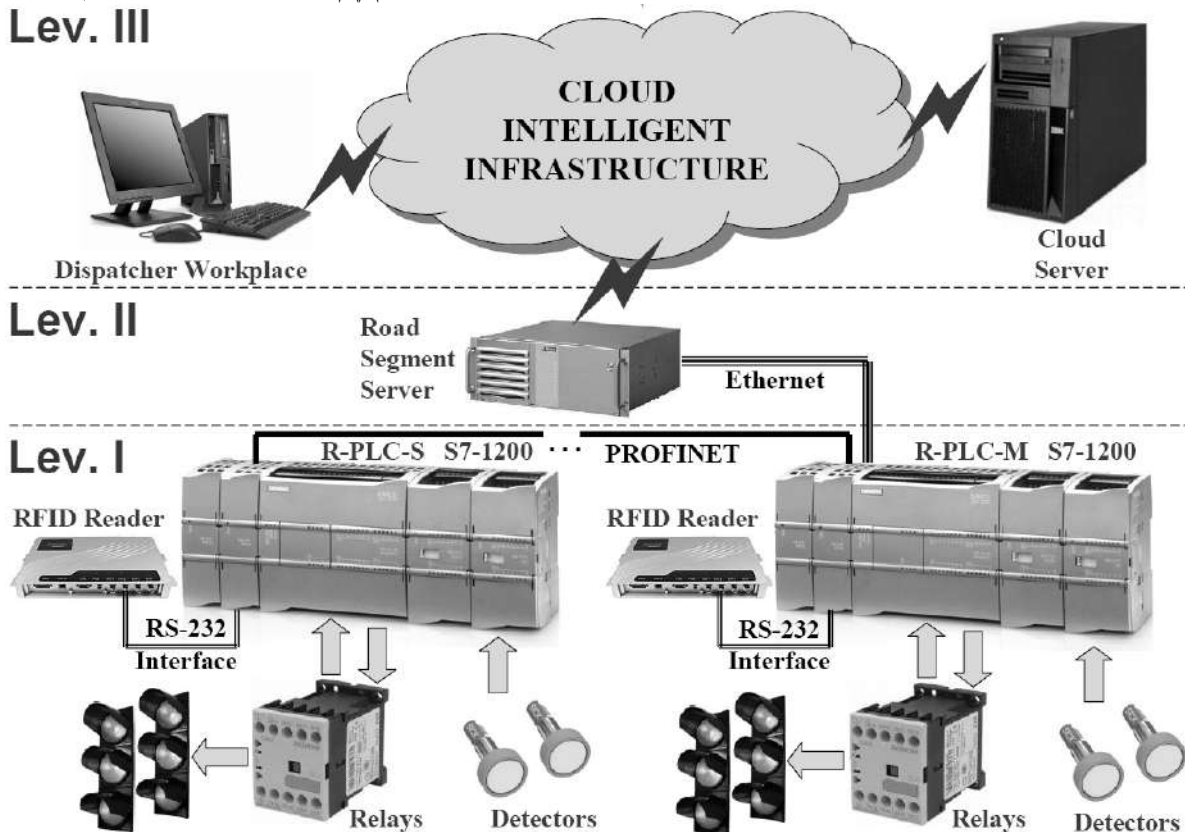


Рис. 4. Техническая реализация инфраструктуры городского дорожного движения

В качестве управляющего вычислительного устройства выбран вышеупомянутый ПЛК S7-1200. Это один из самых новых контроллеров фирмы SIEMENS для программирования технологических процессов [8]. Контроллер SIMATIC S7-1200 способен решать задачи автоматического регулирования и управления перемещением и может использоваться в машиностроении, системах управления предприятием, во многих других областях. Обладает широкими функциональными возможностями и отличается относительно невысокой стоимостью.

Каждый контроллер комплектуется стационарным RFID-считывателем, который обеспечивает одновременное чтение данных с большого количества транспондеров участников дорожного движения в радиусе до 12-ти метров и имеет несколько интерфейсов для взаимодействия с контроллером (см., напр., [10]). Взаимодействие контроллера с другими периферийными устройствами и с управляющим центром структурно показано на рис. 5.

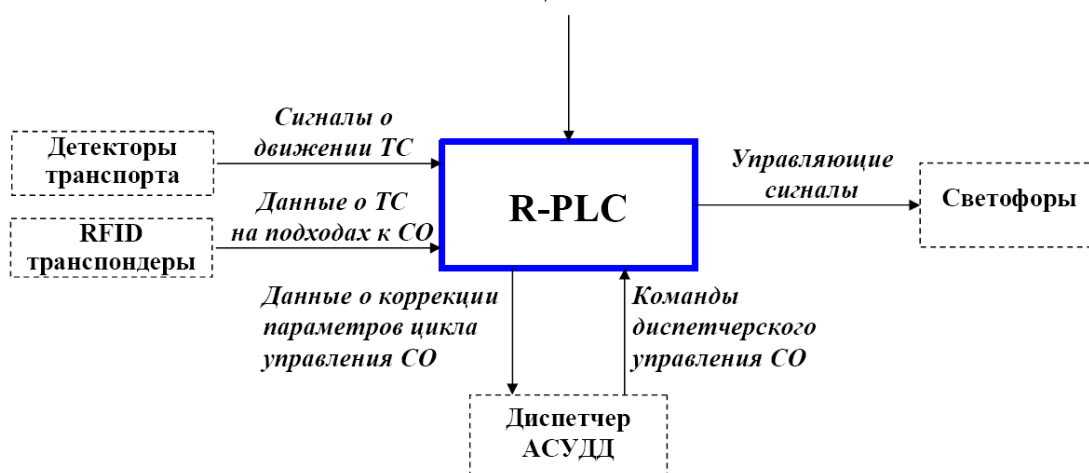


Рис. 5 Ячейка транспортной инфраструктуры

Подсистема реализует такие автоматизированные функции:

1. установка параметров цикла светофорного регулирования по команде диспетчера (диспетчерское управление);
2. регистрация данных о движении транспорта на подходах к светофорному объекту.
3. выработка корректив параметров цикла светофорного регулирования (интеллектуальное управление);
4. формирование и вывод управляющих воздействий на светофоры (непосредственное управление);
5. передача данных о коррекции параметров цикла светофорного регулирования диспетчеру (обратная информационная связь).

Функция 2 обеспечивает необходимую обратную связь с объектом управления. Технически ее выполнение обеспечивают детекторы транспорта и оборудование радиочастотной идентификации транспорта.

Для реализации интеллектуального управления (функция 3) необходимо учитывать также и параметры потоков на смежных светофорных объектах, принятые по сети от соответствующих дорожных контроллеров. Идентификация и оценивание этих параметров, а также подготовка на их основе данных для интеллектуального управления – предмет отдельного исследования. Кроме того, при проведении системных исследований предметной области было установлено влияние на движение транспорта по улично-дорожной сети крупного города большого количества разнородных параметров, некоторые из которых, такие как сложность и степень опасности пересечения, погодные условия, видимость дорожной разметки, насыщенность средствами светофорного регулирования и дорожными знаками, а также эффективность их восприятия участниками дорожного движения и др., трудно поддаются формализации классическими аналитическими и вычислительными методами. Более того, большинство выявленных параметров характеризуются высокой степенью неопределенности, которую не удалось погасить вероятностными и статистическими методами. В связи с этим для погашения неопределенности оказалось целесообразно применить методику нечеткого вывода и построить на ее основе модель нечеткого управления светофорным регулированием на заданном светофорном объекте, структура которой показана на рис. 6.



Рис. 6. Структура модели нечеткого управления светофорным регулированием на заданном светофорном объекте

На вход модели поступают указанные выше данные о транспортных потоках (входная информация на рис. 5) с заданного (2) и смежных (3) светофорных объектов, а также смоделированные параметры на предыдущем цикле регулирования (обратная связь 1). Характеристики УДС (4), обладающие, как указано выше, высокой степенью неопределенности, определяются нечеткими переменными. Все входные данные в модели на рис. 6 подлежат фаззификации классическими методами нечеткой логики [6]. Логика регулирования дорожного движения представляется в виде базы нечетких правил продукции. Блок нечеткого вывода генерирует значения трех выходных нечетких переменных, выражающих параметры цикла светофорного регулирования на пересечении: длительность цикла, длительность зеленой фазы по главному направлению и длительность переходного интервала. Агрегированные и дефаззифицированные значения этих параметров применяются для задания фактических параметров светофорного регулирования на данном светофорном объекте.

Такова в общем концепция нечеткого управления светофорным регулированием на заданном светофорном объекте. Детальная разработка этой концепции – предмет отдельного исследования, требующего работы с экспертами по организации и регулированию дорожного движения и, по крайней мере, имитационного моделирования данного процесса в условиях как детерминированного, так и нечеткого управления.

Ожидаемый эффект от внедрения интеллектуальных технологий в инфраструктуру городского дорожного движения:

- сокращение времени прохождения маршрута внутри дорожно-транспортной сети, покрываемой ИГДД, всеми участниками дорожного движения;
- уменьшение материальных затрат, которые несут участники дорожного движения и владельцы транспортных средств;
- повышение точности следования транспортных средств, движущихся по установленным маршрутам;
- улучшение экологической обстановки за счет сокращения задержек транспортных средств внутри сети;
- понижение аварийности на дорогах дорожно-транспортной сети, покрываемой ИГДД;
- устойчивое и безотказное функционирование ИГДД.

Дальнейшие приоритетные исследования в этой области [9]:

- экспериментальные исследования интеллектуального дорожного контроллера в реальных условиях городского дорожного движения;

- исследования и конструкторские работы по обязательному обеспечению транспортных средств стационарными устройствами радиочастотной идентификации для интеграции в ИГДД;
- разработка облачных сервисов, образующих структурированное киберпространство организации и обеспечения безопасности дорожного движения в крупных городах;
- создание регионального сектора всемирной интеллектуальной инфраструктуры дорожного движения, основанной на технологиях радиочастотной идентификации участников дорожного движения (всех без исключения).

Выводы.

1. Выполнен обзор существующих технологий автоматизированного управления дорожным движением в крупных городах Украины. Определены приоритетные направления исследований для создания современных и перспективных технологий в данной отрасли.
2. Представлено взаимодействие инфраструктурных средств мониторинга и интеллектуального управления дорожным движением, показана схема технической реализации данной инфраструктуры на основе программируемого логического контроллера S7-1200.
3. Предложена концепция модели нечеткого управления дорожным движением на заданном регулируемом пересечении в условиях крупного города, интегрированная на базе программируемых логических контроллеров.

Перечень ссылок.

1. Дерех З.Д., Рейцен Є.О. Наукові шляхи реалізації програми забезпечення безпеки дорожнього руху в Україні // Наук.-техн. вісник "Безпека дорожнього руху України". – № 1(2). – 1999. – С.19 – 23
2. Душник В.Ф. До питання організації дорожнього руху на вулично-дорожній мережі // Наук.-техн. вісник «Безпека дорожнього руху України».—К.: ТОВ "Журнал "Радуга".— 2003.— № 1-2 (15). – С. 39 – 41.
3. Аппаратно-программный комплекс оборудования для управления дорожным движением / Официальный сайт Киевской АК "Росток".— К., 2014 // режим доступа: <http://rostok-elekom.com/>
4. Белов Ю.В., Гузь О.А., Полетайкин А.Н. Трехуровневая архитектура системы распределенной автоматизации управления дорожным движением // Обеспечение безопасности и комфорта дорожного движения: проблемы и пути решения / Материалы международной научно-практической конференции. – Харьков: ХНУРЭ, ХНАДУ, 2011. – с. 123 – 126.
5. Кременец Ю. А. Технические средства организации дорожного движения. — Москва, ИКЦ «Академкнига», 2005
6. Леоненков, А.В. Нечеткое моделирование в среде MATLAB и flzzyTECH // А.В. Леоненков. – СПб.: БХВ Петербург, 2005. – 736 с.: ил.
7. Оборудование для управления дорожным движением / Официальный сайт ООО "Система Сервис".— Харьков, 2014 // режим доступа: <http://komkon.ua/product/rtc/>
8. Продукты для промышленной автоматизации: Интерактивный каталог СА 01/2014. – Сименс, Украина, Департамент IA&DT
9. Хаханов В.И, и др. Зеленая волна – облако мониторинга и управления дорожным движением (Green wave traffic on cloud) // Автоматизированные системы управления и приборы автоматики. – Харьков: ХНУРЭ, 2012, вып. 160. – с. 4 – 21.
10. Esker F. RFID in Vehicles.— NetWorld Alliance LLC.— 2012.— 143 p.
11. Roads, Injuries, Traffic: Data and statistics / Official site WHO, 2011 // data source: <http://www.who.int/research/en/>

УДК 621.34:621.313:621.9:62.83

Г.В.Савеленко, Ю.О.Єрмолаєв

Кіровоградський національний технічний університет

ДОСЛІДЖЕННЯ РОБОЧОГО ПРОЦЕСУ РОЗМІРНОЇ ОБРОБКИ ДУГОЮ НА ВЕРСТАТІ З ЕЛЕКТРОМЕХАНІЧНИМ ПРИВОДОМ

Г.В. Савеленко, Ю.О. Єрмолаєв. Дослідження робочого процесу розмірної обробки дугою на верстаті з електромеханічним приводом. В статті розглядається склад і технічні можливості програмно-технічного комплексу призначеного для реєстрації і обробки результатів наукових досліджень на прикладі електроерозійного верстата з електромеханічним приводом подачі електрода інструмента.

Ключові слова: реєстрація технологічних параметрів, аналого-цифрове перетворення, обробка результатів дослідження.

Рис.5. Літ. 10.

Г.В. Савеленко, Ю.А. Ермолаєв. Исследование рабочего процесса размерной обработки дугой на станке с электромеханическим приводом. В статье рассматривается состав и технические возможности программно технического комплекса предназначенного для регистрации и обработки результатов научных исследований на примере электроэрозионного станка с электромеханическим приводом подачи электрода инструмента.

Ключевые слова: регистрация технологических параметров, аналого-цифровое преобразование, обработка результатов исследования.

G.V. Savelenko, Y.A. Ermolaev. Study workflow dimensional processing of arc on a machine with electromechanical drive. The article discusses the composition and technical capabilities of software and hardware complex designed for the registration and processing the results of research on the example of electric discharge machine with an electromechanical drive filing elektroda tool.

Keywords: registration process parameters, analog-to-digital conversion, processing of the results of the study.

Постановка проблеми. Одним з розповсюджених способів описання будь-якого робочого процесу є реєстрація значень технологічних параметрів системи, яка досліджується. Сутність розмірної обробки дугою (РОД) полягає в тому, що обробку здійснюють стаціонарною електричною дугою, яка горить поміж електрода-інструмента (ЕІ) та електрода-заготовки (ЕЗ) в потоці робочої рідини при динамічному тиску не менше 1...2 кПа. Принципова схема процесу РОД зображена на рис.1. У порівнянні з відомими способами розмірної електроерозійної обробки, спосіб РОД дозволяє збільшити продуктивність обробки приблизно на порядок і більше, зменшити приблизно в два рази питомі витрати електроенергії та в 3...5 разів знизити вартість джерел живлення технологічним струмом [1]. Конструктивні особливості верстата РОД з електромеханічним приводом розглянуті в праці [2].

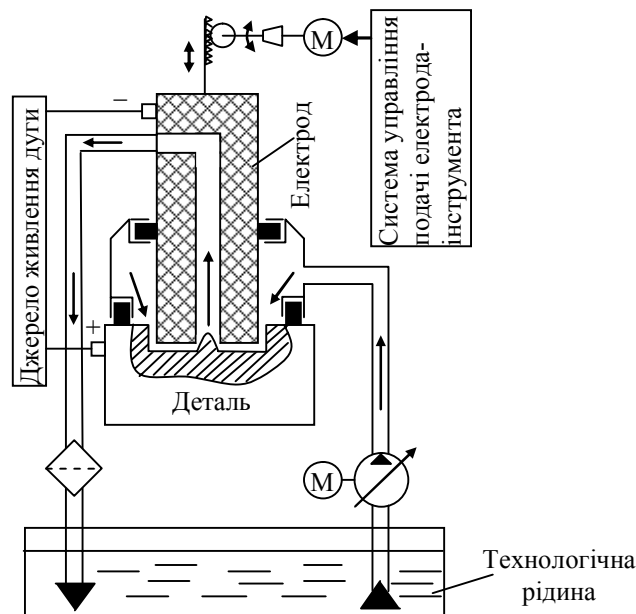


Рис.1. Принципова схема процесу РОД [авторська розробка]

Для дослідження процесу РОД при усталеному режимі і особливо на перехідних режимах, необхідні набори статистичних даних: струму і напруги електричної дуги, тиску технологічної рідини, лінійне переміщення ЕІ, струм, напруга і частота обертання двигуна подачі ЕІ, напруга завдання привода подачі. Вирішити дану задачу можливо, створив спеціальне обладнання для реєстрації миттєвих значень цих величин. Це потребує використання точних швидкодіючих АЦП та мікроконтролерів і розробку та відлагодження програмних і апаратних засобів, що в значній мірі впливає на час розробки та вартість системи.

Аналіз останніх досліджень і публікацій. Останнім часом на ринку вимірювальної техніки з'явилась значна кількість автоматизованих систем збору даних та реєструючих комплексів, які можуть виконувати багатоканальний запис вимірювальних величин. В багатьох наукових працях [3, 4, 5, 6, 7] наводяться приклади побудови таких вимірювальних систем, використовуючи універсальні та спеціалізовані АЦП таких виробників як National Instruments, Texas Instruments, Analog Devices, Microstar Laboratories, L-Card, ХОЛИТ Дэйта Системс, але кожна з представлених систем, що відноситься до нижнього цінового діапазону, розробляється під конкретні умови використання і не може виконувати поставлену перед нами задачу без суттєвої зміни структури вимірювальних каналів та блока спряження. Універсальні реєструючі комплекси середнього і високого цінового діапазону можуть вирішувати поставлену задачу, але їх висока вартість обмежує їх використання.

Невирішені частини проблеми. Готових бюджетних рішень по реєстрації і обробці результатів експериментальних досліджень на ринку не представлено. Окремо пропонуються блоки АЦП, блоки нормування сигналу та програмне забезпечення для обробки оцифрованих даних. Мікропроцесорні системи збору даних різних виробників однакового цінового діапазону мають приблизно однакові технічні можливості. Вибір продукції того чи іншого виробника обумовлений ціною політикою, наявністю представництва в країні та якістю служби підтримки. Саме тому було прийняте рішення розробити власну багатоканальну вимірювальну систему на базі сучасної мікросистеми збору даних вітчизняного виробництва.

Метою дослідження є розробка та впровадження програмно-технічного комплексу для реєстрації робочого процесу параметрів верстату РОД з електромеханічним приводом подачі ЕІ.

Методи вирішення задачі. Вибір програмно-апаратних засобів для проведення експериментальних досліджень процесу РОД на різних режимах роботи системи автоматичного управління електромеханічного приводу переміщення ЕІ.

Основні результати дослідження. Попередня реєстрація сигналів робочого процесу РОД з ціллю визначення їх характеру, а також з ціллю визначення необхідних параметрів аналогово-цифрового перетворювача показала, що перетворювач повинен забезпечувати частоту квантування реалізацій змінних одного каналу не менше ніж 20 кГц, що цілком достатньо для обробки стохастичних процесів за частотним діапазоном. Діапазон вхідної напруги перетворювача повинен складати ± 10 В, орієнтуючись на датчики струму і напруги. Кількість аналогових входів повинно бути не менше восьми. Усім вище означеним критеріям відповідає вітчизняна мікросистема збору даних виробництва ХОЛИТ Дэйта Системс типу m-DAQ12 [8], яка й була вибрана нами для побудови програмно-технічного комплексу (ПТК). Мікросистема дозволяє працювати з 4-ма диференційними або 8-ма каналами із загальною землею для введення аналогових сигналів.

Кожен з аналогових каналів підключається до АЦП через програмно-керуємий підсилювач, що дозволяє задавати один з чотирьох діапазонів вимірювання напруги (± 10 В ± 5 В, ± 2.5 В, ± 1.25 В). Модулі забезпечують безперервний (синхронізований по внутрішньому таймеру) збір даних на частотах дискретизації АЦП від 65 Гц до 100 кГц.

Вибір первинних перетворювачів, що застосовуються в вимірювальних каналах системи описується в [9]. Миттєві значення вимірювальних величин, що підлягають реєстрації можуть бути за рівнем амплітуди сигналу більше допустимого значення, яке дозволено згідно технічної документації на мікросистему збору даних (МЗД) з інтерфейсом USB типу m-DAQ12. Тому між датчиками і МЗД застосовується блок спряження сигналу, який перетворює і нормує сигнал по рівню допустимих значень по входу пристрою МЗД. Принципова електрична схема блоку спряження каналу напруги наведена на рис.2. Схемні рішення блоку спряження для каналів, що вимірюють інші фізичні величини наведені в [9].

Програмне забезпечення ПТК повинно виконувати задачі по налаштуванню каналів введення-виведення МЗД, приймати і відображати поточні дані, вносити корекцію і записувати коментарії до записаних блоків інформації, виконувати математичну обробку отриманих даних, зберігати, імпортувати і друкувати результати досліджень. Відповідно програмне забезпечення (ПЗ) з таким функціоналом не поставляється разом з МЗД. До складу поставки МЗД входять драйвери для роботи з операційною системою, DLL-бібліотеки і приклади роботи з нею для середовища графічного програмування Labview, і програма «Осцилограф-реєстратор». Даній комплектації достатньо для написання програм під конкретну задачу реєстрації параметрів робочого процесу РОД, але створення спеціалізованого програмного забезпечення потребує тривалого часу написання і відладки програмного коду, що неможливо виконати без залучення спеціалістів високої кваліфікації і додаткового фінансового ресурсу.

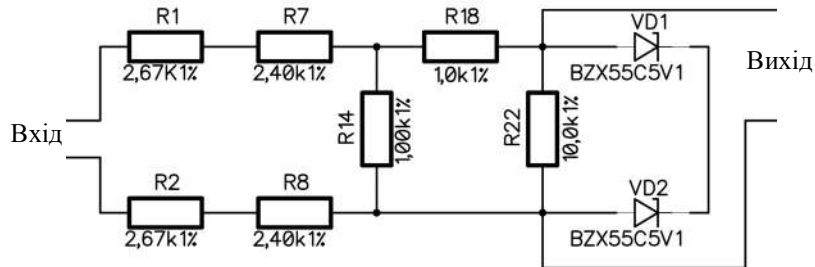


Рис.2. Принципова електрична схема блока спряження каналу напруги [авторська розробка]

Наші пошуки зупинились на готовому до використання програмному продукті «PowerGraph Professional», який позиціонується на ринку ПЗ, як продукт що дозволяє організувати запис, візуалізацію, обробку та збереження результатів дослідження з плат АЦП і мікросистем збору даних в тому числі і m-DAQ12. До складу програми «PowerGraph Professional» входить бібліотека функцій математичної і цифрової обробки сигналів, призначених для проведення обчислень і отримання розрахункових даних. Функції обробки сигналів можуть бути використані як після закінчення реєстрації (режим пост-обробки), так і безпосередньо в процесі отримання даних з мікросистеми збору даних m-DAQ12 (режим реального часу) [10]. Структурна схема описаної системи наведена на рис.3.

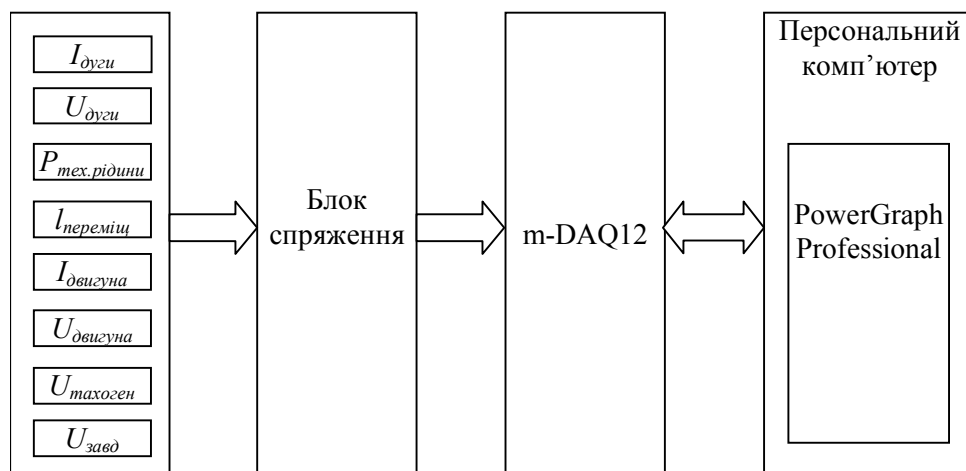


Рис. 3. Структурна схема програмно-технічного комплексу для дослідження РОД: $I_{дуги}$ – струм електричної дуги; $U_{дуги}$ – напруга дуги; $P_{тех.рідини}$ – тиск технологічної рідини; $I_{переміщ}$ – переміщення ЕІ; $I_{двигуна}$ – струм двигуна подачі ЕІ; $U_{двигуна}$ – напруга двигуна подачі ЕІ; $U_{тахоген}$ – напруга тахогенератора; $U_{завод}$ – напруга завдання. [авторська розробка]

Характерні часові ряди значень вимірювальних каналів, отримані в ході проведення експериментальних досліджень (марка графітового електрода МПГ7, марка деталі сталь Ст3), наведені на рис. 4. На рис. 4(а) зображені реальні миттєві значення вимірювальних величин при стаціонарному робочому процесі РОД. На рис. 4(б) зображені ті ж самі дані, але після обробки їх фільтром NoiseFilter, який був застосований до сигналу каналів напруги дуги, напруги тахогенератора, струму двигуна подачі ЕІ та тиску технологічної рідини, що призвело до видалення в них високочастотних шумів та випадкових викидів. Зображений процес на рис. 4 є квазіоптимальним режимом обробки деталі при РОД, який характеризується незначним коливанням миттєвих значень напруги дуги 23...40 В та струму дуги, значення якого залежить від підведеної потужності джерела живлення дуги.

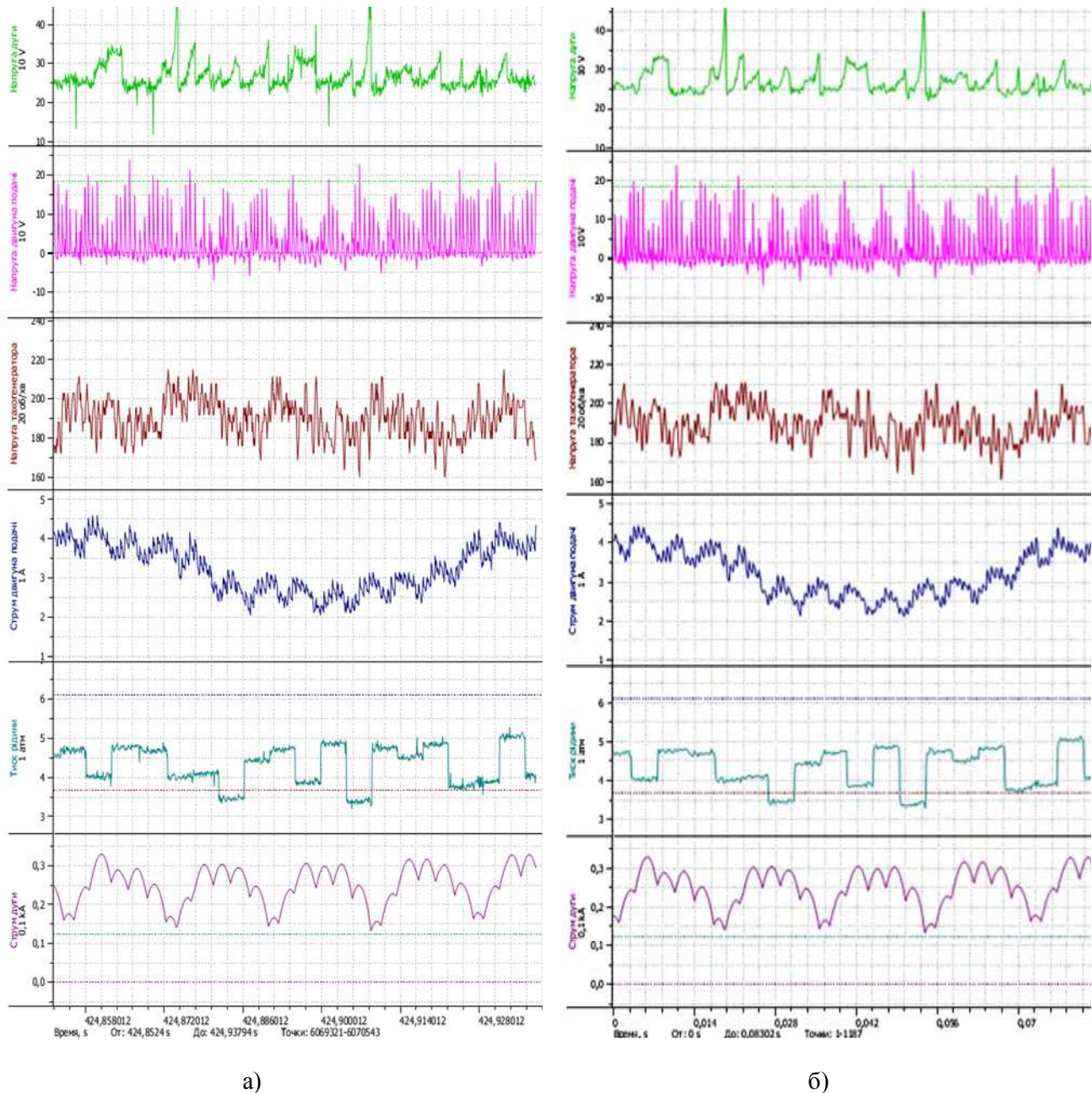


Рис. 4. Типові часові ряди значень вимірювальних каналів при усталеному режимі обробки деталі: а – без фільтрації сигналу; б – з фільтрацією від високочастотних шумів та випадкових викидів [авторська розробка]

Програма «PowerGraph Professional» виконує блочний запис результатів аналого-цифрового перетворення та дозволяє виконувати велику кількість операцій над блоками або їх виділеною частиною по переміщенню, копіюванню, видаленню, математичній обробці сигналів, як в масштабі реального часу так і після запису експерименту. Також в програмному продукті передбачена можливість аналізу даних, шляхом збільшення масштабу і редагування окремих ділянок і значень оцифрованого сигналу, шляхом додаткової графічної побудови на вісі часу і амплітуди, розрахунку кута нахилу та побудови дотичної в точці, пошук максимального, мінімального та середнього значення з побудовою відповідної проекції і багато інших інструментів (рис. 5).

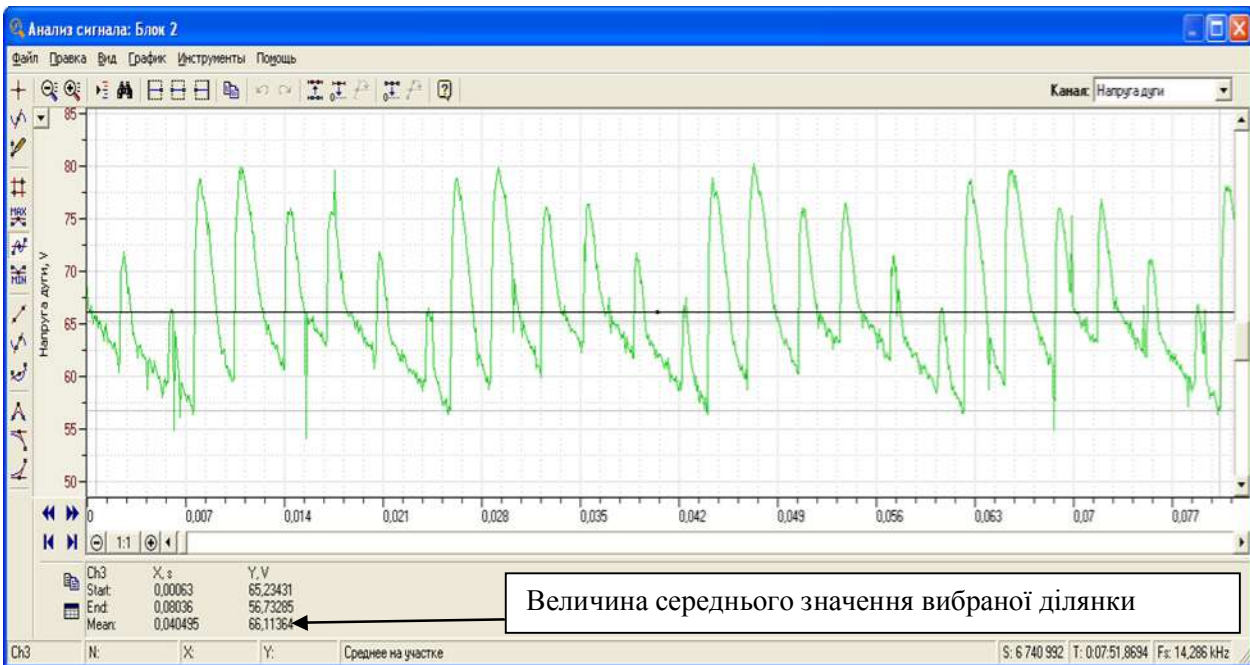


Рис. 5. Аналіз каналу «Напряга дуги»: пошук середнього значення [авторська розробка]

Висновки. В даній статті наведені:

- 1) приклад побудови програмно-технічного комплексу для дослідження робочого процесу параметрів верстату РОД з електромеханічним приводом подачі ЕІ з використанням універсального програмного продукту «PowerGraph Professional», як одного з компонентів автоматизованої вимірювальної системи для дослідження процесів РОД;
- 2) реалізації процесів, що характеризують РОД в усталеному режимі;
- 3) вплив застосування фільтра NoiseFilter програмного продукту «PowerGraph Professional»;
- 4) аналіз сигналу каналу «Напряга дуги» з визначеним середнім значенням вибраної ділянки.

Використання програмно-технічного комплексу довело його перспективність і доцільність при дослідженні робочого процесу розмірної обробки дугою на верстаті з електромеханічним приводом.

1. Носуленко В.І. Розмірна обробка металів електричною дугою: Автореф. дис. д-ра техн. наук: 05.03.07/ НТУУ «КПІ». – К., 1999. – 36 с., 51-54
2. Звіт про НДР № держреєстрації 0111U007656. Модернізація електромеханічної частини привода подачі електроерозійного верстата для розмірної обробки дугою. /Єрмолаєв Ю.О., Савеленко Г.В./ Кіровоград: КНТУ, 2011 – 34 с.
3. Девин Л.Н. Применение пакета PowerGraph для исследования процесса резания / Девин Л.Н., Сулима А.Г. //ИСМ им. В.Н. Бакуля // ПиКАД.- 2008.- № 3.- С. 24-26.
4. Врублевский А.Н. Техническое решение для исследования рабочего процесса двигателя внутреннего сгорания / Врублевский А.Н., Дзюбенко А.А., Вахрушев В.И. / ХНАДУ // ПиКАД.- 2008.- № 4.- С. 12-15.

5. Вахрушев В.И. Исследование крутильных колебаний двигателя ЗТД / Вахрушев В.И., Дороженко А.Н., Кондратенко В.Г. / Харьковское конструкторское бюро по двигателестроению // ПиКАД.-2009.- № 1.- С. 30-32.

6. Демиденко С.Н. Модульные КАМАК-системы автоматизации эксперимента / Демиденко С.Н., Апанасенко Л.С., Дашук В.Н., Куновский Э.Б. / Минск : Навука і тэхніка.- 1990. - 208 с

7. Зленко В.О. Автоматизованій комплекс досліджень терморезистивних властивостей наноструктурованих плівкових систем / Зленко В.О., Проценко С.І., Сафаріч Р. // Ж. нано- та електронної фізики. - 2009.-1 №2. - С. 34-41

8. Руководство пользователя V1.2. m-DAQ12. Микросистема сбора данных с интерфейсом USB. ХОЛИТ Дэйта Системс. - [Електронний ресурс] — Режим доступу:
<http://www.holit.ua/download/common/docs/hds/m-DAQ.pdf>

9. Звіт про НДР № держреєстрації 0111U007657. Розробка інформаційно-вимірювального комплексу для реєстрації стохастичних параметрів роботи електроерозійного верстата для розмірної обробки дугою. /Єрмолаєв Ю.О., Савеленко Г.В./ Кіровоград: КНТУ, 2012 – 25 с.

10. Справочное руководство ПО «PowerGraph» [Електронний ресурс] — Режим доступу:
<http://www.powergraph.ru>

УДК 004.94; 674.047

Соколовська Я.І., Семенюк В.Я., Мокрицька О.В.

Національний лісотехнічний університет України, Луцький національний технічний університет

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МОДЕЛЮВАННЯ В'ЯЗКОПРУЖНОГО СТАНУ ДЕРЕВИНИ У ПРОЦЕСІ СУШІННЯ З ВРАХУВАННЯМ БАГАТОФАЗНОСТІ

Соколовська Я.І., Семенюк В.Я., Мокрицька О.В. Математичне та програмне забезпечення моделювання в'язкопружного стану деревини у процесі сушіння з врахуванням багатофазності. Стаття присвячена математичному моделюванню деформаційно-релаксійних і тепломасообмінних процесів у гігроскопічних капілярно-пористих матеріалах як анізотропних трифазних середовищах, що має важливе значення для розроблення та обґрунтування енергоощадних технологій гідротермічного оброблення деревини. Сформульовано математичну модель тепломасоперенесення для періодів сталої і спадаючої швидкості сушіння капілярно-пористих матеріалів. Побудовано математичну модель реологічної поведінки деревини як трифазного середовища з врахуванням анізотропії тепломеханічних характеристик. Розроблено об'єктно-орієнтоване прикладне програмне забезпечення для чисельної реалізації математичних моделей на основі адаптації методу скінченних елементів. Встановлено закономірності впливу технологічних параметрів сушіння на процеси в'язкопружного деформування і тепломасоперенесення у твердій, рідкій і паровій фазах деревини.

Ключові слова: сушіння капілярно-пористих матеріалів, напруги і деформації дерева в процесі сушіння.

Форм. 23. Табл. 0. Рис. 4. Лит. 22.

Ya.I. Sokolowsky, V.Ya. Semenyuk, O.V. Mokritska Mathematical and simulation software viscoelastic state of wood during drying with regard phases. The paper is devoted to the mathematical modeling of strain- relaksiiy and heat and mass transfer processes in hygroscopic capillary- porous materials like 3-phase anisotropic media, is essential for the development and study of energy efficient technologies hydrothermal treatment of wood. Formulated a mathematical model for the period was heat and mass transfer and decreasing rate of drying of capillary- porous materials. A mathematical model of the rheological behavior of wood as a three-phase environment, taking into account the anisotropy of mechanical properties. Developed object-oriented application software for the numerical implementation of mathematical models based on the adaptation of the finite element method. The regularities of the influence of process parameters on the process of drying viscoelastic deformation and heat and mass transfer in solid, liquid and vapor phases wood.

Key words: drying of capillary- porous materials, stresses and strains of wood in the drying process.

Equat. 23. Tabl. 0. Fig. 4. Ref. 22.

Соколовський Я.І., Семенюк В.Я., Мокрицька О.В. Математическое и программное обеспечение моделирования вязкоупругого состояния древесины в процессе сушки с учетом многофазности. Статья посвящена математическому моделированию деформационно- релаксационных и тепломассообменных процессов в гигроскопичных капиллярно- пористых материалах как анизотропных трехфазных средах, имеет важное значение для разработки и обоснования энергосберегающих технологий гидротермической обработки древесины. Сформулирована математическая модель тепломассопереноса для периодов постоянной и убывающей скорости сушки капиллярно- пористых материалов. Построена математическая модель реологического поведения древесины как трехфазной среды с учетом анизотропии тепломеханических характеристик. Разработано объектно- ориентированное прикладное программное обеспечение для численной реализации математических моделей на основе адаптации метода конечных элементов. Установлены закономерности влияния технологических параметров сушки на процессы вязкоупругого деформирования и тепломассопереноса в твердой, жидкой и паровой фазах древесины.

Форм. 23. Табл. 0. Рис. 4. Лит. 22.

Вступ

Актуальність досліджень. Створення нових та вдосконалення існуючих енерго- та ресурсозберігаючих технологій процесу зневоднення гетерогенних капілярно-пористих матеріалів набуває важливого практичного значення у зв'язку з високими вимогами до якості готової продукції, потребою зниження фінансових та часових витрат на процес промислового впровадження. У вирішенні цієї важливої проблеми значну роль відіграє розроблення математичних моделей для дослідження деформаційно-релаксацийних і тепломасообмінних процесів під час сушіння капілярно-пористих матеріалів, зокрема деревини, з врахуванням багатофазності і багатокомпонентності матеріалу.

Аналіз відомих досліджень. Незважаючи на значні успіхи у цій галузі, на сьогодні не існує єдиного феноменологічного підходу щодо моделювання процесів деформування і тепломасоперенесення у гетерогенних капілярно-пористих структурах. У теоріях сушіння, сорбції та фільтрації розходження носить принциповий характер, хоча за своїм змістом вони мають відображати сумісні фізичні процеси у капілярно-пористих середовищах. Наявні математичні моделі та методи аналізу деформування та тепломасоперенесення в деревині у процесі сушіння[1-5], в основному, базуються на підході до структурної будови матеріалу як гомогенної, не беруть до уваги особливостей зміни в'язкопружної поведінки матеріалу з врахуванням кінетики фазових переходів. Ці процеси в основному досліджені для задач в одновимірній постановці із залученням значної кількості допущень[6,7]. Тому актуальним завданням є розроблення математичних моделей процесу сушіння капілярно-пористих матеріалів, зокрема деревини, як трифазного середовища, що складається з твердої фази (деревної речовини), рідкої і пароповітряної фаз. Математичні моделі тепломасоперенесення та деформування, що враховують багатофазність капілярно-пористих матеріалів у процесі сушіння, уможливають прогнозування особливостей зміни вологовмісту, температури окремих фаз, напружено-

деформівного стану на всіх етапах процесу сушіння деревини та дозволяють вирішувати задачі інтенсифікації технологій сушіння.

Метою дослідження є розроблення двовимірних математичних моделей та прикладного програмного забезпечення для їх чисельної реалізації та встановлення закономірностей анізотропного в'язкопружного деформування деревини як багатофазної структури у процесі конвективного сушіння.

Математична модель в'язкопружного стану деревини як трифазної системи. У силу складної стохастичної капілярної структури деревини, яка характеризується різною за величиною і неоднаковою геометричною формою елементів, визначити дійсні геометричні розміри капілярів є практично неможливо. Перехід від опису явищ в окремій фазі до континуальних рівнянь капілярно-пористого матеріалу може бути отриманий на основі об'ємного усереднення мікрорівнів для макроскопічних параметрів кожної фази. Тому в подальшому приймаються допущення[8-10]:

– характерні розміри капілярно-пористої структури деревини є набагато більші за молекулярно-кінетичні розміри і набагато менші за відстані, на яких відбувається суттєва зміна макроскопічних параметрів;

– малі значення деформацій і переміщень твердої фази та її нестисливість і сталість густини;

– парогазова суміш (повітря і волога) характеризується властивостями ідеального газу;

– сукупність мікрокапілярів становить собою систему циліндрів у клітинах деревини із змінним радіусом r_K , який залежить від вологості у гігроскопічній області деревини;

– система макрокапілярів деревини моделюється як різні анатомічні елементи для різних порід та описується як сукупність паралельних капілярів різних радіусів у клітинних стінках деревини.

Для розроблення математичної моделі записуємо повний тензор напружень для гетерогенного середовища σ^{kl} у вигляді суми усереднених напружень у фазах

$$\sigma^{kl} = c_T \langle \sigma_T^{kl} \rangle_T + c_P \langle \sigma_P^{kl} \rangle_P + c_{II} \langle \sigma_{II}^{kl} \rangle_{II}, \quad (1)$$

$$\text{де } \langle \sigma_T^{kl} \rangle_T = \frac{1}{dV} \int_{dV_T} \sigma_T^{kl} d'V, \quad \langle \sigma_P^{kl} \rangle_P = \frac{1}{dV} \int_{dV_P} \sigma_P^{kl} d'V, \quad \langle \sigma_{II}^{kl} \rangle_{II} = \frac{1}{dV} \int_{dV_{II}} \sigma_{II}^{kl} d'V, \quad c_T = dV_T/dV;$$

$c_P = dV_P/dV$; $c_{II} = dV_{II}/dV$ – об'ємні концентрації твердої (Т), рідкої (Р) і парогазової (II) фаз; V_T, V_P, V_{II} – відповідні їх об'єми; верхні індекси, зокрема k і l позначають компоненти тензора напружень.

Згідно з другим допущенням, можемо записати

$$\varepsilon_T^{ik} - \varepsilon_{T0}^{ik} = 1 - c_T/c_{T0}; \quad k = 1, 2, 3, \quad (2)$$

де ε_{T0}^{ik} , c_{T0} – компоненти деформації та об'ємна концентрація у початковий момент часу.

Приймаємо, що деформації твердої фази ε_T^{ij} складаються з деформацій деревної речовини $\langle \varepsilon_T^{ij} \rangle_T$ і фіктивних деформацій ε_Φ^{ij} , які зумовлюють перебудову капілярно-пористої системи деревини, тобто $\varepsilon_T^{ij} = \langle \varepsilon_T^{ij} \rangle_T + \varepsilon_\Phi^{ij}$.

Для побудови тензора фіктивних напружень[9,11], що відповідає тензору фіктивних деформацій ε_Φ^{ij} , користуємось прийнятими фізичними допущеннями. Тоді структуру деревини розглядаємо як середовище з подвійною пористістю[12]. Тверда фаза із системою мікрокапілярів складає матрицю матеріалу. Для листяних порід судини, волокна лібриформи, серцевинні промені можна віднести до системи мікропор. Для хвойних порід мікрокапіляри у клітинних стінках моделюються як вкладене пористе середовище. У подальшому скористаємось методами дослідження таких систем [9,12].

Тоді повні усередненні напруження σ^{ij} представляємо у вигляді

$$\sigma^{ij} = (1 - c_{MK}) \langle \sigma_{KC}^{ij} \rangle_{KC} + c_{MK} \langle \sigma_{MK}^{ij} \rangle_{MK}, \quad (3)$$

де $\langle \sigma_{KC}^{ij} \rangle_{KC}$, $\langle \sigma_{MK}^{ij} \rangle_{MK}$ – усереднені компоненти напружень у клітинних стінках і макропорах, c_{MK} – об'ємний вміст пор у деревині.

Величини $\langle \sigma_{KC}^{ij} \rangle_{KC}$ записуємо аналогічно

$$\langle \sigma_{KC}^{ij} \rangle_{KC} = (1 - c_K) \langle \sigma_T^{ij} \rangle_T + c_K \langle \sigma_K^{ij} \rangle_K, \quad (4)$$

де $\langle \sigma_K^{ij} \rangle_K$ – усередненні напруження у капілярах клітинних стінок; c_K – об'ємний вміст пор у клітинних стінках.

Для визначення питомих вкладів тепломасоперенесення у кожній фазі вважаємо, що загальна геометрична поверхня (поперечний перетин) вологої деревини дорівнює сумі поверхонь (поперечних перетинів) твердої, рідкої і пароповітряної фаз. Причому поверхня твердої фази є величина стала, а поверхні рідкої і пароповітряної фаз змінюються залежно від вологовмісту деревини. Тоді для визначення величин $\langle \sigma_K^{ij} \rangle_K$, $\langle \sigma_{MK}^{ij} \rangle_{MK}$ отримуємо:

$$\langle \sigma_{MK}^{ij} \rangle_{MK} = c_{MK}^P \langle \sigma_P^{ij} \rangle_P + c_{MK}^{\Pi} \langle \sigma_{\Pi}^{ij} \rangle_{\Pi}; \langle \sigma_K^{ij} \rangle_K = c_K^P \langle \sigma_K^{ij} \rangle_K + c_K^{\Pi} \langle \sigma_{\Pi}^{ij} \rangle_{\Pi};$$

$$c_{MK}^P + c_{MK}^{\Pi} = 1; c_K^P + c_K^{\Pi} = 1, \quad (5)$$

де c_{MK}^P , c_{MK}^{Π} ; c_K^P , c_K^{Π} – об'ємний вміст рідкої і пароповітряної фаз у матеріалах і капілярах.

Величини тисків у рідкій P_P і пароповітряній P_{Π} фазах пов'язані співвідношенням $P_P = P_{\Pi} + P_{KT}$, де P_{KT} – величина капілярного тиску залежно від вологості деревини. Для визначення P_{KT} з врахуванням допущення про циліндричну форму капілярів використано формулу Лапласа $P_{KT} \approx 2\sigma(T)/r^*$, де $\sigma = 0,07564(1 - 0,02T)$. Величина r^* характеризує дисперсність розмірів пор і визначена на основі апроксимації відомих експериментальних даних[13]

Залежно від розмірів капілярів перенесення у рідкій фазі може здійснюватися не тільки дифузійним потоком пари, але й потоком, що переноситься плівковим механізмом під дією градієнта розклинюючого тиску. Для оцінювання внеску плівкового механізму використано рівняння Дерюгіна – Нерпіна.

Отримано співвідношення для визначення фіктивного тензора напружень, що характеризує зміщення твердої фази деревини

$$\sigma_{\Phi}^{ij} = \frac{1}{1 - c_{MK}} \left\{ \sigma^{ij} - (c_{MK} c_{MK}^P + c_K^P (1 - c_{MK})) \langle \sigma_P^{ij} \rangle_P - (c_{MK} c_{MK}^{\Pi} + c_K^{\Pi} (1 - c_{MK})) \langle \sigma_{\Pi}^{ij} \rangle_{\Pi} \right\}. \quad (6)$$

Математичне моделювання зв'язку між компонентами напружень $\sigma_{ij}(\tau)$ і деформацій $\varepsilon_{ij}(\tau)$ для твердої фази (деревина скелету) з врахуванням анізотропії механічних властивостей базується на інтегральних рівняннях Больцмана – Вольтера[14,15], які доповнені залежністю всихання гігроскопічних матеріалів від вологості

$$\langle \sigma_T^{ij}(\tau) \rangle_T = C \left(\langle \varepsilon_T^{ij}(\tau) \rangle_T - \langle \varepsilon_T^U(\tau) \rangle_T \right) - C \int_0^{\tau} R(t, \tau) \left(\langle \varepsilon_T^{ij}(\tau) \rangle_T - \langle \varepsilon_T^U(\tau) \rangle_T \right) dt, \quad (7)$$

де C – тензор компонентів пружності деревної речовини; ε_T^U – вектор деформацій, зумовлений всиханням деревини; $R(t, \tau) = R^{ijkl}(t, \tau)$ – тензор ядра релаксації, за допомогою якого визначається реологічна поведінка деревини.

Аналогічно здійснено математичне моделювання зв'язку між тензорами напружень твердої фази і тензорами фіктивних деформацій.

Таким чином, отримано математичну модель деформаційно-релаксаційних процесів у деревині під час сушіння як трифазному середовищі з врахуванням анізотропії тепломеханічних характеристик

$$\begin{aligned} \varepsilon^{ij}(\tau) = & (D_T + A_{MK})\sigma^{ij} + (D_T c_{II} + A_{MK}\gamma_1) p_{II} \delta^{ij} + (D_T c_P + A_{MK}\gamma_2) p_P \delta^{ij} + \\ & + \int_0^\tau \left((D_T K(t, \tau) + A_{MK} K_\phi(t, \tau)) \sigma^{ij} + (D_T K(t, \tau) c_{II} + \right. \\ & \left. + A_{MK} K_\phi(t, \tau) \gamma_1) p_{II} \delta^{ij} + (D_T K(t, \tau) c_P + A_{MK} K_\phi(t, \tau) \gamma_2) p_P \delta^{ij} \right) dt - \beta \varepsilon_T^U. \end{aligned} \quad (8)$$

Тут введено позначення: $D_T = \Pi / c_T$; $A_{MK} = \Pi_\phi / (1 - c_{MK})$; $\gamma_1 = c_{MK} (c_{MK}^P - c_K^P) + c_K^P$; $\gamma_2 = c_{MK} (c_{MK}^\Pi - c_K^\Pi) + c_K^\Pi$; Π – тензор миттєвих податливостей, який визначається за допомогою тензора C ; Π_ϕ – фіктивний тензор податливості, що визначається за допомогою тензора пружності C_ϕ ; β – коефіцієнти тензора всихання.

Функції реологічної поведінки деревини вибираються у вигляді

$$K(t, \tau) = d_0 + \sum_{n=1}^N d_{ij} \exp\left(\frac{-\beta_{ij}(t - \tau)}{\tau_P}\right), \quad (9)$$

де коефіцієнти d_0 , d_{ij} , β_{ij} і час релаксації τ_P визначаються шляхом апроксимації відомих експериментальних даних деформацій повзучості [16,17].

Для визначення деформаційно-релаксаційних процесів за математичною моделлю (8) знайдено об'ємні концентрації фаз як для початкового стану деревини, так і з врахуванням зміни вологості. У подальшому прийнято рівномірний розподіл фаз за об'ємом деревини, що дало змогу скористатися умовами адитивності.

Об'ємний вміст пароповітряної суміші визначається за формулою[2]

$$c_{II0} = 1 - \rho_W \left(\frac{1}{\rho_T} + \frac{W}{100\rho_P} \right) \frac{100}{100 + W}. \quad (10)$$

Оскільки значення c_{T0} можна одержати на основі апроксимації експериментальних даних повітроємності деревини, а значення густини деревної твердої фази ρ_T і води ρ_B відомі[17], то з врахуванням (10) одержуємо:

$$\begin{aligned} c_{P0} = & \frac{1}{\rho_{II} - \rho_T} \left(\rho_W \left(1 + \frac{1}{\rho_T} + \frac{W}{100\rho_P} \right) \frac{100}{100 + W} - \rho_{II} \right); \\ c_{T0} = & \frac{1}{\rho_{II} - \rho_T} \left(\rho_W (\rho_{II} - \rho_T - 1) \left(1 + \frac{1}{\rho_T} \frac{W}{100\rho_P} \right) \frac{100}{100 + W} - \rho_{II} \right), \end{aligned} \quad (11)$$

де c_{T0} , c_{P0} , c_{II0} – об'ємні концентрації фаз у початковий момент часу, ρ_W – густина деревини для конкретної вологості. Її визначаємо для різних порід з густиною ρ_{12} для нормалізованої вологості за формулою [17]

$$\rho_W = \begin{cases} k_{\alpha 1} \rho_{12} \frac{100 + W}{100 + k_{\alpha 2} W}, W \leq 30\%; \\ k_{\alpha 3} \rho_{12} (1 + 0.1W), W > 30\%, \end{cases} \quad (12)$$

де $k_{\alpha 1} = 0,957$, $k_{\alpha 2} = 0,6$, $k_{\alpha 3} = 0,811$ (для акації, берези, бука, граба) і $k_{\alpha 1} = 0,946$, $k_{\alpha 2} = 0,5$, $k_{\alpha 3} = 0,823$ для інших порід.

Математична модель визначення в'язкопружного стану деревини як капілярно-пористого трифазного середовища включає рівняння рівноваги механіки гетерогенних середовищ

$$\begin{aligned} \frac{\partial (c_T \langle \sigma_{TX} \rangle_T)}{\partial x} + \frac{\partial (c_T \langle \sigma_{TX} \rangle_T)}{\partial y} + p_T \frac{\partial c_T}{\partial x} + Q_{1,2} &= 0; \\ \frac{\partial (c_T \langle \sigma_{TY} \rangle_T)}{\partial x} + \frac{\partial (c_T \langle \sigma_{TY} \rangle_T)}{\partial y} + p_T \frac{\partial c_T}{\partial y} + Q_{2,2} &= 0, \end{aligned} \quad (13)$$

де $p_T = -\frac{1}{3}\langle\sigma_T\rangle_T$, $Q_{1,2}$, $Q_{2,2}$ – складові потоків масоперенесення.

Граничні умови характеризують стан деревини у частковий момент сушіння і мають вигляд:

$$\begin{aligned} \langle\sigma_{TX}\rangle_T = 0, x = 0, x = l_1; \langle\sigma_{TY}\rangle_T = 0, y = 0, y = l_2; \\ \langle\sigma_{TXY}\rangle_T = 0, x = 0, x = l, y = 0, y = l_2, \end{aligned} \quad (14)$$

де l_1, l_2 – геометричні розміри поперечного перетину деревини.

Математичні моделі тепломасоперенесення. Математичні моделі для визначення концентрації рідини, пари, повітря і пароповітряної суміші у деревній пластині запропоновано у вигляді диференційних рівнянь вологепровідності з граничними умовами, характерними для першого та другого періодів процесу сушіння[3,18]. Базуючись на розв'язках диференційних рівнянь вологеперенесення, а також рівняннях стану газової фази і законі Дальтона, з врахуванням частки вільного від рідини об'єму матеріалу, отримано закономірності розподілу перенесення вологості, тепла та концентрації парогазової суміші у деревній пластині. Зокрема, для першого періоду процесу сушіння отримано:

- для перенесення вологи у рідкій фазі

$$\begin{aligned} U_P(\bar{x}, Fo_{UP}) = A_P (Fo_{UP} + 0,5 \cdot \bar{x}^2 - 1/6) + \frac{2A_P}{\pi^2} \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n^2} \cos(\pi n \bar{x}) \exp(-\pi^2 n^2 Fo_{UP}) + \\ + 2 \sum_{n=1}^{\infty} \cos(\pi n \bar{x}) \exp(-\pi^2 n^2 Fo_{UP}) \int_0^1 U_{0P}(z, 0) \cos(\pi n z) dz + \int_0^1 U_{0P}(z, 0) dz; \end{aligned} \quad (15)$$

- для переміщення вологи у паровій фазі

$$\begin{aligned} U_{II}(\bar{x}, Fo_{UII}) = A_{II} (Fo_{UII} + 0,5 \bar{x}^2 - 1/6) + \frac{2A_{II}}{\pi^2} \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n^2} \cos(\pi n \bar{x}) \exp(-\pi^2 n^2 Fo_{UII}) + \\ + 2 \sum_{n=1}^{\infty} \cos(\pi n \bar{x}) \exp(-\pi^2 n^2 Fo_{UII}) \int_0^1 U_{0II}(z, 0) \cos(\pi n z) dz + \int_0^1 U_{0II}(z, 0) dz, \end{aligned} \quad (16)$$

де $\bar{x} = x/l$; Fo_{UP} , Fo_{UII} – масообмінні критерії Фур'є; A_P , A_{II} – величини, залежні від характеристик масообміну, початкових значень розподілу вологи U_{0P} , U_{0II} у рідкій і паровій фазах; l – геометричний розмір.

Аналогічно отримано математичні моделі для визначення перенесення вологи у рідкій і паровій фазах та перенесення повітря і пароповітряної суміші для другого періоду процесу сушіння. Зокрема:

- для перенесення вологи у рідкій фазі:

$$\begin{aligned} U_P(\bar{x}, Fo_{mp}) = 2 \sum_{n=1}^{\infty} \cos\left(\left(2n-1\right) \frac{\pi \bar{x}}{2}\right) \exp\left(-\left(\left(2n-1\right) \frac{\pi}{2}\right)^2 Fo_{UP}\right) \times \\ \times \int_0^1 U_{0P}(z, 0) \cos\left(\left(\left(2n-1\right) \frac{\pi}{2}\right) z\right) dz; \end{aligned} \quad (17)$$

- для перенесення вологи у паровій фазі

$$\begin{aligned} U_{II}(\bar{x}, Fo_{UII}) = 2 \sum_{n=1}^{\infty} \cos\left(\left(2n-1\right) \frac{\pi \bar{x}}{2}\right) \exp\left(-\left(\left(2n-1\right) \frac{\pi}{2}\right)^2 Fo_{UII}\right) \times \\ \times \int_0^1 U_{0II}(z, 0) \cos\left(\left(\left(2n-1\right) \frac{\pi}{2}\right) z\right) dz, \end{aligned} \quad (18)$$

- для знаходження температурного поля

$$\begin{aligned}
 T(\bar{x}, Fo) = & 2 \sum_{n=1}^{\infty} \frac{\cos \mu_n \bar{x} (Bi \sin \mu_n + \mu_n \cos \mu_n)}{(Bi + 1) \sin \mu_n + \mu_n \cos \mu_n} \exp(-\mu_n^2 Fo) \int_0^1 T_0(z) \cos \mu_n z dz + \\
 & + 2 \sum_{n=1}^{\infty} \frac{\cos \mu_n \bar{x} (Bi \sin \mu_n + \mu_n \cos \mu_n)}{\mu_n^2 [(Bi + 1) \sin \mu_n + \mu_n \cos \mu_n]} \exp(-\mu_n^2 Fo) \int_0^1 Po(z) \cos \mu_n z dz + \\
 & + \left[\frac{1}{Bi} - 2 \sum_{n=1}^{\infty} \frac{\cos(\mu_n \bar{x})}{\mu_n [(Bi + 1) \sin \mu_n + \mu_n \cos \mu_n]} \exp(-\mu_n Fo) \right] Ki(Fo),
 \end{aligned} \tag{19}$$

де Fo , Bi , Ki , Po – теплообмінні критерії Фур'є, Біо, Кірпічова, Померанцева; μ_n – корені характеристичного рівняння $ctg \mu_n = \mu_n / Bi$.

Прийнято допущення про те, що тиск водяної пари на поверхні деревини визначається з врахуванням середньої вологості сушіння матеріалу та рівноважної вологості повітря, а тиск водяної пари у середині деревини дорівнює тиску насиченої пари, що залежить від температури. Загальний тиск пароповітряної суміші у деревині визначається за законом Дальтона, а на поверхні матеріалу він дорівнює атмосферному. Моделювання впливу вологоперенесення на процес теплоперенесення здійснюється з врахуванням внутрішнього джерела у рівнянні теплоперенесення, яке описує потік випаровуючої вологи у деревині.

Алгоритмічне та прикладне програмне забезпечення У рамках об'єктно-орієнтованого підходу розроблено прикладне програмне забезпечення для чисельної реалізації отриманих математичних моделей в'язкопружного деформування деревини у процесі сушіння з врахуванням багатофазності.

Для чисельної реалізації математичної моделі (8) – (14) метод скінченних елементів (МСЕ) [4,5,19] адаптовано для в'язкопружної області деформування гетерогенного середовища. Для цього виведено еквівалентне варіаційне формулювання математичної моделі визначення в'язкопружного стану на основі використання принципу мінімуму повної потенціальної енергії. Функціонал Лагранжа, мінімальне значення якого збігається з розв'язком математичної моделі (8) – (14), остаточно записано у вигляді

$$\Lambda = \frac{1}{2} \int_V \varepsilon^T C \varepsilon dV - \frac{1}{2} \int_V \varepsilon^T C \int_0^{\tau} R(t-s, t, U) \varepsilon ds dV - \int_V \varepsilon^T C \varepsilon_U dV + \int_V \varepsilon^T C \int_0^t R(t-s, t, U) \varepsilon_U ds dV, \tag{20}$$

де C – компоненти тензора, які характеризують анізотропні пружні характеристики твердої фази (деревної речовини) та характеристики багатофазності структури.

Для знаходження основних співвідношень МСЕ використано скінченно-різницеву апроксимацію векторів переміщень $\{u(\tau)\}$ і деформації $\{\varepsilon(\tau)\}$ та функції реологічної поведінки деревини $R(\tau, \tau')$ у часі. Зокрема, для $\{\varepsilon(\tau)\}$ та ядра релаксації отримано:

$$\{\varepsilon(\tau)\} = \{\varepsilon(\tau_i)\} + \frac{\{\varepsilon(\tau_{i+1})\} - \{\varepsilon(\tau_i)\}}{\tau_{i+1} - \tau_i} (\tau - \tau_i); \tag{21}$$

$$R_i^* = \frac{\Delta \tau}{2} R_i^*(\tau_0) + \Delta \tau \sum_{j=1}^M R_i^*(\tau_j) + \frac{\Delta \tau}{2} R_i^*(\tau_M). \tag{22}$$

З умови мінімуму функціонала Лагранжа отримано систему алгебраїчних рівнянь для знаходження невідомих переміщень на кожному часовому кроці $\Delta \tau_i$ ($i = \overline{1, M}$, де M – кількість часових інтервалів):

$$\begin{aligned}
 & \sum_{n=1}^N U_n^k \left[\frac{1}{2} \int_{V_n} B^T C B \varphi_e(x) dV \right] + \sum_{n=1}^N U_n^k \left[\frac{1}{4} \int_{V_e} B^T C R(s_k, \tau_k) B \varphi_n(x) dV \right] = \\
 & = \sum_{n=1}^{k-1} \sum_{n=1}^N U_n^j \left[\frac{\Delta \tau}{2} \int_V B^T C R(s_k, \tau_j) B \varphi_n(x) dV \right] + \sum_{n=1}^N \left[\int_{V_n} B^T C \varepsilon_U^k d\Omega \right] - \\
 & - \sum_{n=1}^{k-1} \sum_{n=1}^N \left[\frac{\Delta \tau}{2} \int_{V_e} B^T C (R(s_k, \tau_j)) \varepsilon_U^j + R(s_k, \tau_{i+1}) \varepsilon_U^{i+1} \right] dV, k = 1, \dots, N.
 \end{aligned} \tag{23}$$

Програмна реалізація методу скінченних елементів на основі об'єктно-орієнтовного підходу полягала у розробці пакетів класів та відношень між ними[20,21]. На окремі пакети розділено класи, які відображають сутність об'єктно-орієнтовної реалізації МСЕ та реалізують: температурні та вологісні коефіцієнти, які містить задача визначення потоків масоперенесення у рідкій, твердій та газоподібній фазах (вони описані у вигляді функцій, що залежать від температури, вологості та інших аргументів); коефіцієнти, необхідні для розв'язування задачі в'язкопружності, які також обчислюють залежно від температури та вологовмісту матеріалу; параметри зовнішнього середовища, а саме температуру середовища t_c , відносну вологість φ (виокремленні в окремий інтерфейс); початкові значення температури t_0 , вологовмісту u_0 ,

компонент переміщень \vec{U} та напружень σ , а також геометричні розміри матеріалу (l_1 і l_2) та тривалість процесу τ ; параметри чисельного розв'язування, такі як кількість розбиттів за часом, кількість розбиттів за координатними осями, порядок квадратурних формул для обчислення інтегралів, тощо. Класи орієнтовані на конкретні обчислення матричної та векторної алгебри для розв'язання систем алгебраїчних рівнянь; інтерфейс користувача.

Розроблено програмно-аналітичний комплекс скінченно-елементного аналізу деформаційно-релаксаційних полів у процесі сушіння капілярно-пористих тіл як багатофазних середовищ з врахуванням анізотропії в'язкопружних властивостей матеріалу, механізмів переродження залишкових деформацій.

Досліджено адекватність математичної моделі та порядок збіжності алгоритму МСЕ для в'язкопружної області деформування капілярно-пористих матеріалів. На основі статистичного критерію на базі коефіцієнта кореляції показано задовільну збіжність чисельного моделювання з відомими результатами, отриманими для часткових випадків.

Аналіз чисельних експериментів. Для проведення чисельних експериментів дослідження динаміки зміни температури, вологовмісту, концентрації рідини, пари та пароповітряної суміші і в'язкопружного деформування у процесі сушіння опрацьовано експериментальні дані. Уточнено значення деяких теплофізичних характеристик деревини, зокрема коефіцієнта вологопровідності як функції від температури і вологості: $a_{m1}(T, U) = a_{mt}(T) a_{mu}(U)$, $a_{m1}/a_{m2} = 1,25$. Для визначення коефіцієнта вологообміну використано залежність[22]

$\alpha = 0,95 \left(T / \varphi \varepsilon \exp(-2\sigma V_p / rTR) \right) 10^{-9}$, де V_p , σ – молярний об'єм та поверхневий натяг рідини, φ – відносна вологість середовища. Значення $r = r(U)$ отримано шляхом моделювання структури деревини як системи непостійних капілярів радіуса r , який залежить від вологості. У чисельних експериментах приймалися такі значення фізичних параметрів: для повітряної та парогазової фази[2,3,8]: $c_0 = 9,05 \cdot 10^2$ Дж/(кг·К); $\alpha_{II} = 3,3 \cdot 10^{-4}$ Вт/(м²·К); $\varepsilon_{II} = 284$ Дж/(кг·К); $R_{III} = 8,3144$ Дж/(моль·К); $\varepsilon_{III} = 461,9$ Дж/(кг·К); $\lambda_{III} = 0,0248$ Вт/(м·К); $c_{III} = 2,034 \cdot 10^3$ Дж/(кг·К); для рідкої фази: $\rho_p = 10^3$ кг/м³; $\lambda_p = 0,648$ Вт/(м·К); $c_p = 4,2 \cdot 10^3$ Дж/(кг·К); $\alpha_p = 6 \cdot 10^{-5}$ Вт/(м²·К); для твердої фази: $\rho_T = 1540$ кг/м³; $\lambda_T = 0,3$ Вт/(м·°C); $c_T = 3,7 \cdot 10^{-3}$ Дж/(кг·К); $\alpha_T = 1,66 \cdot 10^{-3}$ Вт/(м²·К).

На рис. 1 показано зміну об'ємного вмісту фаз деревини сосни залежно від вологості, а рис. 2 характеризує зміну в часі об'ємного вмісту рідкої фази. Необхідно зазначити, що відмінність розподілу температурних полів з плином тривалості сушіння деревини посилюється, а саме температура твердої фази зростає, а збільшення температури рідкої фази сповільнюється, і вона не перевищує температури насиченої пари.

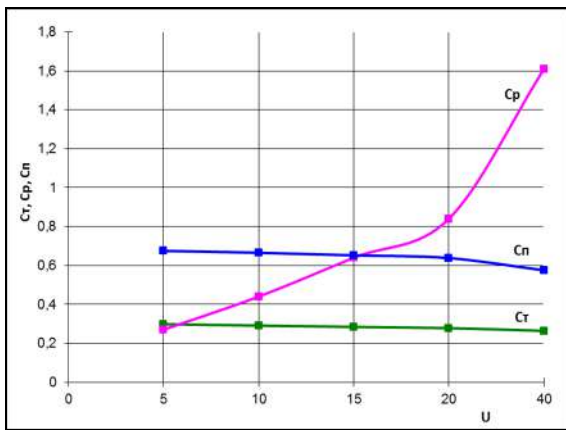


Рис.1. Розрахункові значення об'ємного вмісту фаз для деревини сосни

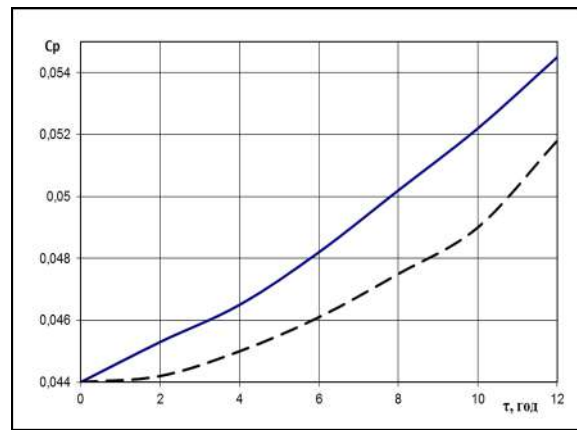


Рис.2. Зміна у часі об'ємного вмісту рідкої фази (1 – на поверхні, 2 – у деревині)

Аналіз графічних залежностей розподілу вологовмісту і температури у деревній пластині (рис. 3 та рис. 4) свідчить про те, що, незважаючи на більші значення температури у твердій фазі порівняно з рідинною, інтенсивність досягнення рівномірних значень у процесі сушіння у рідкій фазі є вищою, ніж у твердій. Такий взаємопротилежний розподіл значень вологовмісту і температури та швидкості їх зміни у різних фазах зумовлюється вищою температуропровідністю води порівняно із зовнішнім теплообміном твердої фази.

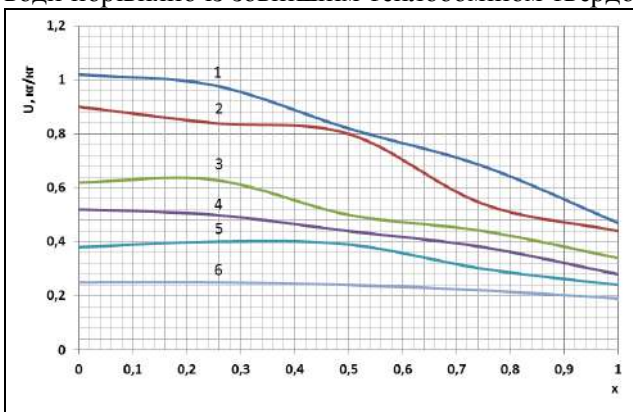


Рис.3. Розподіл вологовмісту твердої фази у деревній пластині для різних значень часу (крива 1 – 10 год; 2 – 20 год; 3 – 30 год; 4 – 40 год; 5 – 50 год; 6 – 60 год)

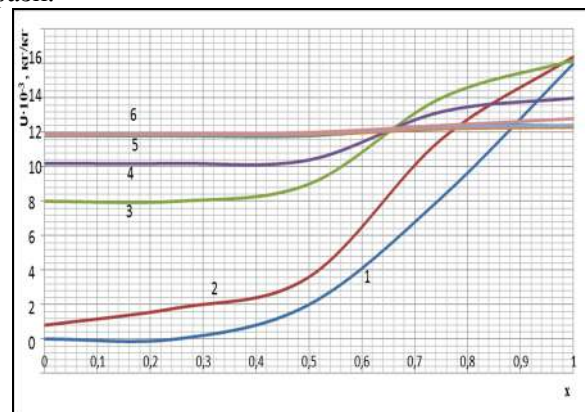


Рис.4. Розподіл вологовмісту пароповітряної фази у деревній пластині для різних значень часу (крива 1 – 10 год; 2 – 20 год; 3 – 30 год; 4 – 40 год; 5 – 50 год; 6 – 60 год)

Температура у газовій фазі досягає значень, які на порядок вищі, ніж в інших фазах. Окрім цього, інтенсивність зміни парогазової та рідкої фаз суттєво змінюється у процесі зневоднення деревини. Спостерігається значний вплив структурної анізотропії деревини на ці процеси. На початковій стадії процесу для взірців пиломатеріалів радіального напрямку значення парогазової суміші збільшується від центральної частини до поверхні. Для тангенціальних взірців розподіл парогазової суміші є більш рівномірним. Інтенсивність фазових переходів навіть для початкових етапів зневоднення деревини неоднакова у різних точках деревини і суттєво залежить від тиску парогазової суміші.

Аналіз розподілу температури і перенесення вологи у рідкій, твердій і повітряній фазах свідчить про те, що математичні моделі дозволяють прогнозувати особливості взаємопов'язаних процесів перенесення у різних фазах і враховувати фізичну нелінійність цих процесів, зумовлену залежністю фізичних властивостей деревини від температури і вологості.

Висновки

Побудовано математичну модель реологічної поведінки деревини як трифазного середовища з врахуванням анізотропії тепломеханічних характеристик, яка дає змогу враховувати пружні і

в'язкопружні та залишкові деформації деревини залежно від зміни капілярно-пористої структури матеріалу.

Сформульовано математичну модель тепломасоперенесення для періодів сталої і спадаючої швидкості сушіння капілярно-пористих матеріалів у виді взаємозв'язаних крайових задач з граничними умовами першого і третього родів, яка дає змогу знаходити залежності перенесення тепла, вологи, пари, повітря і пароповітряної суміші та прогнозувати їхню зміну для періодів процесу сушіння деревини.

Розроблено прикладне програмне забезпечення для чисельної реалізації математичних моделей на основі адаптації методу скінченних елементів для в'язкопружної області деформування багатозв'язного середовища зі змінними вологісними полями. Об'єктно-орієнтоване програмне забезпечення базується на задокументованих класах, що уможливує їх повторне використання у рамках компонентного підходу для розроблення інших прикладних програмних засобів.

1. Білей П. В. Тепломасообмінні процеси деревообробки / П. В. Білей, І. В. Петришак. – Львів : ЗУКЦ, 2013. – 376 с.
2. Серговский П. С. Гидротермическая обработка и консервирование древесины / Серговский П. С. – М. : Лесн. пром-сть, 1981. – 304 с.
3. Шубин Г. С. Сушка и тепловая обработка древесины / Шубин Г. С. – М. : Лесная промышленность, 1990. – 336 с.
4. Соколовський Я. І. Математичне моделювання двовимірного в'язкопружного стану деревини у процесі сушіння / Я. І. Соколовський, М. В. Дендюк // Фізико-математичне моделювання та інформаційні технології. – Львів, 2008. – Вип. 7. – С. 17-26.
5. Соколовський Я. І. Моделювання та оптимізація технологічних режимів сушіння деревини / Я. І. Соколовський, А. В. Бакалець // Вісник Національного університету "Львівська політехніка" : Комп'ютерні науки та інформаційні технології. – Львів, 2008. – Вип. 629. – С. 105-111.
6. Гайвась Б. І. Вплив режиму сушильного агента на осушення пористих тіл / Б. І. Гайвась, І. Б. Борецька // Комп'ютерні технології друкарства : зб. наук. праць УАД. – Львів, 2011. – №26. – С. 102-113.
7. Гринчик Н. Н. Процессы переноса в пористых средах, электролитах и мембранах / Гринчик Н. Н. Минск : Институт тепло- и массообмена АН Беларуси, 1991. – 251 с.
8. Богданов В. С. Справочник по сушке древесины / В. С. Богданов, В. А. Козлов, В. Б. Кунтуш. – М. : Лесн. пром-сть, 1990. – 304 с.
9. Нигматулин Р. И. Основы механики гетерогенных сред / Нигматулин Р. И. – М. : Наука, 1987. – 462 с.
10. Чудинов Б. С. Вода в древесине / Чудинов Б. – Новосибирск : Наука, 1984. – 270 с.
11. Николаевский В. Н. Механика пористых и трещиноватых сред / Николаевский В. Н. – М. : Недра, 1984. – 232 с.
12. Дорняк О. Р. Математическое моделирование, компьютерная оптимизация технологий, параметров оборудования и систем лесного комплекса / Дорняк О. Р. // Межв. сборн. назн. тр. ВГЛТА. – Воронеж, 2001. – С. 132-139.
13. John F. Sian. Wood: influence of moisture on physical properties / John F. Sian. – Virginia, 1995. – 227 p.
14. Bodic J. Mechanics of Wood and Composites / Bodic J., Jayne A. – New York : Van Nostrand Reinhold, 1982. – 712 p.
15. Можаровський М. С. Теорія пружності, пластичності і повзучості / Можаровський М. С. – К. : Вища школа. – 2002. – 312 с.
16. Соколовський Я. І. Методика та результати експериментальних досліджень реологічної поведінки деревини / Соколовський Я. І., Андрашек Й. В. // Науковий вісник УкрДЛТУ України : зб. наук.-техн. праць. – Львів, 1999. – Вип. 9.13. – С. 15-26.
17. Уголев Б. Н. Древесиноведение с основами лесного товароведения : учеб. для лесотехн. вузов / Б. Н. Уголев ; М-во образования Рос. Федерации, Моск. гос. ун-т леса. – Изд. 3-е, перераб. и доп. – М. : МГУЛ, 2002. – 340 с.
18. Лыков А. В. Теория сушки / Лыков А. В. – М. : Энергия, 1968. – 472 с.
19. Сегерлинд Л. Применение метода конечных элементов / Сегерлинд Л. – М. : Мир, 1979. – 378 с.
20. Соколовський Я. І. Об'єктно-орієнтована реалізація методу скінчених елементів для розрахунку в'язкопружного стану капілярно-пористих матеріалів / Соколовський Я. І., Мокрицька О. В. // Вісник національного університету "Львівська політехніка" : Комп'ютерні науки та інформаційні технології. – Львів, 2011. – Вип. 710. – С. 181-188.
21. Li X. Finite element method and constitutive modeling and computation for unsaturated soils / X. Li, H. R. Thomas // Comput. Meth. Appl. Mech. and Eng. – 1999. – V. 169. – No 1-2. – P. 135-169.
22. Гороховский А. Г. Повышение эффективности управления процессом сушки пиломатериалов / А. Г. Гороховский – Екатеринбург: УГЛТУ, 2008. – 128 с.

УДК 004:338.48

С.А. Яремко, В.В. Коваленко

Вінницький торговельно-економічний інститут КНТЕУ

ДОСЛІДЖЕННЯ ПРОБЛЕМ ВПРОВАДЖЕННЯ СУЧАСНИХ ІНФОРМАЦІЙНИХ СИСТЕМ АУДИТУ В КОНТРОЛЬНО-РЕВІЗІЙНІЙ ДІЯЛЬНОСТІ

Яремко С.А., Коваленко В.В. Дослідження проблем впровадження сучасних інформаційних систем аудиту в контрольно-ревізійній діяльності. У статті досліджено проблеми впровадження інформаційних систем і технологій в аудиті. Розглянуто спеціалізоване програмне забезпечення, яке використовується в аудиті, його види та проведена порівняльна характеристика параметрів сучасних інформаційних систем фінансового контролю, що дозволило розробити рекомендації щодо підвищення якості аудиторських послуг на основі ефективних інформаційних технологій і систем.

Ключові слова: аудит, аудитор, аудиторське програмне забезпечення, автоматизована обробка, програма аудиту.

Табл. 1. Рис.1.

Яремко С.А., Коваленко В.В. Исследование проблем внедрения современных информационных систем аудита в контрольно-ревизионной деятельности. В статье исследованы проблемы внедрения информационных систем и технологий в аудите. Рассмотрено специализированное программное обеспечение, которое используется в аудите, его виды и проведена сравнительная характеристика параметров современных информационных систем финансового контроля, что позволило разработать рекомендации по повышению качества аудиторских услуг на основе эффективных информационных технологий и систем.

Ключевые слова: аудит, аудитор, аудиторское программное обеспечение, автоматизированная обработка, программа аудита.

Yaremko S.A., Kovalenko V.V. Studies of the introduction of modern information systems audit control and audit activities. This paper investigates the problem of implementation of information systems and technologies in audit. A specialized software that is used in the audit, its types and the comparative characteristic parameters of modern information systems audit, which allowed to develop recommendations to improve the quality of audit services based on effective information technology.

Keywords: audit, auditor, auditing software, automated processing, audit program.

Постановка проблеми. Сучасний етап розвитку ринкової економіки характеризується зростанням кількості та складності господарських операцій. Тому необхідність використання аудиторськими сучасних інформаційних систем для обробки фактографічної інформації як в ході перевірки, так і при виконанні аналітичних процедур є актуальною задачею досліджень. Автоматизована обробка дозволяє швидко та точно виявляти відхилення і диспропорції показників, більш ефективно планувати і проводити сам процес аудиторської перевірки.

Впровадження інформаційних систем як внутрішнього, так і зовнішнього аудиту є гарантом розвитку та підвищення ефективності управління підприємством. Сьогодні важливим є пошук шляхів зниження витрат робочого часу аудитора та його оптимізація, що можна забезпечити шляхом використання високоякісного комп'ютерного забезпечення. Крім того, майбутнє комп'ютеризації аудиту визначає його здатність орієнтуватися на інвесторів, власників, акціонерів, а отже, задовольняти інформаційні вимоги різних напрямків аудиту.

Аналіз останніх досліджень та публікацій. Теоретичні аспекти проблем комп'ютеризації аудиту у своїх працях досліджували такі вітчизняні та зарубіжні науковці, як Ф. Ф. Бутинець, С. В. Івахненко, В. І. Подольський, В.П. Завгородній, М. Т. Білуха, Б. Е. Одинцов, А.Н. Романов, Г.В. Федорова, Е.Л. Шуремов та ін. Вони визначили теоретико-методичні підходи, що застосовують аудиторі під час комп'ютеризації, сформували основні вимоги щодо автоматизації аудиту на основі вітчизняних, а також міжнародних стандартів.

Невирішені частини проблеми. Аудит як вид діяльності і сегмент ринку ще зовсім молодий. Однак швидкий розвиток даної галузі (збільшення числа аудиторських фірм, розширення спектру надаваних ними послуг), вдосконалення і розвиток законодавчої та нормативної бази, а також майже суцільна автоматизація бухгалтерського та податкового обліку, поставили питання щодо автоматизації і в даній сфері. Працівники більшості підприємств вже не уявляють організацію своєї роботи без застосування інформаційних технологій.

Проте, проблемою залишається оптимальний вибір відповідного забезпечення для аудиту та безпосередньо застосування відповідного продукту.

Метою дослідження є підвищення ефективності та якості надання аудиторських послуг на основі дослідження основних інформаційних систем.

Основні результати дослідження. Використання комп'ютерних програм у ході аудиторської перевірки відображено в спеціальній нормативній документації. У Національному нормативі № 30 "Використання комп'ютерів в аудиті" вказується, що мета і сфера діяльності аудитора не змінюються, коли йдеться про перевірку середовища електронної обробки даних (ЕОД). Цим же нормативом регулюються й інші питання використання комп'ютерів в аудиті. Різні

варіанти використання комп'ютера відомі як "Методи аудиту за сприяння комп'ютера (МАСК)". Їх можна використовувати із залученням усіх відомих типів і конфігурацій комп'ютерів. Разом з тим, поряд зі створенням відповідного правового поля для автоматизації аудиту, на сьогоднішній день існує ряд проблем на шляху впровадження і використання інформаційних систем в аудиті. До найважливіших з них належать такі:

1. Низький рівень розвитку аудиторського ринку, викликаний незначним періодом його становлення.

2. Аудит фінансової звітності згідно з міжнародними стандартами багато в чому має за основу так зване «професійне судження аудитора», яке ґрунтується на його досвіді та інтуїції. Тобто, бездумна комп'ютеризація аудиту може призвести до такої його автоматизації, яка потягне за собою ряд помилок при проведенні аудиту.

3. Різна галузева спеціалізація клієнтів аудиторських фірм, що при динамічно мінливому законодавстві заважає детально налагодити спеціалізовані системи на усі можливі випадки.

4. Ряд методичних проблем, зокрема, необхідність розроблення методичних підходів реалізації трансформаційних процедур перетворення фінансових звітів по заданих форматах, не розголошуючи при цьому комерційну таємницю тих аудиторів, які вже виконують подібні замовлення й знають, як цю роботу виконувати.

5. Низький рівень комп'ютерної грамотності користувачів.

6. Розробка аудиторського програмного забезпечення є досить важким процесом, який потребує значних фінансових витрат [3].

Коло розробників програмного забезпечення для аудиту не таке широке, як для бухгалтерських і управлінських систем. На даний момент існують спеціалізовані програми, представлені на українському ринку програмних продуктів, орієнтованих на комплексне вирішення завдань, пов'язаних з аудиторською перевіркою [2, с. 120].

В Україні ринок аудиторського програмного забезпечення на сьогодні тільки починає розвиватися. Серед спеціалізованого аудиторського програмного забезпечення, яке представляє собою завершені програмні продукти для продажу, можна назвати такі: програмний продукт „Асистент Аудитора” (розробник фірма „Сервіс-аудит”), програмний продукт „Помічник аудитора” (фірма „Гольдберг-аудит”), програмний продукт „Abacus Professional”, програмний комплекс „ЕкспрессАудит: ПРОФ” та програмний продукт „ІТ Аудит: Аудитор” (КСБ „Мастер-Софт”).

Що стосується спеціалізованого софту, призначеного для складання програми аудиту і формування експертних висновків, то тут позицію лідера впевнено тримають розробки компаній «Майстер-Софт», «Термика» і «Гольберг-Софт».

Проведемо аналітичний огляд найбільш поширеного ліцензійного програмного забезпечення на прикладі програмних продуктів «ІТ Аудит: Аудитор», «ЕкспрессАудит: ПРОФ» і AuditXP «Комплекс Аудит» та української новинки "КІТ Аудит" .

За результатами досліджень, проведених аналітиками Інституту професійних бухгалтерів і аудиторів Росії, більшість inside-фахівців віддає перевагу програмі «ІТ Аудит: Аудитор v. 2.1.0». Основними перевагами цієї програми є можливість взаємодії з «1С: Підприємство» (версії 7.7, 8.1, 8.2) і повна сумісність функціонування з різними типами операційних систем, починаючи від традиційної XP і закінчуючи Windows 7. «ІТ Аудит: Аудитор v. 2.1.» надає користувачам цілий спектр можливостей:

- складання детального графіка аудиторських перевірок;
- формування програми аудиту;
- розстановки пріоритетів і вагових коефіцієнтів;
- аналізу даних бухгалтерського обліку.

За даними розробників, використання «ІТ Аудит: Аудитор v. 2.1.0» дозволяє скоротити термін виконання проекту на 10-20%, якщо мова йде про податковий аудит великих підприємств. У випадку з представниками малого бізнесу зниження трудомісткості діловодства не настільки відчутно.

Основний конкурент «ІТ Аудит: Аудитор v. 2.1.0» – програма «ЕкспрессАудит: ПРОФ», розроблена фахівцями консалтингового агентства «Термика». Цей ліцензійний продукт призначений для:

- розробки загальної схеми і програми аудиту;
- створення проектної документації;
- комплексного аналізу систем бухгалтерського обліку;

– підготовки аудиторського висновку.

Головна відмінність програми від конкурентів – інтегрований інформаційний базис, що складається з 2-х тисяч запитань і відповідей. Одна з переваг програми – регулярне оновлення розділів. Її суттєвий недолік – несумісність з операційною системою Linux.

Окремої уваги заслуговує і технічна розробка компанії «Гольберг-Софт» – програма AuditXP «Комплекс Аудит». Цей інформаційний продукт містить повний комплект документації, необхідної для здійснення аудиту. До переваг софту слід віднести:

- автоматичний пошук розбіжностей у фінансовій документації підприємства,
- об'єднання декількох БД в одну без використання спеціальної мережевої версії;
- можливість використання баз даних системи 1С, «Турбо Бухгалтер» [1, с. 98].

Функціональний потенціал AuditXP «Комплекс Аудит» є досить вагомим. На сьогодні її технічними напрацюваннями користується більше 1500 організацій Росії, що свідчить про ефективність даної розробки. За твердженнями творців цієї системи, будь-яка консалтингова компанія може повністю змінити програму під внутрішньокорпоративні стандарти аудиту.

Новинкою на українському ринку програм з автоматизації аудиту є система «КІТ.Аудит», розроблена командою АІС «КІТ.Аудит» в 2011 році. «КІТ.Аудит» створена для забезпечення:

- повної відповідності міжнародним та національним стандартам проведення аудиту;
- суттєвого зменшення собівартості аудиторської перевірки;
- автоматизації значної частини трудомістких робіт:
- планування та контролю витрат робочого часу персоналу;
- значного підвищення якості та прозорості робіт аудиторської фірми [3, с. 283].

«КІТ.Аудит» доступна в різних комплектаціях: легка, повна, професійна.

Для покращення аналізу представимо основні характеристики розглянутих сучасних інформаційних систем аудиту в таблиці 1.

Таблиця 1. Порівняння сучасних інформаційних систем в аудиті

Характеристика	“ІТ Аудит: Аудитор”	“Експрес Аудит: Проф”	AuditXP “Аудит звітності”	“КІТ.Аудит”
Наявність конфігурацій	ІТ Аудит: Аудитор Професіонал, ІТ Аудит: Аудитор Експрес, ІТ Аудит: Методолог	Не зазначено	Комплекс аудит, Робоче місце та робочі документи аудитора	легка, повна, професійна
Вартість продукту, грн..	3600	7300	2800	14800
Сумісність з ІС обліку та аудиту	1С:Підприємство 7.7 (8.2)	Не зазначено	1С, «Турбо Бухгалтер»	Не зазначено
Кількість робочих документів та процедур	400	Не зазначено	700	300
Автоматичне заповнення документів	+	+	+	+
Остання версія	4.1	2.2	1.2.102	1.8

Аналіз таблиці свідчить про те, що найбільш ефективною інформаційною системою для здійснення аудиту на даний час є AuditXP «Аудит звітності», адже вона має в наявності понад 700 робочих документів, сумісна з інформаційними системами обліку. Спільною перевагою усіх розглянутих інформаційних систем є автоматичне заповнення документів, що економить час аудитора під час перевірки, а також постійне оновлення версій, які є більш зручними у застосуванні.

Одним із основних показників обрання певної інформаційної системи є її вартість. На рисунку 1 представлені показники вартості найбільш поширених інформаційних систем аудиту.

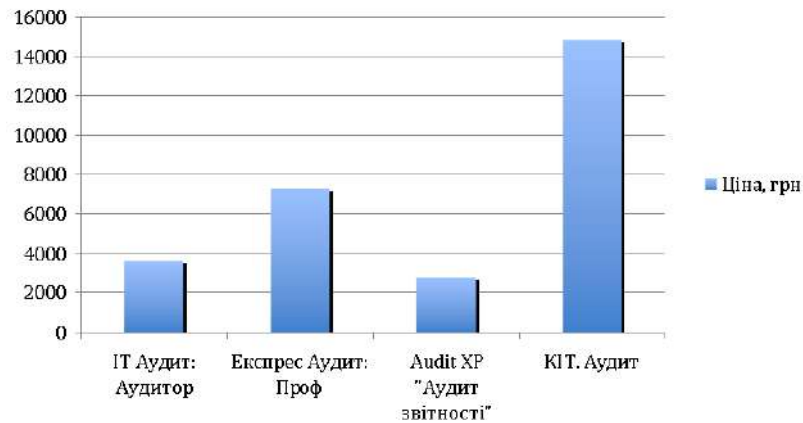


Рис. 1. Графічний аналіз вартості сучасних інформаційних систем аудиту

Як видно з рисунка 1, найбільш доступною по ціні є також інформаційна система AuditXR «Аудит звітності», на сьогоднішній день її ціна складає 2800грн. Вищу ціну має «ІТ Аудит: Аудитор» – 3600грн. А найдорожчою інформаційною системою аудиту у наш час являється «КІТ.Аудит»– 14800грн.

На основі проведеного огляду сучасних інформаційних систем, що використовуються в аудиті та їх аналізу можна запропонувати ряд рекомендацій для підвищення ефективності автоматизованих систем аудиту:

- використання шаблонів робочої документації аудитора;
- застосування методів математичного моделювання для обробки первинної інформації;
- забезпечення можливості обробки бази клієнта аудиту методами і засобами, доступними користувачу ПК;
- використання бази нормативно-довідкових документів, яка постійно оновлюється;
- проведення контрольних операцій з використанням структури дерева рішень, а не окремих блоків питань.

Висновки. Таким чином, на основі дослідження проблем автоматизації аудиту та аналітичного огляду і графічного аналізу сучасних інформаційних систем аудиту були розроблені рекомендації, що дозволять підвищити якість аудиторських послуг та їх ефективність. В цілому ж, науково обгрунтована автоматизація аудиту дозволить значно підвищити якість роботи аудитора.

1. Береза А. М. Основи створення інформаційних систем [Текст] / А. М. Береза. – 2-е вид., перероб. і допов. – К.: КНЕУ, 2001. – 214 с.
2. Гуцаленко Л.В. Аудит в Україні: розвиток, проблеми та шляхи вирішення / Л.В.Гуцаленко // Інноваційна економіка. – 2011. – №20. – С. 97-99.
3. Івахненко С.В. Комп'ютерний аудит: контрольні методики і технології. / С.В. Івахненко – К.: Знання, 2005. – 286 с.
4. Івахненко С. В. Автоматизація аудиту в Україні та світі: підходи і програмне забезпечення // Аудитор України. – 2007. – № 3. – С.19-24.
5. Інформаційні технології в обліку та аудиті. Аудит інформаційних технологій //Матеріали Міжнар. наук-практ. конф., 24-25 листопада 2006 р. – Харків, 2006. – 213 с.
6. Калініченко Л.Т. Проблемні питання щодо автоматизації аудиту в Україні / Л.Т. Калініченко // Вісник Дніпропетровського університету. – 2011. – №5. – С. 282-284.
7. Міжнародний стандарт аудиту 401. Аудит середовищ комп'ютерних інформаційних систем. – К., 2008. – 68 с.
8. Оліфіров О.В. Управлінські інформаційні системи в аналізі і аудиті. – Донецьк, 2011. – С.51-84.
9. Переход к международным стандартам бухгалтерского учета/[Электронный ресурс]. – Режим доступа <<http://consulting.ru>> .
10. Терещенко Л. О., Матієнко-Зубенко І. І., 34 Інформаційні системи і технології в обліку: Навч. посіб. — К.: КНЕУ, 2004. — 187 с.

УДК 796/799:37.034

О.П. Панасюк

Східноєвропейський національний університет імені Лесі Українки

ТЕОРЕТИЧНІ ЗАСАДИ СПРИЙНЯТТЯ ІНФОРМАЦІЇ ПРО РОЗВИТОК МОРАЛЬНИХ ЦІННОСТЕЙ

Панасюк О.П. Теоретичні засади сприйняття інформації про розвиток моральних цінностей. У даній роботі висвітлені дослідження нових підходів у формуванні моральних цінностей майбутнього вчителя фізичної культури на сучасному етапі актуалізувало проблему якісного оновлення гуманітарної сфери вищої освіти, загалом і галузі фізичної культури та спорту, що є стратегічним напрямом формування нової інноваційної ідеології в сучасній педагогічній науці. Професійна підготовка майбутнього вчителя вимагає нових підходів до розвитку моральних цінностей у особистості. Такий підхід дозволяє динамічно реагувати на нові виховні процеси, пов'язані з досягненнями актуальних досліджень, соціальної практики, підготовкою молоді засобами фізичної культури та спорту.

Ключові слова: розвиток, моральні цінності, інформація.

Panasjuk A.

ANALYSIS OF THEORETICAL BASES PERCEPTION INFORMATION ON THE DEVELOPMENT OF MORAL VALUES IN PERSONALITY

This article shows the study of new approaches in shaping the moral values of the future teachers of physical culture at present actualized quality problem updating the humanitarian sphere of higher education in general and the field of physical culture and sport, which is the formation of a new strategic direction of innovation ideology in modern pedagogical science. Training future teachers requires new approaches to moral values in the individual. This approach allows you to dynamically respond to new educational processes associated with the achievements of current research, social practices, training young people by means of physical culture and sports.

Keywords: development, moral values, information.

A.П. Панасюк

АНАЛИЗ ТЕОРЕТИЧЕСКИХ ОСНОВ ВОСПРИЯТИЕ ИНФОРМАЦИИ О РАЗВИТИИ МОРАЛЬНЫХ ЦЕННОСТЕЙ В ЛИЧНОСТИ

В данной работе освещены исследования новых подходов в формировании нравственных ценностей будущего учителя физической культуры на современном этапе актуализировало проблему качественного обновления гуманитарной сферы высшего образования, в целом и отрасли физической культуры и спорта, является стратегическим направлением формирования новой инновационной идеологии в современной педагогической науке. Профессиональная подготовка будущего учителя требует новых подходов к развитию нравственных ценностей у личности. Такой подход позволяет динамично реагировать на новые воспитательные процессы, связанные с достижениями актуальных исследований, социальной практики, подготовкой молодежи средствами физической культуры и спорта.

Ключевые слова: развитие, моральные ценности, информация.

Постановка проблеми у загальному вигляді. Аналізуючи теоретичні засади наукової проблеми виявили, що академік Бех І.Д. акцентує увагу на тому, як виховний процес у своєму історичному поступі завжди певним чином був пов'язаний з тією філософською проблематикою, яка осмислювала на відповідному світоглядному рівні ставлення особистості до світу, сутність культури, вихідні цінності буття людини в світі, співвідношення її свободи і відповідальності, форми і типи менталітету студента. Характер вирішення цієї проблематики визначав особливості виховних процесів, на основі яких розвивалась сучасна інформація про моральні цінності майбутнього вчителя фізичної культури.

Дефініція "інформація" адаптована нами від французького Information - надаю форму, створює уявлення про щось, зображає - як повідомлення про яку-небудь подію, чийось діяльність, акцентує увагу на відомостях, що є об'єктом збереження, накопичення, переробки і передачі комунікації.

Інформація - комунікаційна інфраструктура суспільства, яка визначає можливості людей своєчасно одержувати, передавати й використовувати необхідні відомості, а потім оперативно здійснювати ті чи інші інформаційні комунікації [1].

Аналіз останніх досліджень та публікацій, в яких започатковано вирішення даної проблеми. Здійснюючи науковий аналіз інформації про формування моральних цінностей у студентів факультетів фізичної культури, ми спирались на пізнання аспектів даної проблеми через роль відчуттів особистості, її компонентів сприйняття та педагогічного мислення. Функцією педагогічного мислення можна назвати розширення меж пізнання про моральні цінності за допомогою виходу за межі чуттєвого сприйняття об'єктивної дійсності. Реальне мислення особистості дає можливість за допомогою рефлексії (умовивід, осмислення) розібратися в суті моральних цінностей, встановити істинну інформацію про досліджувану проблему, що безпосереднім чином не дається нам у сприйнятті.

Наукові знання про моральні цінності майбутні вчителі фізичної культури одержують не тільки через відчуття, а і через сприйняття. У відчуттях відображаються лише окремі властивості моральних цінностей, наприклад їх зовнішній вплив (гімн України, конституція України,

спортивні змагання, туризм) через слухові, зорові аналізатори та ін., тоді як у сприйнятті всі компоненти досліджуваної проблеми подаються у їх сукупності і взаємозв'язку.

Сприйняттям називається відображення у свідомості студента предметів і явищ дійсності при їх безпосередньому впливові на органи чуття. Результатом сприйняття є цілісний образ моральних цінностей. Наприклад, взявши в руку нову книгу, ми одночасно відображаємо колір її сторінок, вагу, запах типографської фарби, тощо. Всі зорові, тактильні, нюхові та інші відчуття поєднуючись дають образ книги як одного із елементів моральних цінностей для особистості.

Здійснений нами аналіз наукових першоджерел показує, що сприйняття інформації про моральні цінності не зводиться до простої сумачії відчуттів, а складає якісно новий ступінь пізнання об'єктивної дійсності.

Історичні аспекти моральних цінностей знайшли своє відображення у наукових поглядах українських філософів П.Юркевича, Г. Кониського, Г.Сковороди. На початку ХХ ст., у зв'язку з процесами інтеграції особливий науковий інтерес почали викликати проблеми, пов'язані з впливом моральних цінностей на розвиток і поведінку окремої особистості. Ці аспекти досліджували зарубіжні вчені: У.Мак Даугалл, Г.Тард, С.Сигеля, Ф.Олпорт.

Основні засади вдосконалення процесу фахової підготовки особистості аргументовані у працях видатних педагогів К. Ушинського, П. Лесгафта, П. Блонського, С. Шацького, А. Макаренка, В. Сухомлинського, М. Стельмаховича. Окремі напрями підготовки майбутніх вчителів досліджували О. Абдуліна, І.Д. Бех, Є. Білозерцев, І.Зязюн, М. Євтух, О. Дубасенюк, А. Капська, Л. Міщик, В. Мадзігон, О. Мороз, Н. Ничкало, Г. Троцько, Є.Пристапа, Б.Шиян та ін.

Формулювання цілей статті. Мета статті полягає в обґрунтуванні наукової проблеми, сприйнятті інформації про сутність формування моральних цінностей у ході навчально-виховної роботи та розвитку культурологічних поглядів студентів в позааудиторній діяльності.

Виклад основного матеріалу. Наукове обґрунтування технологій професійної підготовки особистості через різноманітні форми навчально-виховної діяльності майбутнього вчителя, розкривається через підбір запропонованих управлінських ідей у фаховій моральній спрямованості студентів вищого навчального закладу.

Здійснюючи науковий пошук, ми намагались дослідити засоби діагностики динаміки стану професійної підготовки особистості; відбір обґрунтованих моделей позааудиторної діяльності студентів для розвитку моральних цінностей студента.

Дисертант виходив з того, що у навчальному процесі та позааудиторній діяльності варто розвивати професійне мислення на основі нових освітніх технологій, ефективно здійснювати управління професійно-моральною підготовкою студентів через різноманітні форми навчально-пізнавальної діяльності особистості [8; 11; 3].

Ми встановили, що необхідність теоретичного обґрунтування ролі педагогічних явищ у навчальній та позааудиторній діяльності студентів впливає на сприйняття інформації про сутність практичних матеріальних і духовних надбань суспільства, які відображають історично досягнутий рівень розвитку сучасної науки у відповідності до вимог Болонського процесу, які втілюються в результати продуктивної діяльності, що суттєво відображає технологію професійної підготовки особистості [4].

Проведення дослідження, здійснення констатувального та формувального експериментів, мета яких довести або спростувати положення науково-пошукової роботи про те, що сконструйована нами система управління професійно-моральною підготовкою студентів сприяє активізації засвоєння духовних знань майбутніми бакалаврами, магістрами, підвищує ефективність педагогічної діяльності викладачів у процесі реалізації нормативних курсів через різноманітні форми навчально-виховної діяльності особистості.

Встановили, що управлінський аспект освітньої технології (англ. "an educational technology") часто називають напрямком закордонної підготовки, що має на меті підвищення ефективності освітнього процесу, який гарантує досягнення позитивного результату в професійно-моральній підготовці майбутнього вчителя сучасної школи [6; 9].

Термін "технологія" визначається як наука техніки, отже управління освітньою технологією - це наука про організацію техніки професійної освіти, тобто духовна підготовка студентів має передбачити одержання системи моральних знань з кожного курсу (дисципліни), а сприйняття моральних цінностей - розкриває формування сучасних моральних поглядів та моральних знань, які займають провідне місце в системі професійної підготовки студентів.

Отже, в нашому дослідженні у поняття технологія управління фаховою підготовкою ми вкладаємо комплекс ланок, який включає: прогнозування запланованих результатів в моральній

підготовці особистості; засоби діагностики динаміки стану професійної підготовки; підбір обґрунтованих моделей навчання; критерії вибору оптимальної моделі для одержання конкретних умов у розвитку морального мислення особистості [10].

У процесі дослідження ми використали новітні підходи до розуміння наукових термінів у сучасній педагогічній науці і використовуватимемо їх у ході науково-пошукової роботи через адаптування їх змісту.

Науковий термін "виховальні відносини" - відносини, орієнтовані на повноцінний розвиток кожної особистості, здатної на активну роль у збереженні і примноженні соціальних цінностей і до співжиття в соціумі за моральними принципами [2, 5].

Виховання - це усвідомлене і цілеспрямоване зрощування людини як особистості у відповідності зі специфікою цілей, соціальних груп і організацій, в яких воно здійснюється. На відміну від соціалізації - процесу неперервного, виховання - процес дискретний, перервний, здійснюється планомірно, обмежений місцем і часом [7, 4].

Виховна технологія - науково обґрунтований засіб, що містить необхідні й достатні умови для розвитку та реалізації особистісного потенціалу виховання, свідомого залучення його до системи вироблених людством цінностей. Умови, що мають забезпечувати повноцінний морально-духовний розвиток особистості трактується як створення педагогом моделі організації виховного процесу з послідовною орієнтацією на певну систему цілей, доцільну послідовність у їх досягненні та на психологічно виражену гуманістичну суб'єкт-суб'єктну взаємодію [10, 11].

У дослідженні ми подаємо виховне середовище - як середовище безпосереднього і опосередкованого впливу на учня на мікрорівні; сукупність об'єктивних і суб'єктивних факторів, що створює умови життєдіяльності особистості, передачі їй суспільно-історичного досвіду людства і національної культури, впливає на формування її фізичних, психічних і соціально-адаптивних можливостей, процес і повноту творчої самореалізації [2, 10].

Моральні цінності - поняття етики, яке відображає значущість для індивіда проявів моралі як регулятора суспільно належної поведінки і є втіленням його суб'єктивного діяльно-зацікавленого ставлення до реалізації у житті моральних норм і принципів [11, 5].

Сприйняття культури мови - це дотримання усталених норм вимови, слова та форм вживання моральних побутових фраз.

У виховному процесі вищого навчального закладу культура поведінки - це дотримання основних вимог і правил людського співжиття, вміння знаходити правильний тон у спілкуванні з оточуючими, зокрема з учнями та молоддю.

На основі поданих вище наукових термінів у дослідженні ми ввели поняття "культурологічні моральні погляди".

Культурологічні моральні погляди студентів - це інтерактивне об'єднання практичних, матеріальних і духовних надбань суспільства, які відображені у свідомості особистості і реалізуються через дотримання усталених норм вимови, основних вимог і правил людського співжиття та конкретної педагогічної діяльності особистості [6].

У процесі дослідження ми зробили спробу охарактеризувати у ході пошукової роботи окремі компоненти таксономії цілей поведінки особистості у побуті. Серед них виділяємо:

- Когнітивний (пізнавальний) компонент. Сюди входять цілі від запам'ятовування і відтворення вивченого матеріалу до вирішення проблеми, в ході якого необхідно переосмислити набуті моральні знання, будувати їх нові співвідношення з попередньо вивченими ідеями, методами, способами дій, включаючи створення нового бачення моральної проблеми. За даними експертних оцінок, а також аналізу наукової літератури та анкет викладачів вузу виявили, що до моральної сфери відносяться більшість цілей навчання та поведінки студентів, які висуваються нормативними документами у професійній підготовці майбутнього вчителя фізичної культури.

Ми встановили, що теоретична діяльність особистості сама по собі не приводить до змін у предметах і явищах зовнішнього світу, але дозволяє передбачати їх, завчасно прогнозувати майбутні моральні наслідки. Психологічна діяльність студента тому називається теоретичною, що дозволяє вдосконалювати результати практичних дій у процесі конкретної поведінки. Отже, теоретично-виховні види побутової діяльності займають не тільки провідне місце в інтелектуальному розвитку майбутніх фахівців, а й визначають успіх у практичній праці через моральні зобов'язання.

- Афективний (емоційно-ціннісний) компонент. До нього відносяться цілі формування морально-особистісного ставлення до явищ навколишнього світу, починаючи від простого сприйняття, інтересу, готовності реагувати на засвоєння ціннісних моральних орієнтацій

© О.П. Панасюк

і їх взаємодії та активного прояву. До цього компоненту належать також цілі, які передбачають формування інтересів і нахилів у поведінці особистості, переживання тих чи інших почуттів, формування ставлення до реальної дійсності, свідомості студентів і її прояву в побутовій та релігійній оцінці явищ.

Для розуміння теорії педагогічної технології формування моральних цінностей важливо знати певні закономірності формування нових розумових дій, уміти передбачати різні види пізнавальної побутової діяльності. Формування моральних поглядів відбувається під впливом дій зовнішніх, матеріальних, бо вони дають можливість невидимі внутрішні дії зробити видимими (доступними), показати їх зміст студентам, зробити їх зрозумілими для кожного майбутнього вчителя [4].

- Психомоторний компонент. До нього належать цілі, які зв'язані з формуванням тих чи інших видів рухомої (моторної, маніпулятивної) побутової діяльності, нервово-м'язової координації, необхідної для формування культури педагогічної праці. До цього компоненту відноситься порівняно невелика кількість цілей із загальних об'ємів навчальної мети. Серед них виконання письмових завдань, удосконалення мовлення при вивченні іноземних мов, вправи фізичного виховання та реабілітації.

Проводячи дослідження переконалися, що створення надійної, достовірної системи цілей у професійній підготовці особистості питання динамічне і складне, яке цікавить не тільки теоретиків-педагогів, а перш за все викладачів-практиків і має певні причини у формуванні позитивних моральних якостей студентів фізичної культури.

- У дослідженні передбачили, що цей аспект передбачає концентрацію зусиль на головному. Користуючись технологією формування моральних поглядів викладачі вищих закладів освіти повинні вміти не тільки виділяти і конкретизувати цілі, але й впорядковувати їх, визначати першочергові моральні завдання, порядок і перспективи майбутньої побутової навчально-виховної діяльності студента.

- Таке розуміння моральних завдань впливає на ясність і гласність спільної роботи викладача і студентів. Конкретні навчально-виховні цілі у формуванні моральних поглядів дають можливість викладачу пояснити студентам орієнтири в їх загальній побутовій навчально-виховній діяльності, обговорити разом з ними механізм їх реалізації моральних якостей особистості.

- Обраний нами підхід у дослідженні моральних цінностей дав змогу врахувати створення еталонів оцінки результатів побутової навчально-виховної діяльності. Звертання до чіткого формулювання цілей розвитку моральних поглядів, які виражаються через результати побутової та духовної діяльності, піддаються більш надійній і об'єктивній оцінці. Це дає змогу систематично формувати у студентів об'єктивні оцінні моральні навички.

Ми встановили, що головна особливість процесу засвоєння побутових моральних дій полягає у тому, що такі дії завжди є активними: моральні знання можна передати тільки тоді, коли їх беруть, тобто коли студенти виконують певну творчу діяльність, в процесі якої формуються основи культури праці та побуту, розвиваються моральні цінності [2].

Отже, процес засвоєння моральних знань - це завжди виконання певних пізнавальних дій, усвідомлення їх суті. Викладач повинен бути впевнений, що студенти володіють всією необхідною системою моральних навичок, реальних дій, які складають уміння здобувати майбутню професію. Дія - це одиниця аналізу навчально-виховної праці студента. Викладач має знати, передбачати її структуру, функціональні можливості, основні риси, етапи та закономірності становлення творчої особистості, яка володіє системою моральних знань.

Встановили тезу про те, що студенти розуміють, що будь-яка людська дія завжди спрямована на певний предмет, моральне поняття чи явище. Наприклад, студент аналізує поняття реабілітації, вивчає будову комп'ютера, норми і правила поведінки прилучається до наукової роботи у сфері оцінки моральних якостей. Отже, дія завжди цілеспрямована, і в процесі її виконання одержується певний результат у формуванні моральних цінностей. Він може збігатися з поставленою професійною метою, а може йти всупереч з нею.

У ході дослідження виявили, що мета навчально-виховної дії нерозривно пов'язана з таким важливим компонентом, як мотив моральної діяльності. Мотив спонукає студента ставити і досягати різноманітних моральних результатів, виконувати відповідні навчально-виховні дії. Коли мотиви не обґрунтовані, не усвідомлений кінцевий результат, навчання як вид професійної пізнавальної діяльності втрачає сенс.

У ході наукового пошуку виявили, що у структуру будь-якої дії входить та чи інша система операцій, за допомогою яких ця дія виконується. Так, для дії порівняння необхідно

виділити ознаку (основа для порівняння: проаналізувати слова Гімну України та Гімну Росії), на базі якої будуть порівнюватися моральні явища. Потім робиться висновок, одержаний за результатами порівняння, який дозволяє завершити пізнавальну дію у формуванні моральних якостей [9; 1].

Ми передбачили, що наступний компонент формування моральних цінностей - орієнтовна основа дії. Це система умов, на яку реально спирається студент при виконанні певних моральних дій. Важливо знати, що ця система умов педагогічної діяльності може бути подана з різних позицій. Ці педагогічні умови можуть відбивати окремі особливості конкретного морального випадку, а можуть фіксувати і загальні, суттєві для цілого класу явищ морального спрямування (аналіз тексту Конституції України).

У процесі реалізації моральних дій із засвоєння основ професійної праці у навчанні та побуті студент проявляє свою індивідуальність, яку треба враховувати в ході організації побутової навчально-виховної діяльності майбутнього вчителя (спортивні змагання).

Висновки і перспективи подальших досліджень. Природа та механізм засвоєння моральних знань з основ розвитку професійних якостей є складовою частиною технології формування моральних цінностей, основою педагогічної діяльності у сучасній школі.

Пристаючи до дослідження, ми обґрунтували методику проведення науково-пошукової роботи, розпочавши її з визначення наукового терміну самосвідомості, яка є складовою ланкою технології формування моральних цінностей. Свідомість - усвідомлення студентом себе самого як особистості, своїх фізичних і розумових здібностей, вчинків і дій, їхніх мотивів і мети, свого ставлення до зовнішнього світу, інших людей і до самого себе [11].

У процесі дослідження ми передбачили, що сприйняття кожного соціального явища має свою історію, яка охоплює процес її зародження, становлення, конкретизації, формування та реалізації. Свідомість, охоплена в генетичному аспекті, може бути визначена об'єктивно у певних межах. У такому разі вона окреслюється як явище закономірне, а не випадкове, як феномен, що має внутрішню логіку, а не щось штучно сконструйоване.

Завдання дослідження полягало в тому, щоб обґрунтувати свідомість особистості студента вищого закладу освіти в її реальних взаємозв'язках із формуванням моральних проглядів, іншими явищами суспільного життя. Тільки в такому разі воно розкривається достатньо повно.

У ході пошукової роботи ми виробили власне ставлення до розуміння проблеми свідомості особистості та спрямували обґрунтування концепції дослідження у вірному соціальному руслі. Специфіка професійної свідомості студентів вищих закладів освіти полягає у тому, що вона не лише відображає об'єкт пізнання, але допомагає ефективно формувати професійні риси особистості, її моральні погляди.

1. Андрійчук О. Я. Виховання гуманності у студентів медичного коледжу в процесі фахової підготовки. Автореф. дис... на здобуття наукового ступеня канд. под. наук. - Київ, 2003. - 20 с
2. Андрійчук О. Я., Смолюк В. І. Гуманістична освіта молоді: історико-дидактичний аспект / Монографія. - Луцьк: Видавництво «Волинська обласна друкарня», 2002. - 148 с.
3. Алексюк А. М. Методи навчання і методи учіння / Анатолій Миколайович Алексюк. - К.: Знання, 1980. - 48 с.
4. Алексюк А.М. Педагогіка вищої школи [Текст]: курс лекцій: модульне навчання: Навчальний посібник/ Анатолій Миколайович Алексюк; Мін. освіти України. Ін-т сист. досліджень освіти. Київськ. ун-т ім. Т.Шевченка; Ред. С. С. Самойлова. - Київ: ІСДО, 1993. - 220с.
5. Ананьев Б. Г. О проблемах современного человекознания / Борис Іе-расимович Ананьев. - М.: Наука, 1977. - 380 с.
6. Андреев В. И. Педагогика: Учебный курс для творческого саморазвития / Виталий Иванович Андреев. - 2-е изд. - Казань: КГУ, 2000. - 608 с.
7. Андреев В. И. Диалектика воспитания и самовоспитания творческой личности / Виталий Иванович Андреев. - Казань: КГУ, 1988. - 228 с.
8. Андреева Г. М. Социальная психология [текст]: учебник для вузов / Галина Михайловна Андреева; Гос. ком. РФ по высш. образов. Ин-т «Открытое общество»; [ред. совет: В. И. Бахмин, Я. М. Бергер, Е. Ю. Гениева и др.] - М.: Аспект Пресс, 1996. - 376 с.
9. Андреева Г. М. Современная социальная психология на Западе / Галина Михайловна Андреева, Н. Н. Богомолова, Л. А. Петровская. - М.: Свет, 1978.-233 с.
10. Андрощук А. О. Рейтингова технологія оцінки знань в навчально-виховному закладі / А. О. Андрощук // Педагогіка і психологія. - 1996. - № 3. -С.86-93.
11. Панасюк Н.Л. Теоретична модель управління якістю освіти. /Н.Л. Панасюк// Комп'ютерно-інтегровані технології: освіта, наука, виробництво: наук. журнал – Вип. 13 – Луцьк: ЛНТУ, 2014. – С. 135–140.

УДК 658.51:631.3

О.В. Сидорчук

Національний науковий центр «Інститут механізації та електрифікації сільського господарства».

П.М. Луб

Львівський національний аграрний університет.

В.В. Грабовець

Луцький національний технічний університет.

В.С. Спічак

Володимир-Волинський агротехнічний коледж

АЛГОРИТМ СТАТИСТИЧНОЇ ІМІТАЦІЙНОЇ МОДЕЛІ ҐРУНТООБРОБНО-ПОСІВНИХ ПРОЦЕСІВ ЛІТНЬО-ОСІННЬОГО ПЕРІОДУ

Сидорчук О.В., Луб П.М., Грабовець В.В., Спічак В.С. Алгоритм статистичної імітаційної моделі ґрунтообробно-посівних процесів літньо-осіннього періоду. Розкрито особливості впливу некерованих складових механізованих процесів обробітку ґрунту та сівби на календарні терміни їх виконання та функціональні показники. Розроблено блок-схему алгоритму статистичної імітаційної моделі ґрунтообробно-посівних процесів літньо-осіннього періоду.

Ключові слова: обробіток, сівба, мінливість, комплекс машин, алгоритм, моделювання.

Форм. 2. Рис. 1. Літ. 9.

Сидорчук А.В., Луб П.М., Грабовець В.В., Спічак В.С. Алгоритм статистической имитационной модели почвообрабатывающе-посевных процессов летне-осеннего периода. Раскрыты особенности влияния неуправляемых составляющих механизированных процессов возделывания почвы и посева на календарные сроки их выполнения и функциональные показатели. Разработана блок-схема алгоритма статистической имитационной модели почвообрабатывающе-посевных процессов летне-осеннего периода.

Ключевые слова: возделывание, посев, изменчивость, комплекс машин, алгоритм, моделирование.

Sydorchuk O.V., Lub P.M., Grabovets V.V., Spichak V.S. Algorithm of statistical simulation model of soil-tillage processes in the summer-autumn period. The features of out of control constituents influencing in the soil-tillage and sowing processes are exposed on the calendar terms of their implementation and functional indexes. The algorithm chart of statistical simulation model of soil-tillage and sowing processes in summer-autumn period is developed.

Key words: soil-tillage, sowing, changeability, machines complexes, algorithm, modeling.

Постановка проблеми. Об'єктивний вплив агрометеорологічних умов на стан агрофону полів сільськогосподарських підприємств (СП) зумовлює мінливість термінів виконання механізованих процесів щодо удобрення, підготовки ґрунту та сівби. Це зумовлює потребу технологічного адаптування до стану предмету праці як у розрізі весняного так і літньо-осіннього періодів. Тому для своєчасної сівби культур у якісно підготовлений та удобрений ґрунт СП повинне володіти "адаптивним" комплексом сільськогосподарських машин. Сучасні методи та моделі із дослідження ефективності таких комплексів, на жаль, ще не дають змоги врахувати системно-подієві особливості їх функціонування та потребують удосконалення.

Аналіз останніх досліджень і публікацій. Чинні методи та моделі обґрунтування параметрів комплексів машин СП [6] ґрунтуються на нормативах потреб у техніці і дають змогу встановити "базовий" комплекс машин для заданих агротехнічних термінів робіт [3]. Їх застосування для дослідження адаптивних технологічних систем [9], на жаль, не дає змоги об'єктивно оцінити комплекс машин, який функціонує в умовах що постійно змінюються, а відтак потребують корегування ходу ґрунтообробно-посівних процесів у розрізі календарного періоду.

Невирішені раніше частин проблеми. Сучасний розвиток інформаційних технологій та, зокрема потужності персональних комп'ютерів, дає змогу застосовувати складні методи дослідження механізованих процесів рільництва, а відтак обґрунтовувати параметри технологічних систем. До таких методів відноситься й статистичне імітаційне моделювання, яке дає змогу враховувати ймовірнісний вплив агрометеорологічних умов на терміни початку, тривалості та завершення відповідних робіт, а відтак отримати статистичні закономірності функціональних показників відповідних технологічних комплексів машин.

Метою статті є розкрити особливості впливу некерованих складових механізованих процесів обробітку ґрунту та сівби на календарні терміни їх виконання та функціональні показники, а також представити блок-схему алгоритму статистичної імітаційної моделі ґрунтообробно-посівних процесів літньо-осіннього періоду.

Виклад основного матеріалу дослідження. Результатом процесів обробітку ґрунту та сівби (ОГС) є поля із посіяними культурами. Вираження ефективності цих процесів можливе на підставі декількох критеріїв, основним із яких є врожайність (обсяг) вирощеної

сільськогосподарської культури. Відповідно до цього, особливістю процесів ОГС є те, що формування їх ефективності здійснюється за сукупного впливу керованих, частково керованих та некерованих елементів зовнішнього та внутрішнього середовища. Тому для встановлення функціональних показників відповідних комплексів машин необхідно враховувати вплив таких складових, як предмет праці та агрометеорологічні умови, природно дозволений фонд часу на виконання ґрунтообробно-посівних процесів, виробнича програма сівби культур (зокрема озимих), а також трудові ресурси, які, власне, сукупно й формують параметри технологічної (ТС).

Аналіз агротехнічних вимог до процесів ОГС переконує у тому, що роботи, які у них виконуються, скеровані на своєчасне задоволення вимог озимих культур до якісного стану агрофону поля на початкових фазах (проростання та появи сходів) їх вегетаційного періоду.

Створення цих сприятливих умов досягають за рахунок впливу на керовані чинники ефективності зазначених процесів, а також їх причинно-наслідкові зв'язки. На практиці це відбувається за допомогою поточного аналізу системно-подієвих тенденцій зміни предметної (агрофонові) та агрометеорологічної складових і прийняття рішень щодо виконання множини технологічних операцій із ОГС. Відповідно до цього досягнення кінцевого результату згаданих процесів здійснюється за декілька етапів якісного перетворення агрофону поля, що відбувається впродовж певного календарного проміжку часу. Згідно із технологіями ОГС етапи цього перетворення можуть бути різними, однак спільною їх ознакою є завдання, на які вони скеровані, а саме: забезпечити оптимальну щільність ґрунту, водяний та повітряний режими, сприятливі умови для органно-мінерального живлення тощо, а також розташувати добрива і насіння озимих культур на відповідній глибині [4].

Важливою особливістю цих процесів також є те, що "негативний" вплив агрометеорологічних умов на стан ґрунту зумовлює виникнення так званих непогожих проміжків, за яких фізико-механічні властивості ґрунту зумовлюють призупинення робіт на полях СГП. Унаслідок цього, скорочується тривалість природно дозволеного фонду часу (t_{nz}) на виконання множини згаданих робіт. Тривалість цього фонду часу є ймовірнісною величиною, а її статистичні характеристики залежать від низки складових. Загалом, роботи із ОГС необхідно виконати в проміжку часу між двома головними (базовими) подіями – часом (календарним терміном) завершення (τ_{33}^n) збирання врожаю попередника та часом початку (τ_{ne}) технологічних втрат озимих культур від несвоєчасного виконання згаданих процесів. Настання подій τ_{33}^n та τ_{ne} в розрізі календарного періоду виконання механізованих процесів ОГС характеризується стохастичністю, що призводить до мінливості t_{no} . Окрім того, випадання дощу між τ_{33}^n та τ_{ne} зумовлює виникнення непогожих проміжків [2], простій техніки та скорочення t_{no} .

Врахування цих об'єктивних складових під час встановлення функціональних показників технологічних комплексів машин для виконання ґрунтообробно-посівних процесів, зокрема впродовж літньо-осіннього періоду, потребує застосування методів статистичного імітаційного моделювання, а також методів системно-подієвого відображення термінів та темпів виконання множини технологічних операцій згаданих процесів. Першочерговим завданням щодо розроблення цих моделей є розроблення блок-схеми їх алгоритму (рис.).

Відповідно до теорії моделювання, для зниження складності моделей виробничих систем здійснюють їх декомпозицію [1]. Зокрема, процеси механізованого обробітку ґрунту та сівби озимих культур розглядаємо на трьох рівнях декомпозиції ТС: 1) узагальненому; 2) часткових процесів; 3) операційному. Кожен із цих рівнів системно пов'язаний з іншим. Отримані результати того чи іншого рівня залежать від попереднього та доповнюють початкові дані для відображення механізованих процесів у наступному.

Щодо методики системно-подієвого відображення потоку вимог на виконання технологічних операцій із ОГС то вона передбачає використання встановлених нами залежностей (та закономірностей), а також розподілів ймовірнісних показників, які характеризують вплив агрометеорологічної та предметної складових на перебіг ґрунтообробно-посівних процесів: 1) залежність тривалості вегетації попередників озимих культур від часу їх сівби; 2) закономірність зміни тривалості природно дозволеного фонду часу на виконання ґрунтообробних посівних процесів літньо-осіннього періоду; 3) час початку сівби озимих культур; 4) тривалість погожих та непогожих проміжків; 5) час призупинення осінньої вегетації озимих культур тощо.

Потреба виконання тієї чи іншої технологічної операції в j -ту добу формується потоком вимог ґрунтообробно-посівних процесів, який моделюється у календарному проміжку між часом завершення збирання врожаю попередника та крайніми термінами сівби озимої культури.

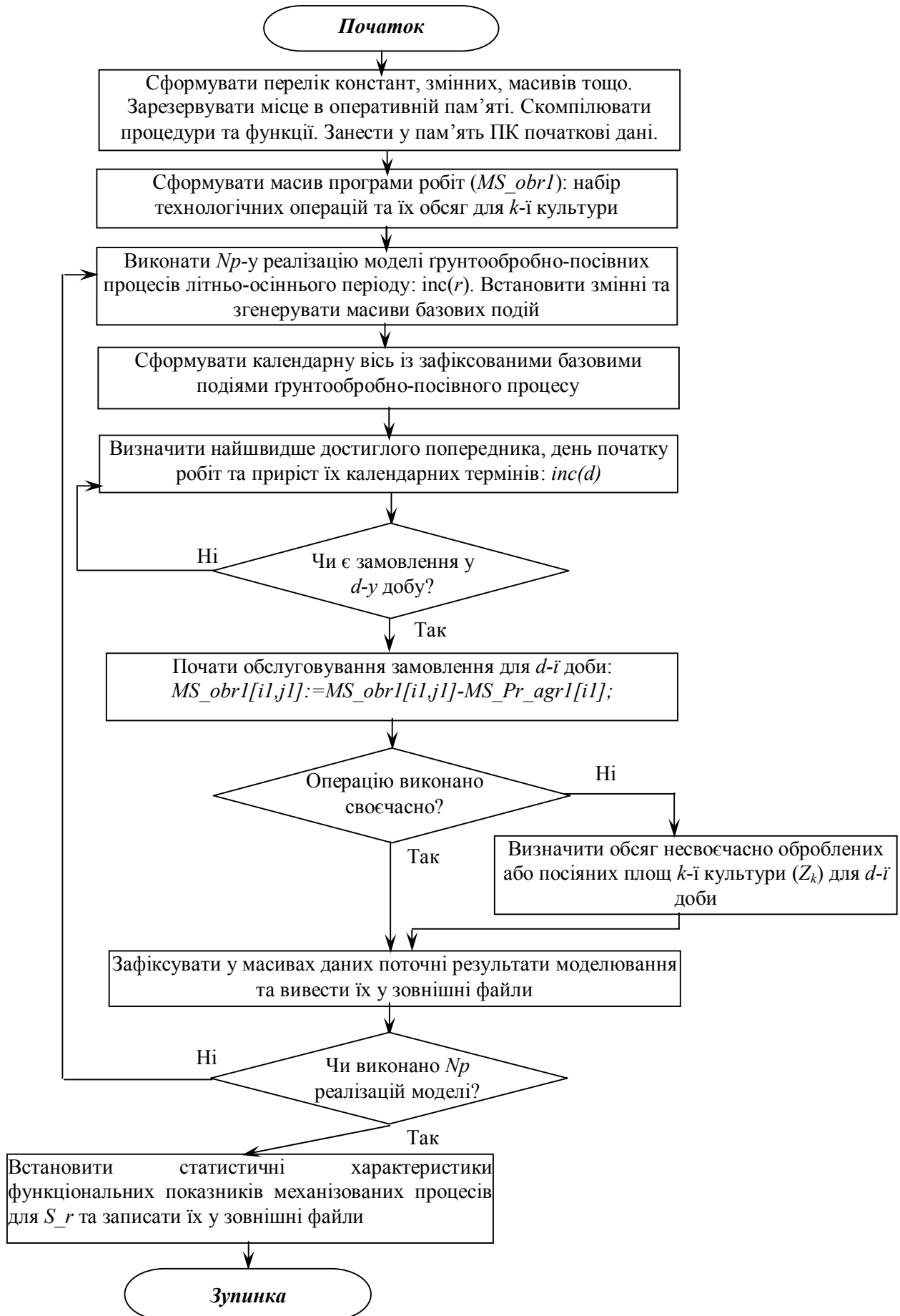


Рис. Узагальнений алгоритм статистичної імітаційної моделі ґрунтообробно-посівних процесів літньо-осіннього періоду

Площа, на якій необхідно виконати відповідну технологічну операцію, визначається виходячи з характеристик виробничої програми сівби (загальної площі та структури озимих культур). У міру виконання технологічних операцій площа, що потребує застосування відповідного машинного агрегату, зменшується до нуля. Після цього поле очікує на виникнення події, за якої формується вимога у потоці вимог на виконання наступної технологічної операції і т.д. Такі етапи відбуваються до виконання всіх робіт цього потоку вимог під k -ту озиму культуру. Щодо машинного агрегату, то в механізованих процесах рільництва, зокрема ґрунтообробно-посівних, насамперед розглядають такі його характеристики, які позначаються на темпах робіт. Зокрема, для встановлення добової ($W_{доб}^r$) продуктивності r -го машинного агрегату користуються відомою формулою:

$$W_{доб}^r = W_{20д}^r \cdot T \cdot k_{зм} \cdot k_{взм} \quad (1)$$

де $W_{доб}^r$ - нормативна годинна продуктивність r -го машинного агрегату, га/год; T – тривалість зміни, год; $k_{зм}$ - коефіцієнт змінності; $k_{взм}$ – коефіцієнт використання часу зміни.

Отже, у розрізі модельованого проміжку часу обслуговування потоку вимог відображається чергуванням відповідних базових та наслідкових подій. Як уже зазначалося, ґрунтообробно-посівний процес починається з базової події завершення збирання врожаю попередника ($\tau_{зз}^n$). Згідно з агротехнічними вимогами першу операцію луцення починають без розриву в часі з подією, що дає змогу зберегти вологу затінення [4]. Після завершення цієї технологічної операції за традиційної технології обробітку ґрунту наступну починають після відповідного розвитку бур'янів на полі і т.д. Щодо технологічної операції сівби, то її розпочинають із настанням базової події τ_c тощо. Черговість обслуговування замовлень здійснюється відповідно до черговості настання відповідних базових подій [8].

Для чисельного відображення системно-подієвого перебігу ґрунтообробно-посівних процесів розроблено відповідну математичну модель. Зокрема, відображення календарних термінів у чисельному виразі здійснено на підставі встановлення кількості днів між часом (календарною добою) настання відповідної події та 1 січня.

Процес обслуговування потоку вимог на виконання тієї чи іншої технологічної операції з обробітку ґрунту та сівби k -ї озимої культури в j -ту добу відображається відніманням добової продуктивності ($W_{доб}^{rj}$) r -го машинного агрегату від площі поля (S_{kj-1}^H), яка підлягає обробітку:

$$S_{kj}^H = S_{kj-1}^H - W_{доб}^{rj} \quad (2)$$

де S_{kj}^H - площа k -ї культури, яка залишилась необробленою в j -ту добу, га.

Необроблена площа S_{kj}^H м залишається на наступну добу ґрунтообробно-посівних процесів. Необхідно зазначити, що коли наступна доба виявиться непогожою, то польові роботи не відбуваються, у цьому разі $S_{kj}^H = S_{kj+1}^H$ і т.д. Зазначені роботи відбуваються до моменту досягнення умови $S_{kj}^H = 0$ га. Після цього переходять до обслуговування наступного замовлення потоку вимог і т.д. У разі, коли обслуговування певного замовлення завершено, а наступного ще немає, виникає простій техніки.

Запізнення із ґрунтообробно-посівними процесами відносно їх оптимальних термінів на практиці призводить до зниження врожайності культур [4]. Для кількісного оцінення втрат врожаю озимих через несвоєчасність робіт використано методику пошуку обсягів (Z_n) несвоєчасно оброблених та посіяних площ [5]. Сутність цієї методики полягає у встановленні суми добутоків несвоєчасно виконаної площі (S^H) та кількості днів (t_{nc}) запізнення із її виконанням, що вимірюється у гектаро-добах. За отриманим значенням Z_n виникає можливість вартісно оцінити технологічні втрати [7].

Таким чином, врахування об'єктивного впливу агрометеорологічної та предметної складових ґрунтообробно-посівних процесів на системно-подієвий їх перебіг у часі є важливою передумовою адекватного відображення сезонних умов функціонування ТКІП у відповідній статистичній імітаційній моделі, а відтак об'єктивного оцінення показників його ефективності. Розроблення блок-схеми алгоритму моделі, її комп'ютерної програми та виконання комп'ютерних експериментів із скінченною кількістю реалізацій цієї моделі для фіксованих параметрів ТКІП дає

змогу отримати множини даних, які характеризують той чи інший функціональний показник ефективності для заданих характеристик виробничої програми сівби озимих культур та мінливих агрометеорологічних і предметних умов літньо-осіннього періоду. Математичне опрацювання результатів цих комп'ютерних експериментів дає змогу встановити закономірність зміни оцінок математичного сподівання функціональних показників, здійснити їх вартісне оцінення та визначити екстремум значень, що є підставою для обґрунтування параметрів ТС ОГС на підставі узгодження характеристик виробничої програми сівби озимих культур із параметрами ТКП.

Висновки. Передумовою отримання вірогідних результатів комп'ютерних експериментів щодо встановлення функціональних характеристик технологічних комплексів машин для виконання ґрунтообробно-посівних процесів є використання даних спостережень агрометеорологічної станції. Розроблена блок-схема алгоритму та методика комп'ютерних експериментів скерована на системно-подієве відображення згаданих механізованих процесів у статистичній імітаційній моделі за прогнозованих агрометеорологічних умов та відповідних тенденцій зміни предметної складової (агрофону поля). Це дає змогу відобразити системно-подієві особливості функціонування відповідного комплексу машин, виконати багаторазову реалізацію статистичної імітаційної моделі та отримати репрезентативні вибірки показників щодо своєчасності виконання робіт, оцінити їх статистичні характеристики та обґрунтувати організаційно-технологічні рекомендації щодо раціональних параметрів ґрунтообробно-посівного комплексу машин.

Перспективи подальших досліджень полягають у можливості розроблення комп'ютерної програми відповідної статистичної імітаційної моделі та виконання комп'ютерних експериментів щодо встановлення статистичних закономірностей функціональних показників ґрунтообробно-посівних процесів за різних значень характеристик виробничої програми та параметрів технологічного комплексу машин.

Використана література

1. Альянах И. Н. Моделирование вычислительных систем / И. Н. Альянах. – Л. : Машиностроение, 1988. – 233 с.
2. Бомба М. Я. Наукові і практичні основи обробки ґрунту : навч. посіб. / М. Я. Бомба, З. М. Томашівський. – Івано-Франківськ : Галичина, 1993. – 148 с.
3. Завалишин Ф. С. Основы расчета механизированных процессов в растениеводстве / Ф. С. Завалишин. – М. : Колос, 1973. – 319 с.
4. Лихочвор В. В. Рослинництво. Технології вирощування сільськогосподарських культур / В. В. Лихочвор. – Львів : Укр. технології, 2002. – 800 с.
5. Метод визначення втрат врожаю сільськогосподарських культур внаслідок несвоечасності механізованих процесів рільництва / О. Сидорчук, П. Луб, А. Татомир, А. Бурилко // Механізація і енергетика сільського господарства "МОТРОЛ 2005" : матеріали V ювілейної міжнар. наук.-техн. конф. – Одеса, 2005. – Т. 7. – С. 87-91.
6. Пастухов В. І. Обґрунтування оптимальних комплексів машин для механізації польових робіт : автореф. дис. на здобуття наукового ступеня докт. техн. наук: спец. 05.05.11 „Машини і засоби механізації сільськогосподарського виробництва” / В. І. Пастухов; Харк. нац. техн. ун-т сіл. госп-ва ім. П. Василенка. – Х., 2004. – 38 с.
7. Про затвердження Методики обчислення вартості машино-дня та збитків від простою машин : Постанова Кабінету Міністрів України від 12 липня 2004 р. № 885 [Електронний ресурс] . – Режим доступу : URL : <http://zakon.rada.gov.ua/>.
8. Сидорчук О. Означення подій у проектах обробки ґрунту та сівби сільськогосподарських культур / О. Сидорчук, П. Луб, І. Івасюк, В. Українець // Вісник. Львів. НАУ: Агроінженерні дослідження. – Львів : Львів НАУ. – 2009. – № 13. – Т. 2. – С. 13-21
9. Сидорчук О. В. Технологічні вимоги механізованого процесу в рослинництві до темпів ремонту машин / О. В. Сидорчук, М. І. Карпа, В. О. Тимочко, С. А. Федосенко // Підвищення організаційно-технічного рівня ремонтно-відновних процесів в АПК регіону: Пр. ін-ту / Львів с.-г. ін-т. – Львів, 1990. – С 84-90.

УДК 005.8:631

А.М. Тригуба, к.т.н., доцент

Львівський національний аграрний університет

СТРУКТУРА СИСТЕМИ УПРАВЛІННЯ ТЕХНОЛОГІЧНО ІНТЕГРОВАНИМИ ПРОГРАМАМИ МОЛОЧАРСТВА

Тригуба А.М. Структура системи управління технологічно інтегрованими програмами молочарства Виконано аналіз проблем молочарства при вступі у зону вільної торгівлі з Євросоюзом. Проаналізовано існуючі структури управління програмами. Окреслено структуру системи управління технологічно інтегрованими програмами реінжинірингу молочарства. Обґрунтовано ланцюги формування управлінських рішень.

Ключові слова: структура, система, управління, технологічно інтегрована програма, молочарство.

Тригуба А.Н. Структура системы управления технологически интегрированными программами молочного скотоводства.

Аннотация. Выполнен анализ проблем молочного скотоводства при вступлении в зону свободной торговли с Евросоюзом. Проанализированы существующие структуры управления программами. Определена структура системы управления технологически интегрированными программами реинжиниринга молочного скотоводства. Обоснованы цепи формирования управленческих решений.

Ключевые слова: структура, система, управление, технологически интегрированная программа, молочное скотоводство.

A.M. Tryhuba. System structure of the technologically integrated programs of milk production.

Summary. The analysis of milk production problems of free trade area with the EU are given. The structure of existing agricultural production program management is analyzed. The structure of the integrated management system software of milk production reengineering are outlined. The circuit formation of management decisions are grounded.

Keywords: structure, system, management, technologically integrated programs, milk production.

Постановка проблеми та завдання дослідження. Підписання угоди про зону вільної торгівлі з Євросоюзом призведе до того, що обмежиться доступ українських молокопродуктів на ринки [1]. Це пов'язано із тим, що чинні стандарти молочарства України не відповідають стандартам Євросоюзу. Для вирішення існуючої проблеми слід реалізовувати технологічно інтегровані програми реінжинірингу молочарства (ТППРМ) та розробляти відповідні науково-методичні засади управління ними. Управління ТППРМ є багаторівневим і має свою специфіку. Існує потреба обґрунтування структури системи управління ТППРМ та ланцюгів прийняття управлінських рішень стосовно функціонування об'єктів управління.

Аналіз останніх досліджень і публікацій. Дослідженню проблеми формування структури системи управління та прийняття управлінських рішень під час реалізації програм у різних галузях народного господарства приділено достатньо уваги провідними науковцями [2, 3, 4]. Однак, їх використати для обґрунтування структури системи управління ТППРМ неможливо через специфіку прийняття управлінських рішень у цих програмах.

Метою дослідження є означити структуру системи управління ТППРМ та особливості прийняття управлінських рішень у них.

Виклад основного матеріалу. Виділяють декілька рівнів управління ТППРМ. Усі ці рівні є взаємопов'язаними, а кількість суб'єктів управління на кожному із них залежить від наявності інтеграційних зв'язків стосовно прийняття управлінських рішень. Розрізняють два види інтеграційних зв'язків стосовно прийняття управлінських рішень у ТППРМ – горизонтальні та вертикальні. Горизонтальна інтеграція пов'язана із внутрішньогалузевим кооперуванням, яке забезпечує прийняття управлінських рішень стосовно окремих ланок єдиного технологічного ланцюга виробництва молокопродуктів з метою збільшення і поліпшення якості молокопродукції, підвищення ефективності її виробництва [5]. У ТППРМ можна виділити три ланки горизонтальної інтеграції – виробництво, заготівля та переробка молока. Горизонтальне управління ТППРМ здійснюється бізнесовими структурами, які реалізують відповідні програми та проекти. Вертикальна інтеграція у прийнятті управлінських рішень стосовно реалізації ТППРМ відповідає рівням державного управління молочарством: державний, регіональний (обласний), місцевий (районний), селищний (окремих населених пунктів) та об'єктів (окремих підприємств та господарств).

Систему управління ТППРМ розглянемо у вигляді багаторівневого контуру, як сукупність взаємодіючих підсистем, що можна поділити на два види – суб'єкти управління (управлінські підсистеми) і об'єкти управління (підсистеми «ТППРМ» відповідного рівня) (рис. 1).

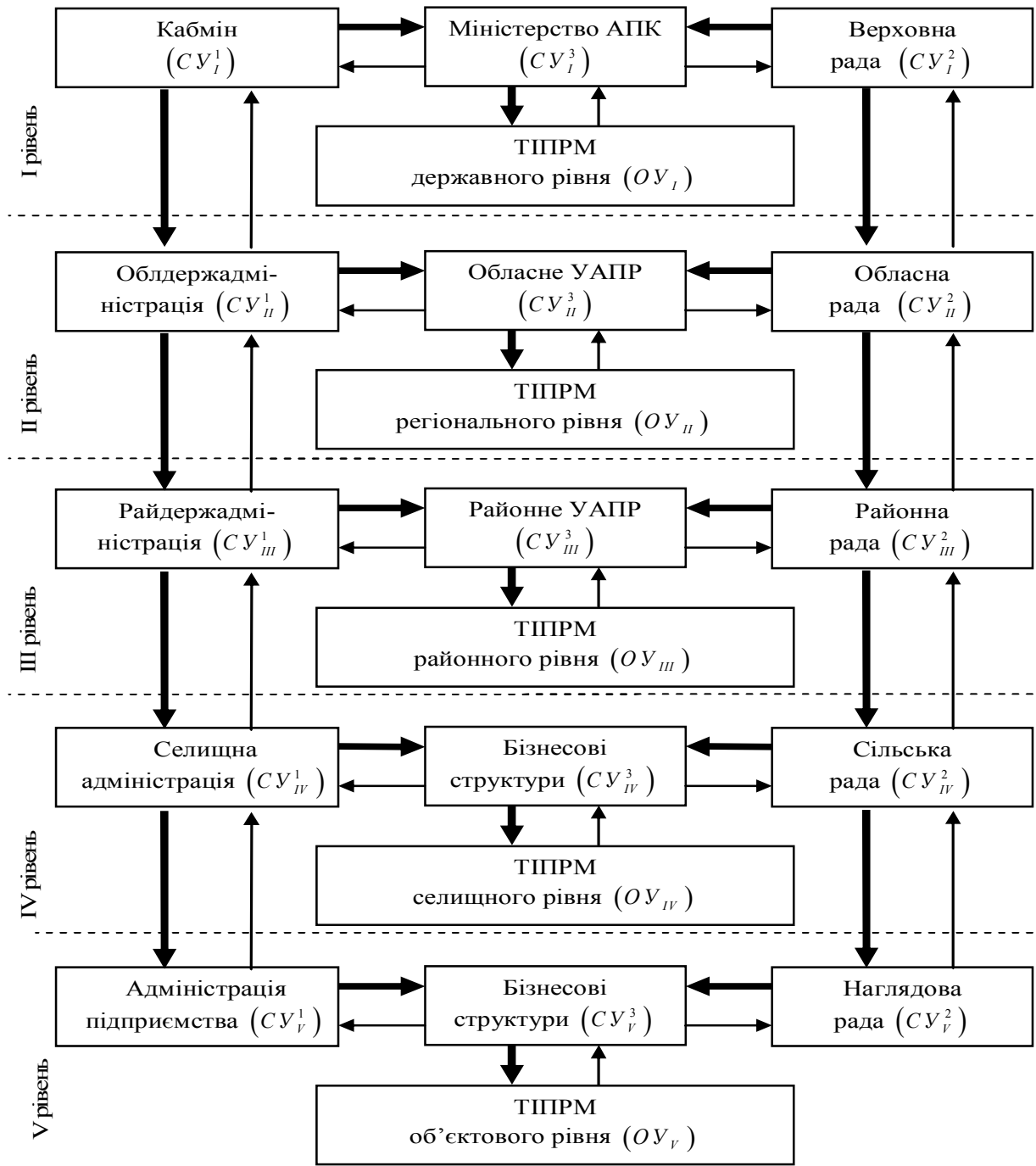


Рис. 1. Структура системи управління ТППРМ:
 —————> – управлінські рішення; —> – інформація

Управління ТППРМ різних рівнів здійснюється декількома суб'єктами. Одні із них здійснюють управління з позицій держави (Міністерство АПКтаПУ, обласні та районні адміністрації), інші з позицій населення (обласні, районні та селищні ради), а ще інші з позицій бізнесових структур, що реалізують ТППРМ. Кожен із яких хоче отримати свої вигоди, що є переважно різнонаправленими.

Управління ТППРМ полягає у взаємодії між суб'єктами (CU) та об'єктами управління (OU). CU надсилають рішення OU, які містять у собі інформацію щодо функціонування OU. Однак, особливістю управління ТППРМ є те, що управлінські рішення формуються у декілька етапів і коригуються із врахування інтересів різних CU.

На кожному із рівнів управління ТППРМ є свої СУ, які приймають рішення стосовно функціонування ОУ. Ланцюги формування управлінських рішень у розрізі окремих рівнів виглядають наступним чином:

➤ I рівень
$$(CY_I^1, CY_I^2) \Rightarrow CY_I^3 \Rightarrow OY_I, \quad (1)$$

➤ II рівень
$$(CY_{II}^1, CY_{II}^2) \Rightarrow (CY_{II}^1, CY_{II}^2) \Rightarrow CY_{II}^3 \Rightarrow OY_{II}, \quad (2)$$

➤ III рівень
$$(CY_{III}^1, CY_{III}^2) \Rightarrow (CY_{III}^1, CY_{III}^2) \Rightarrow CY_{III}^3 \Rightarrow OY_{III}, \quad (3)$$

➤ IV рівень
$$(CY_{IV}^1, CY_{IV}^2) \Rightarrow (CY_{IV}^1, CY_{IV}^2) \Rightarrow CY_{IV}^3 \Rightarrow OY_{IV}, \quad (4)$$

➤ V рівень
$$(CY_{V}^1, CY_{V}^2) \Rightarrow (CY_{V}^1, CY_{V}^2) \Rightarrow CY_{V}^3 \Rightarrow OY_{V}. \quad (5)$$

Прийняття ефективних управлінських рішень кожним із СУ та відповідно їх виконання ОУ можливе за умови системного узгодження інтересів між учасниками ТППРМ. Тобто, управлінські рішення, що приймаються на вищому рівні повинні враховувати управлінські рішення нижнього рівня і навпаки. Це можливо забезпечити за умови балансування цінностей між учасниками ТППРМ. Окрім того, для виконання прийнятих управлінських рішень слід щоб у СУ була потреба і можливість управляти ОУ, а ОУ у свою чергу були готові виконувати ці рішення та забезпечені необхідними ресурсами. Отже, для ефективного управління ТППРМ слід системно узгоджувати рішення СУ та можливості їх виконувати ОУ усіх рівнів.

Структура процесів управління ТППРМ має свою специфіку. Зокрема, завдяки реалізації ТППРМ здійснюється переведення існуючої системи молочарства у бажаний стан без припинення функціонування цієї системи. Окрім того, має свою специфіку середовище в якому реалізуються ТППРМ – мінливий характер його складових та циклічність виконання окремих програм.

Відповідно до методології Р2М [6] управління програмами базується на управлінні їх інтеграцією та середовищем. Свої особливості мають процеси управління ТППРМ на концептуальній фазі їх життєвого циклу. Зокрема, ці управлінські процеси об'єднано у чотири групи: 1) профілювання місії (ідентифікація проблем (M_1), формування місії (M_2), аналіз взаємозв'язків між цілями та задачами (M_3) та обґрунтування сценаріїв (MC)); 2) управління стратегією (опис існуючого стану (C_1); прогноз розвитку внутрішнього та зовнішнього середовища (C_2); ідентифікація конфігурацій систем-продуктів (C_3)); 3) управління архітектурою (структурування програми (A_1), обґрунтування тривалості життєвого циклу програми (A_2), реалізацію функціональності (формування підпрограм) для отримання цінностей від реалізації програми (A_3), розробка концептуального плану програми (KPP)); 4) управління оцінюванням (проектування базової структури цінностей (O_1), визначення індикаторів та показників для кожної складової цінності (O_2), регулярна оцінка проектів і програм (O_3)). Між означеними процесами існують взаємозв'язки. Ці взаємозв'язки відображено у моделі управління ТППРМ на концептуальній фазі їх життєвого циклу (рис. 2).

Нами сформульовано задачі, вирішення яких дасть можливість підвищити ефективність управління ТППРМ на концептуальній фазі їх життєвого циклу. Профілювання місії ТППРМ потребує вирішення наступних задач: 1) аналіз чинників їх цінності; 2) обґрунтування причинно-наслідкових зв'язків між чинниками цінності; 3) визначення ефективного сценарію переведення системи молочарства із існуючого у бажаний стан.

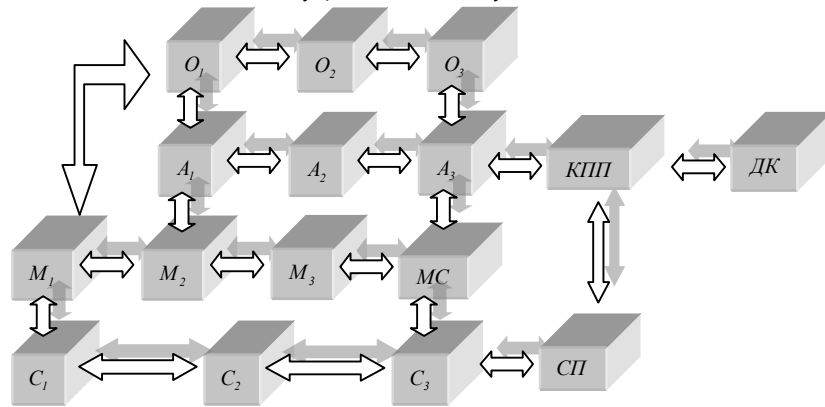


Рис. 2. Модель управління ТПРМ на концептуальній фазі їх життєвого циклу

Для ефективного управління архітектурою ТПРМ слід вирішувати задачі: 1) структурування програми; 2) обґрунтування причинно-наслідкових зв'язків між проектами програми; 3) формування підпрограм програми; 4) визначення тривалості життєвого циклу програми. Для управління стратегією ТПРМ слід вирішити наступні задачі: 1) обґрунтування стратегічного бачення продукту ТПРМ; 2) довгострокове прогнозування проектного середовища ТПРМ; 3) узгодження стратегії розвитку молочного тваринництва із проектним середовищем. Управління оцінюванням ТПРМ потребує вирішення наступних задач: 1) проектування базової структури цінностей ТПРМ; 2) визначення критеріїв та показників оцінювання цінностей ТПРМ; 3) балансування цінностей, які отримують зацікавлені сторони.

Отже, для ефективного управління ТПРМ слід вирішувати сформульовані задачі. Це можливо на підставі розроблення нових та удосконалення існуючих методів та моделей, які враховуватимуть специфіку реалізації цих програм та дадуть можливість позбутися прийняття хибних управлінських рішень.

Висновки. 1. Виконаний аналіз стану молочарства України свідчить про те, що існує загроза ринку вітчизняних молокопродуктів після підписання угоди про зону вільної торгівлі з Євросоюзом, а для її вирішення слід реалізовувати технологічно інтегровані програми реінжинірингу молочарства. 2. Структура системи управління ТПРМ є багаторівневою, а у прийнятті управлінських рішень стосовно реалізації цих програм беруть участь декілька суб'єктів управління різних рівнів. 3. Сформовані ланцюги прийняття управлінських рішень стосовно об'єктів управління (ТПРМ), є основою для узгодження інтересів учасників ТПРМ. 4. Подальші дослідження слід виконувати стосовно розроблення науково-методичних засад узгодження інтересів між учасниками ТПРМ різних рівнів.

Список використаної літератури

1. Молоко для ЄС: місія неможлива? Режим доступу: <http://www.agrotimes.net/moloko-dlya-ec.html>.
2. Бушуева Н.С. Модели и методы проактивного управления программами организационного развития / Н.С. Бушуева. – К.: Наук. світ, 2007. – 270 с.
3. Денисов А.А. Теория больших систем управления / А.А. Денисов, Д.Н. Колесников. – Л.: Энергоиздат, 1982. – 288с.
4. Желтенков А.В. Исследование систем управления : ученик / А.В. Желтенков, С.А. Рябиченко. – М. : МГОУ, 2008. – 113 с.
5. Малік М. Й. Інтеграція – як фактор підвищення ефективності реформування сільськогосподарських підприємств / М. Й. Малік, П. М. Федієнко, М. К. Орлатий. – К., 2000. – 39с.
6. Руководство по управлению инновационными проектами и программами Р2М: т. 1, версия 1.2 / пер. на рус. язык под ред. С.Д. Бушуева. – К. : Наук. Світ, 2009. – 173 с.