

*МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ*

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

НАУКОВИЙ  
ЖУРНАЛ



*Відповідальний редактор – проф., д-р техн. наук В. Д. Рудь*

## **№24-25 2016**

*м. Луцьк  
Видавництво Луцького національного технічного університету*

## РЕДАКЦІЙНА КОЛЕГІЯ:

### Головний редактор:

проф., д.т.н. Рудь В.Д. (м.Луцьк)

### Заступники головного редактора:

доц., к.т.н. Мельник К.В. (м.Луцьк)

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

### Відповідальний секретар:

мол.наук.співробітник Свиридюк К.А. (м.Луцьк)

### Члени редакційної колегії:

проф, PhD. Milosz Marek (Польща, м.Люблін)

проф, PhD. Alison McMillan (Великобританія)

проф, PhD. Дехтяр Ю.Д. (Литва, м.Рига)

проф., д.т.н. Мазур М.П. (м.Хмельницьк)

проф., д.т.н. Мельник А.О. (м.Київ)

проф., д.т.н. Сидорчук О.В. (м.Київ)

проф., д.т.н. Тарасенко В.П. (м.Київ)

проф., д.ф-м.н. Пастернак Я.М. (м.Луцьк)

проф., д.т.н. Андрушак І.Є. (м.Луцьк)

проф., д.т.н. Делявський М.В. (м.Луцьк)

проф., д.е.н. Рудь Н.Т. (м.Луцьк)

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

доц., к.т.н. Драган О.В. (м.Брест, Білорусія)

доц., к.т.н. Лотиш В.В. (м.Луцьк)

доц., к.т.н. Гуменюк Л.О. (м.Луцьк)

доц., к.т.н. Пех П.А. (м.Луцьк)

доц., к.т.н. Самчук Л.М. (м.Луцьк)

доц., к.пед.н. Потапюк Л.М. (м.Луцьк)

доц., к.т.н. Решетило О.М. (м.Луцьк)

доц., к.т.н. Повстяной О.Ю. (м.Луцьк)

### Адреса редколегії:

Луцький національний технічний університет,  
кафедра комп'ютерної інженерії.

вул.Львівська 75, ауд.141

м.Луцьк, 43018

тел. (0332) 74-61-15

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

ekaterinamelnik@gmail.com

сайт журналу: [ki.lutsk-ntu.com.ua](http://ki.lutsk-ntu.com.ua)

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

*№24-25 2016р.*

Журнал засновано у грудні 2010 р.  
Свідоцтво про реєстрацію КВ № 16705-5277 Р.  
Засновник: Луцький національний технічний університет  
**Рекомендовано до друку Вченою радою**  
**Луцького національного технічного університету**  
**(протокол №5 засідання від 27.12.2016)**  
Журнал рішенням МОН України  
наказом №515 від 16.05.2016р,  
включено в перелік наукових фахових видань.  
Журнал має російський індекс наукового цитування  
(РІНЦ)

**ISSN 2524-0560 (Online)**  
**ISSN 2524-0552 (Print)**

## ЗМІСТ

<b>ІНФОРМАТИКА ТА ОБЧИСЛЮВАЛЬНА ТЕХНІКА</b>	
<b>Андрушак І.Є., Ящук А.А.</b> Сучасні технології підвищення інформаційної безпеки інформаційних систем.	<b>5</b>
<b>Бортник К. Я., Волошин М.М.</b> Створення мультиплатформних додатків за допомогою JAVASCRIPT.	<b>10</b>
<b>Бурчак І.Н., Головачук І.П., Величко В.Л.</b> Впровадження додаткових сценаріїв тестування студентів у платформі MOODLE згідно з потребами кафедри ІКТ.	<b>14</b>
<b>Єрмейчук С.Ю., Федонюк А.А.</b> Розробка агрегатора курсу валют для ОС Android.	<b>19</b>
<b>Каганюк О.К., Поліщук М.М., Гаджемура А.М.</b> Розробка раціональної моделі джерела живлення для ПЛІС.	<b>28</b>
<b>Каганюк О.К., Панчук Р.Я.</b> Дослідження швидкодії роботи з базами даних різних типів програм, та вибір раціональних параметрів апаратних ресурсів при однаковому функціоналі.	<b>33</b>
<b>Коцюба А.Ю., Цяпич Я.П., Лавренчук С.В.</b> Про методику оптимізації відмовостійкості веб-серверів на одночасну кількість запитів.	<b>37</b>
<b>Марченко О.О., Марченко О.І.</b> Критерій «глибина-ширина» для контролю форми дерева пошуку при використанні методу Монте-Карло.	<b>42</b>
<b>Пех П.А., Бурчак І.Н., Тихомиров В.В.</b> Розробка моделей в Autodesk Maya з метою подальшого використання у ігровому рушії Unreal Engine 4.	<b>48</b>
<b>Яцюк С.М., Муляр В.П.</b> Використання макросів бази даних Access при вивченні інформатики.	<b>54</b>
<b>АВТОМАТИКА ТА УПРАВЛІННЯ</b>	
<b>Бурбело М. Й., Гадай А.В.</b> Визначення потужностей нелінійних навантажень трифазних електричних мереж.	<b>61</b>
<b>Гінайло П.І.</b> Задача оптимального керування для диференціальних включень з багатовисновковими відображеннями.	<b>68</b>
<b>Гінайло П.І.</b> Про еквівалентність двох задач оптимального керування з диференціальними включеннями.	<b>73</b>
<b>Гуда О.В., Тимошук В.М., Крадінова Т.А.</b> Метод виведення рівнянь рівноваги в некласичній постановці.	<b>78</b>
<b>Жигаревич О.К.</b> Дослідження та представлення навчальної системи PROTUS 2.0.	<b>84</b>
<b>Лактіонов О.І.</b> Верстат з числовим програмним керуванням як об'єкт і суб'єкт управління.	<b>88</b>
<b>Максимович В.М., Соляр Т.Я., Приходько О.С.</b> Числовий алгоритм визначення напружень біля підкріплених отворів у пластинках.	<b>93</b>
<b>Максимович О.В., Лавренчук С.В.</b> Числовий аналіз напружень біля штампів і тріщин у анізотропній півплощині за врахування тертя.	<b>99</b>
<b>Пастернак В.В., Пастернак Я.М.</b> Об'єктно-орієнтована реалізація методу граничних елементів тривимірної термомагнітоелектропружності.	<b>107</b>
<b>Решетило О.М., Павлік П.В., Токарчук В.В.</b> Підвищення потужності та зменшення шкідливих викидів дизельного двигуна внутрішнього згорання автомобіля.	<b>113</b>
<b>Чабан О.В., Костючко С.М., Міскевич О.І., Киричук А.А.</b> Верифікація на основі числового експерименту двох методів дослідження параметричної чутливості.	<b>119</b>
<b>Чиб С.С., Мельник В.М., Мельник К.В., Самарчук В.Ф.</b> Характеристика деяких закономірностей впливу соціальних мереж інтернет на особистісний розвиток студентів.	<b>124</b>

УДК 61:004.651 (075.8)

Луцький національний технічний університет  
Андрушак І.Є., Яшук А.А.

## **MODERN TECHNOLOGIES INCREASING INFORMATION SECURITY INFORMATION SYSTEMS**

**Андрушак І.Є., Яшук А.А.** Сучасні технології підвищення інформаційної безпеки інформаційних систем. В роботі запропоновано застосування комплексної методики тестування як складової забезпечення безпеки інформаційної системи підрозділу вищого навчального закладу. Визначені інструментальні засоби розробки, запропонований план тестування, проведений аналіз результатів тестування, надані рекомендації щодо забезпечення належного рівня безпеки інформаційної системи.

**Андрушак И.Е., Яшук А.А.** Современные технологии повышения информационной безопасности информационных систем. В работе предложено применение комплексной методики тестирования как составляющей обеспечения безопасности информационной системы подразделения высшего учебного заведения. Определены инструментальные средства разработки, предложенный план тестирования, проведен анализ результатов тестирования, даны рекомендации по обеспечению надлежащего уровня безопасности информационной системы.

**Andrushchak I.Ye., Yashchuk A.A.** Modern technologies increasing information security information systems. The paper presents a comprehensive application testing methodology as part of the information system security division of higher education. Defined development tools, the proposed test plan, held the analysis of test results, recommendations to ensure appropriate security of information systems.

In recent years, the topic of integrated management information systems of higher education institutions (HEIs) updated as more modern trends in the field of information technology and the most acute need for universities to increase their efficiency in relation to management of internal organizational and economic processes. Information systems in education serve as the technical basis for the development and improvement of the educational process.

Domestic universities usually have a branched structure, a large number of staff and student contingent, comparable to large enterprise financial performance and require effective management.

Today many universities made efforts of their own development of information systems. Most of these systems can be divided into localized within the controlled area and distributed when data is transferred outside the controlled zone. In addition to distributed information systems environment data protected, is a local or corporate network.

The data circulating in the university management information system can be divided into open data (curriculum, schedule, etc.) and data distribution restricted (confidential) that can be classified as follows:

- personal data of students, staff and others;
- data vidnoshuvani to official secrets;
- data vidnoshuvani to trade secrets.

Naturally, in this situation there is a need to protect the information system of the university and information from unauthorized access, theft, destruction, and other malicious and unwanted actions, including failure of hardware and software systems.

According to the publications accompanying information systems development institutions, the authors suggest the general principles of such systems provide a list of functional subsystems description of network architecture, the structure of the software. But the issue of information security proper attention is not given, none of the existing systems do not address the issue comprehensive protection of confidential information. This confirms that the level of measures to protect information in information systems tend to lag behind the pace of automation.

Due to the lack of elaboration of information security, a large proportion of the declared capacity remains unfulfilled, and even working pieces of the system can not practically be introduced in other universities.

In any case, it is necessary to form a reasoned, integrated views of information security issues during the design, creation, commissioning, operation and upgrading university information system that will talk about the

creation of automated information systems in a secure performance.

In an effort to creation of information security and secure use of information technology developed by a number of international standards, procedures and other documents regulating these issues. The definition and compliance requirements coming out of international standards, and more dependent on the leadership of the organization, but to integrate into the civilized information space of voluntary compliance is needed.

The first success in this field was the British Standard 1995 r. BS 7799 "Practical rules for information security management", which summarizes the experience of providing the information security of information systems in various fields. It was later published several similar documents – standards of various organizations and agencies, the German standard BSI (Bundesamt für Sicherheit in der Informationstechnik). At the end of 2000 passed the ISO 17799, which is based on BS 7799.

The most known and used in international practice standards:

- International Information Systems Security Standard ISO / IEC 17799 "Information Security Management" (Information Technology – Information Security Management);
- International Standard ISO / IEC 15408-1: 1999 "Information technology. Methods and means of security. Security Evaluation Criteria Information Technology" (so-called Common Criteria);
- Standard CobiT - "Control Objects for Information and related Technology".

The ideas contained in these documents are as follows. Practical rules of information security at all stages of the life cycle of information technology should be integrated and based on proven practice techniques and methods. For example, the mandatory use of some means of identification and authentication of users (services), backup, antivirus control, etc.

When creating a system of information security is important not to exclude any substantial aspect – in this case, information technology, as used to be guaranteed an appropriate level of information security.

In the literature devoted to the protection of information, you can find different versions of models of information security threats. This is due to the desire to more accurately describe the various situations impact on the information and determine the most appropriate planning measures to protect it.

Based on the analysis [6], all sources of security threats can be generalized division into three main groups:

- the threat caused by the actions of the subject (androgenic threats);
- threats caused by technical means (man-made threats);
- threats due to natural sources.

The results of the survey show that currently among the main threats to security 43% are viruses, 26% - Failure in the major corporate systems, 15% - attack to cause denial of service, 14% - penetration from the outside, 11% - penetration inside 5% - violation of the integrity of the data, 3% - financial fraud, 2% - the theft of commercial information [6].

These data indicate that the impact on the safety of man-made threats is high enough. In addition, technological threats are less predictable than androgenic directly depend on the properties of the hardware and software and therefore require special attention.

To include man-made threats including poor quality software processing. As a result of errors in the software may be affected performance information processing systems as a whole, which directly affects the security of information.

In modern scientific literature reliable system is defined as "a system that uses sufficient hardware and software to provide simultaneous processing of information of varying degrees of secrecy group of people without access violation" [1].

Thus, without the use of methods of improving the reliability of software can not be building an integrated comprehensive security system. It is in this context necessary to design and build an information management system of higher education institution. Reduce the negative impact of information security threats may have different methods, among which should provide testing software system.

General view of the test software (software) for the recent years has evolved, becoming more constructive, pragmatic and close to the reality of modern software development projects. Testing is no longer seen as an activity that begins only after the design phase. Today testing is considered as the activities that should be carried out throughout the development process and support and is an important part of designing software.

It is easier to prevent a problem than to deal with its consequences. Testing, along with risk management is a tool that allows you to act in that way. Adequate attention to problems of quality testing reduces the risk of errors during operation, ensuring higher user satisfaction, which is, in essence, the goal of any project.

Testing – process analysis and software operation performed to assess and improve the quality of software by identifying defects and problems in software systems. During testing, the detection of deviations from the expected results of, this deviation record as a "problem" to clarify its reasons.

Testing – activities performed for assessing and improving software quality. This activity, in general, is based on the detection of defects and problems in software systems.

Effective testing is necessary to determine the basic process of testing. Its main purpose – to define a set of actions (tasks), inputs (input data to perform the process) and outputs (results of the) testing criteria for the initiation and completion of tasks, roles and responsibilities in the task of testing the aircraft [4].

The testing process can be represented by multiple problems with training, implementation and evaluation of test results, which are distributed by step process lifecycle.

At each step of the preparation of the analysis work products corresponding development process (input for this step of the testing process) for setting goals, objects, scripts, and testing resources, adequate level of testing. The results of test preparation steps are recorded in the test plan. In addition to fixing the results of the tests carried out comparing them with the expected results.

Test results at every level are analyzed in order to determine a current state of the aircraft and to decide on the adequacy of testing at this level.

Full system testing because of its huge complexity in terms of limited resources, often impossible to conduct. In order to find a compromise between a limited time, budget and resources, it is necessary to introduce a system of priorities that should be developed based on the requirements for AIS. The system will determine the priorities that customer requirements (usually perform any function) should be tested first, performance requirement is a secondary character.

Requirements sorted by priority will make it possible to build the scripts use system that will form the basis of test scenarios. Requirements for the aircraft and therefore the importance of appropriate software modules can rank the seriousness of the consequences of failures that occur on condition failure respective functions.

The key point of reference in software testing is to determine the criteria for its completion. Definition of testing criteria conclusion based on risk analysis allows failures to find a compromise between limited resources for testing (money, time, labor costs) on one side and almost unlimited demands on testing. Under conditions of limited resources for testing information management system Faculty completion criteria should be established based on risk assessments of threats related to the software system failures. It should take into account that risk is identified through testing, and what is the severity of defects remaining [2].

To identify and assess the risks of the aircraft to apply techniques such as analyzing tree analysis and event tree analysis of failures. Construction of structural and logical framework identifying threats to the aircraft is performed in two stages:

- construction of event trees and bounce at the system level, elements of which are functional components of the system;
- construction failure trees for each functional component (as well as individual software components and modules).

The analysis of event trees identify components failures events which may affect the safety of the system. The effects of each event (which give rise to other events) trasuyutsya until until identified threats (the highest level) associated with these events (deductive method of search conditions and causes of component failure events).

Fault tree analysis – the process of identifying the risks associated with system components. Disclaimer – hierarchical structure, the top (the root) of which – the state of denial component adjacent branches – events that lead to transition to this state. The origins of each branch – the state component element that led to the occurrence of a specific event (deductive method of search conditions and causes of component failure events). After identifying threats to build scenarios using the system, which can lead to failure of a software system.

It remains necessary to hold the so-called "smoke" testing. "Smoke" testing – a form of testing the software after changing its configuration or after changing his own (software). Smoke test is conducted to detect errors in the

operation of the software system through its operation on real data. In addition, during the "smoke" testing can gather information on the frequency of use of each scenario (in line with the concept of operational profile) and the consequences of failures associated with the use of appropriate software modules.

After the smoke test to decide whether to continue to use unit testing, or to make changes to the source code (in the data) to correct errors identified during smoke testing. After making changes (code or data) once again to conduct smoke tests, and if it will conduct further testing, unit testing to proceed.

For the unit testing should select the method by which this test will be conducted. There are two options: the method of "black box" (functional testing) method and the "white box" (white-box testing). Structural testing though is more labor intensive, provides more efficient (possibility *ceteris paribus* detect more errors) testing compared to functional because it uses information about the structure and content of the code. This method of testing is very time-consuming because in complex systems should be used to confirm the performance of the most important, critical requirements.

Structural testing methods should be used for the modules are related functionality, such as entering, deleting, updating data in a database, security, etc.

To test requirements, not related to a very important and essential to apply rational methods of functional testing, where known in the system, its behavior and expected outputs. If you run some test scenarios provide system behaves, it gives reason to believe that the tested module at certain input is working properly.

After all procedures a module will be tested methods of unit testing, integration must go to testing. After all procedures module will be tested separately (unit testing), as well as the interaction (integration testing) can be argued that the module (a function, sub-functions) tested and working properly [6].

For integration testing is necessary to use an incremental approach, which first tested the interaction of two modules (submodules). If successful interaction of two modules alone should add more, while checking the interaction has a larger number of modules.

The next stage of testing is system testing – testing the system. This performance can be assessed application performance of its reliability. In case of errors at this stage of testing necessary to ensure their removal. Control also provides for the amendment of the re module, integration testing (directional testing) of units affected by the modification.

System test (test) should be conducted under conditions as close to real. There is a concept test configuration – a set of hardware and software, with which the test case run. Test configuration (configuration) must meet the hardware configuration and software installed on the customer's site.

The final stage of testing is testing on-site customer. The purpose of acceptance testing is to verify customer requirements for aircraft – Software certification according to specification.

The next step was conducting "smoke test", which confirmed the performance of the aircraft and decided to conduct unit testing.

Because of limited resources and time of testing proposed decision not to apply the methods of functional testing at unit testing, while the recommended method of functional testing test system in accordance with the previously developed operating profiles the system. This approach implements as unit testing and integration, as well as perform a specific operating profile can require the procedures of different software modules.

In this work of unit testing by "white box" was recommended only in testing modules, whose contribution to the identified threat is very significant, and provided that the corresponding software module structure is complicated. Structural testing methods were used for testing modules that are making a serious contribution to the identified threat software system failures and implement complex algorithm processing. So to test the import module curriculum specialties was the method branches.

Regarding use of test scenarios that could potentially lead to system failures and do not implement complex control algorithm applied methods of "directed tests" as a form of functional testing.

It was also given preference testing requirements under previously developed operating profiles rather than testing structural components corresponding to the classical method of module and integration testing.

The proposed system tests can be used to ensure an adequate level of security similar data processing systems.

1. Калянов Г. Н. Построение архитектуры предприятия. // Корпоративные системы. – 2005 – №3. – С. 9-12.
2. Мур М. Аналитическая підтримка безпеки інформаційних систем // Міжнародний форум за інформацією. – М.: ВІНПІ, 2002. – Т.27 № 2.
3. Международный стандарт ISO/IEC 17799. Информационные технологии – Свод практических правил для управления защитой информации, ISO/IEC, 2000.
4. Основы инженерии качества программных систем / Ф.И.Андон, Г.И.Коваль, Т.М. Коротун, В.Ю. Суслов / Под ред. И.В. Сергиенко. – К.: Академперіодика. – 2002. – 504 с.
5. Коваль Г.И., Мороз Г.Б., Коротун Т.М. Концепция профилей в инженерии надежности программных систем // Математичні машини і системи. – 2004. – № 1. – С. 166-184.
6. О. М. Степанова, А. В. Велігура. Використання архітектурного підходу для проектування інформаційних систем. // Вісник Східноукраїнського національного університету ім. В. Даля. 2008. – №3(121). – С. 86-92.



УДК 004.94

Бортник К. Я., Волошин М. М.

Луцький національний технічний університет

## СТВОРЕННЯ МУЛЬТИПЛАТФОРМНИХ ДОДАТКІВ ЗА ДОПОМОГОЮ JAVASCRIPT

**Бортник К. Я., Волошин М. М. Створення мультиплатформних додатків за допомогою JAVASCRIPT.** У даній статті оглянуто переваги та недоліки мультиплатформних додатків написаних з використанням мови програмування JavaScript. Виконано порівняння існуючих засобів для створення таких додатків.

**Ключові слова:** JavaScript, мультиплатформні додатки, веб-технології, мобільна розробка

**Бортник К. Я., Волошин М. М. Создание мультиплатформенной приложений с помощью JAVASCRIPT.** В данной статье рассмотрено преимущества и недостатки мультиплатформенных приложений написанных с использованием языка программирования JavaScript. Выполнено сравнение существующих инструментов для создания таких приложений.

**Ключевые слова:** JavaScript, мультиплатформенные приложения, веб-технологии, мобильная разработка

**Bortnyk K. Y., Voloshyn M. M. Creating a multiplatform applications with JAVASCRIPT.** This article describe the advantages and disadvantages of multiplatform applications written with the programming language JavaScript. A comparison of existing tools to create such applications.

**Keywords:** JavaScript, cross-platform applications, Web-technologies, mobile development

Мультиплатформне програмне забезпечення – це програмне забезпечення, що може працювати на більше ніж одній апаратній платформі або операційній системі. Його можна поділити на два види: перший вимагає незалежної збірки або компіляції, другий – може запускатись безпосередньо на будь-яких платформах без попередніх підготовок, таке програмне забезпечення створюється на інтерпретованих мовах програмування.

Створювати програми таким чином доцільно лише коли планується їх видавництво на більше ніж одній платформі. Це пропонує єдину спільну частину коду, за рахунок чого тривалість розробки менша ніж при створенні на кожен платформу окремо. Розробка програми ведеться на одній технології, тому в цьому випадку не доведеться наймати відповідних фахівців при розширенні підтримуваних систем.

Разом з перевагами, такого підходу наявні недоліки та складнощі:

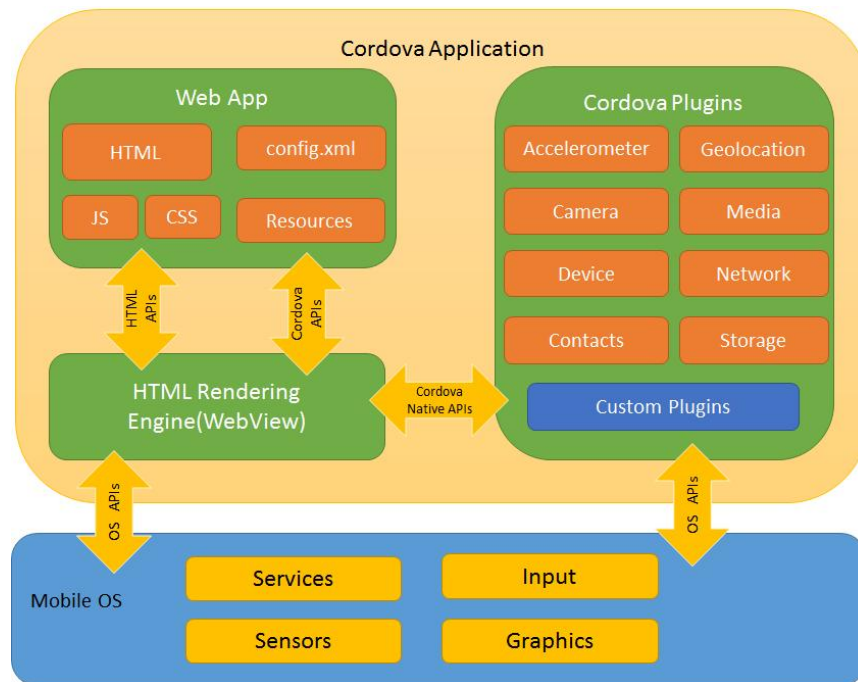
- 1) тестування дещо складніше, оскільки поведінка платформи при одній і тій ж події відрізняється від інших. Це є потенційним місцем де можуть міститись неочевидні та слабо помітні помилки.
- 2) часто використовувані засоби не можуть забезпечити повним, передбаченим системою, функціоналом. Але такі обмеження вирішуються написанням відсутніх частин платформно-орієнтованому коді.
- 3) кожна платформа має унікальний графічний інтерфейс, з яким програма не завжди гармонічно вписуватиметься. Слідування вимогам дизайну платформи робить програму більш передбаченою і послідовною, простішою у використанні.
- 4) суттєвою проблемою також є швидкодія та ефективність таких програм. Інтерпретатори та віртуальні машини повинні бути трансльовані в виконавчий код кожного разу коли програма запускається. Це потребує накладних витрат.

Основними критеріями вибору способу написання прикладних додатків є: знання платформ на яких вони повинні працювати, врахування можливих складностей реалізації функціоналу на тій чи іншій платформі, бюджет розробки.

У мобільній розробці використовуються мови програмування Objective-C, Swift, Java, C# – такі програми називають рідними (від англ. – native) або платформно-орієнтованими. Додатки, що пишуться не лише на рідному коді, зокрема на JavaScript, називаються гібридними.

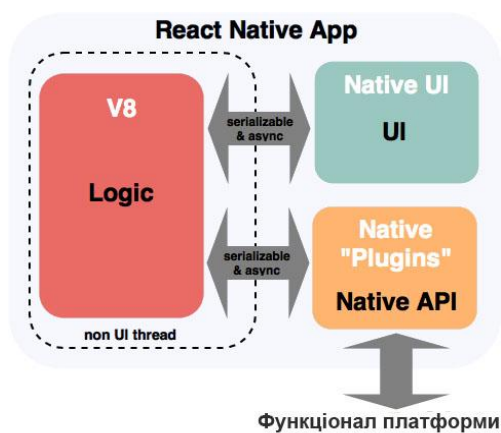
Такий підхід широко застосовується у багатьох програмах, в тому числі Facebook, Instagram, Uber, Google Maps, Discovery VR. Для розробки створені спеціальні набори інструментів: Apache Cordova, PhoneGap, React Native, Electron, Phaser, Titanium. Частина цих технологій використовує спеціальний компонент WebView (UIWebView у iOS), що дозволяє вміщувати веб-сторінки і викликати функції з рідного коду платформи. Він забезпечує роботу звичних веб-технологій всередині додатків.

Apache Cordova формує у WebView контент, для взаємодії з ресурсами платформ (такі як камера, мережа, сховище зберігання, контакти та ін.) передбачені плагіни. Схематичне зображення архітектури, що пропонує Cordova, наведено нижче.



Бібліотека Native script дозволяє робити мультиплатформні додатки, використовуючи XML, CSS, JavaScript. Native script для формування представлення використовує елементи рідного інтерфейсу. Доступ до таких ресурсів як камера, gps забезпечується з коробки. Варто зауважити, що такі додатки працюватимуть на пристроях починаючи з Android 4.2 і вище, та iOS 7.1 і вище.

В React Native (RN) на противагу Apache Cordova логіка програма пишеться і працює на JavaScript, при тому, що інтерфейс повністю рідний. В Cordova при виникненні нової події потік блокується і передається управління на JavaScript-код, очікуючи його інструкцій, React Native виконує код в окремому фоновому потоці, взаємодіючи з головним потоком асинхронно, тобто в потоці збирається ряд команд до головного і певний момент відправляється згрупований запит. Загальна схема роботи зображена нижче.



React Native забезпечує компонентний підхід до розробки, завдяки цьому додатки легко розширювати як власними компонентами, так і великою кількістю сторонніх. Передбачені такі готові компоненти: `ActivityIndicator`, `Button`, `DatePickerIOS`, `DrawerLayoutAndroid`, `Image`, `KeyboardAvoidingView`, `ListView`, `MapView`, `Modal`, `Navigator`, `NavigatorIOS`, `Picker`, `PickerIOS`,

ProgressBarAndroid, ProgressViewIOS, RefreshControl, ScrollView, SegmentedControlIOS, Slider, SnapshotViewIOS, StatusBar, Switch, TabBarIOS, TabBarIOS.Item, Text, TextInput, ToolbarAndroid, , TouchableWithoutFeedback, View, ViewPagerAndroid, WebView.

Оскільки платформи різні, то і компоненти, відповідно, відрізняються. Завдяки компонентам, що є абстракціями над рідним функціоналом, розмітка HTML та стилі CSS не використовуються для формування представлення. Це забезпечує більшу продуктивність ніж при використанні WebView. У випадку відсутності потрібного компоненту необхідно створювати власний на рівні платформи.

RN підтримує дві мобільні платформи – iOS та Android. Для розробки доступні як спільні для обох систем компоненти, так і індивідуальні.

React Native містить вбудований транслятор JavaScript специфікації ECMAScript 2015, завдяки якому код, написаний на найновішій реалізації, трансформується у поточний.

Також доступне зручне налагодження програм в браузері Chrome і миттєве перезавантаження. Для цього потрібно відкрити меню розробника і поставити галочки в пунктах Debug in Chrome і Enable Live Reload. Після чого можна побачити вивід console.log і відслідковувати помилки з консолі браузера. А миттєве перезавантаження пришвидшує та спрощує процес розробки, оскільки зроблені зміни в коді негайно застосовуються.

Нижче наведено приклад коду, що формує текст по центру екрану.

```
import React from 'react';
import {
  AppRegistry,
  StyleSheet,
  Text,
  View
} from 'react-native';

class App extends React.Component {
  render() {
    return (
      <View style={styles.container}>
        <Text style={styles.mainText}>Simple text</Text>
      </View>
    )
  }
}

var styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
  },
  mainText: {
    fontSize: 20,
    textAlign: 'center',
  },
});

AppRegistry.registerComponent('App', () => App);
```

При розробці важливо передбачати розвиток функціоналу програми, і слідувати підходящій архітектурі. Компоненти повинні бути пов'язаними між собою правильним чином. Робота компонентів залежить від подій та даних. Існують два типи даних, що керуються компонентом: властивості props і стан state. Властивості props встановлюються батьківським компонентом і є незмінними протягом всього життєвого циклу компонента. Для даних, які можуть змінюватися з плином часу, потрібно використовувати state. Для спрощення управління потоком даних можна використовувати контейнер стану, такий як Redux.

Для настільних операційних систем також існує кілька інструментів створення програм на JavaScript. Гібридні програми для настільних комп'ютерів використовують Webkit та Node.js.

Electron містить в собі браузер Chromium та Node.js. Обробка коду та генерація представлення відбувається в Chromium, а робота з операційною системою – через Node.js.

Принцип роботи Electron заснований на двох типах процесів:

- 1) основний процес, який відповідає за інтеграцію та взаємодію з графічним інтерфейсом операційної системи. Такий процес може бути запущений тільки один раз на весь період роботи додатка.
- 2) процес формування представлення, що відповідає за відображення вікна браузера, це може бути сторінка додатка або будь-яка інша веб-сторінка. Таких процесів може бути декілька. За рендеринг вмісту відповідає основний процес.

На відміну від мобільних додатків, розмір таких програм буде значно більшим, оскільки міститиме повноцінний веб-браузер.

Підтримуються операційні системи OSX, Linux, Windows.

Підсумовуючи вище написане, можна сказати, що розробка мультиплатформних програм з використанням JavaScript є доцільною у багатьох випадках.

## Література

1. React Native [Electronic resource] / React Native. – Mode of access: <https://facebook.github.io/react-native/>
2. Kyle Simpson. You Don't Know JS: ES6 & Beyond. – O'Reilly Media, 2015. – 228 p.
3. Bonnie Eisenman. Building Native Mobile Apps with JavaScript, 2015. – 272 p.
4. Architectural overview of Cordova platform [Electronic resource] / Apache Cordova. – Mode of access: <https://cordova.apache.org/docs/en/latest/guide/overview/index.html>

УДК 514.18

UDC 514.18

Бурчак І.Н., Головачук І.П., Величко В.Л.

Луцький національний технічний університет

## **ВПРОВАДЖЕННЯ ДОДАТКОВИХ СЦЕНАРІЇВ ТЕСТУВАННЯ СТУДЕНТІВ У ПЛАТФОРМИ MOODLE ЗГІДНО З ПОТРЕБАМИ КАФЕДРИ ІКГ**

**Бурчак І.Н., Головачук І.П., Величко В.Л. Впровадження додаткових сценаріїв тестування студентів у платформі MOODLE згідно з потребами кафедри ІКГ.** В статті розглянута можливість впровадження додаткових сценаріїв тестування студентів у оболонці дистанційного навчання MOODLE, що надало можливість використання запитань для тестів з великою кількістю графічних зображень. Розглянуто та приведено приклад з рекомендаціями по підготовці одного із сценаріїв.

**Ключові слова:** сценарії тестування, оболонка MOODLE, дистанційне навчання, електронний засіб навчального призначення, система управління навчанням, категорії курсів, банк запитань.

**Бурчак И.Н., Головачук И.П., Величко В.Л. Внедрение дополнительных сценариев тестирования студентов на платформе MOODLE в зависимости от нужд кафедры ИКГ.** В статье рассмотрена возможность внедрения дополнительных сценариев тестирования студентов в оболочке дистанционного обучения MOODLE, что позволило использование вопросов для тестов с большим количеством графических изображений. Рассмотрен и приведен пример с рекомендациями по подготовке одного из сценариев.

**Ключевые слова:** сценарии тестирования, оболочка MOODLE, дистанционное обучение, электронное средство учебного назначения, система управления обучением, категории курсов, банк вопросов.

**Burchak I.N., Golovachuk I.P., Velychko V.L., Implementation of additional scenarios of the students testing on platform MOODLE in accordance with the needs of the department ECG.** Article consider implementing additional scenarios of students testing in LMS MOODLE, which made helpful use the questions to test with lots of graphics. The example considered and recommendations for the preparation of one of the scenarios.

**Keywords:** testing scripts, platform MOODLE, distance learning, e-learning tool for educational purposes, learning management system, course category, a bank of questions.

**Постановка проблеми.** В даний час значно зростає роль використання дистанційного навчання [1]. Особливу увагу на нього звернули в останні роки по причині подовження для студентів зимових канікул. Багато "вишів" використовує потужну, відкриту навчальну платформу MOODLE [2] також і для електронної підтримки навчального процесу денної форми навчання. Згідно з визначенням [2] на сайті спільноти користувачів MOODLE:

"...Moodle (Modular Object-Oriented Dynamic Learning Environment, вимовляється «Мудл») - це модульне об'єктно-орієнтоване динамічне навчальне середовище, яке називають також системою управління навчанням (LMS), системою управління курсами (CMS), віртуальним навчальним середовищем (VLE) або просто платформою для навчання, яка надає викладачам, учням та адміністраторам дуже розвинутий набір інструментів для комп'ютеризованого навчання, в тому числі дистанційного.

Moodle можна використовувати в навчанні школярів, студентів, при підвищенні кваліфікації, бізнес-навчанні, як в комп'ютерних класах навчального закладу, так і для самостійної роботи вдома.

Moodle - це найбільш досконала і поширена в Україні і в світі система такого призначення. На даний момент Moodle вже має 65 мільйонів користувачів в усьому світі й продовжує розвиватися темпами, значно швидшими, ніж її конкуренти. Тобто обрати для впровадження в навчальному закладі саме Moodle - це те саме, що обрати, як іноземну мову для вивчення - англійську.



Moodle - це безкоштовна, відкрита (Open Source) система. Вона не лише безкоштовна сама, а й не потребує для своєї роботи жодного платного програмного забезпечення. Тобто кожний навчальний заклад може впровадити у себе не просто безкоштовну і найбільш досконала, а ще й абсолютно ліцензійну систему, не витративши жодної копійки на придбання програмного забезпечення. При цьому він може вносити зміни у код у відповідності до своїх потреб".



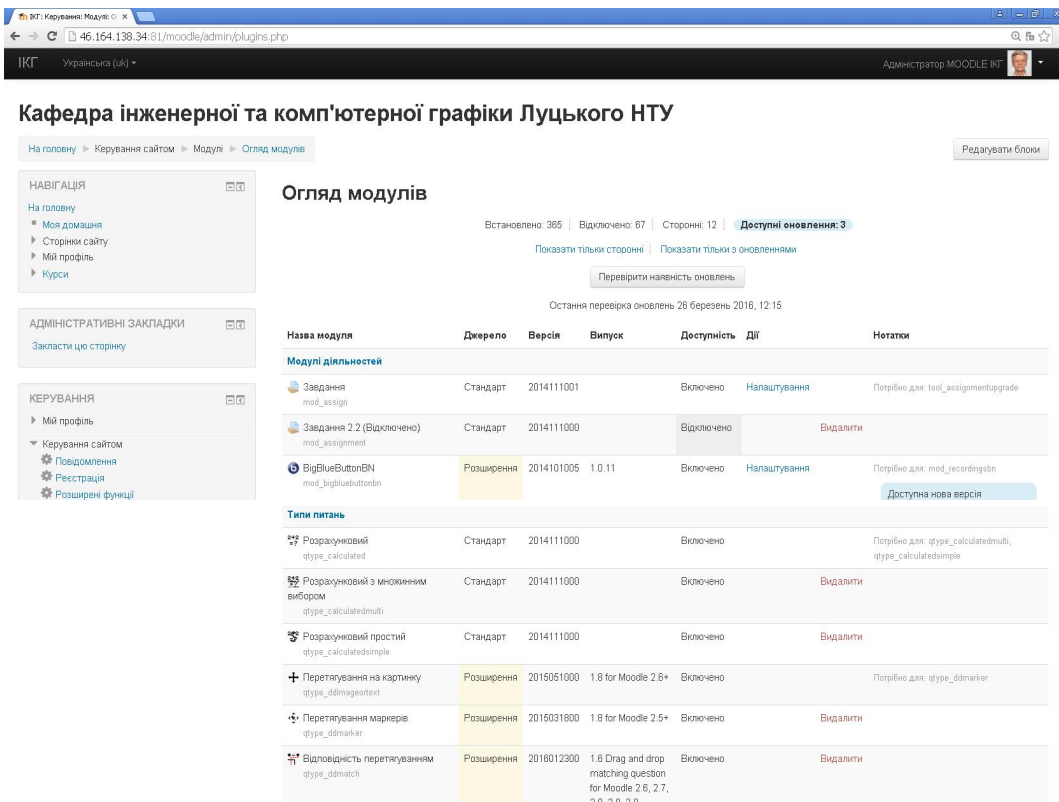
Рисунк 1. Ресурс української спільноти користувачів Moodle

### Видані в Україні книги про Moodle

розміщені тут з дозволу авторів, яким ми за це щиро вдячні

-  Анисимов А.М. Работа в системе дистанционного обучения Moodle.
-  Триус Ю.В., Герасименко І.В., Франчук В.М. Система електронного навчання ВНЗ на базі Moodle

Рисунк 2. Книги, розміщені на сайті Moodle





Рисунк 3. Встановлені додаткові модуллі на Moodle кафедри ІКГ

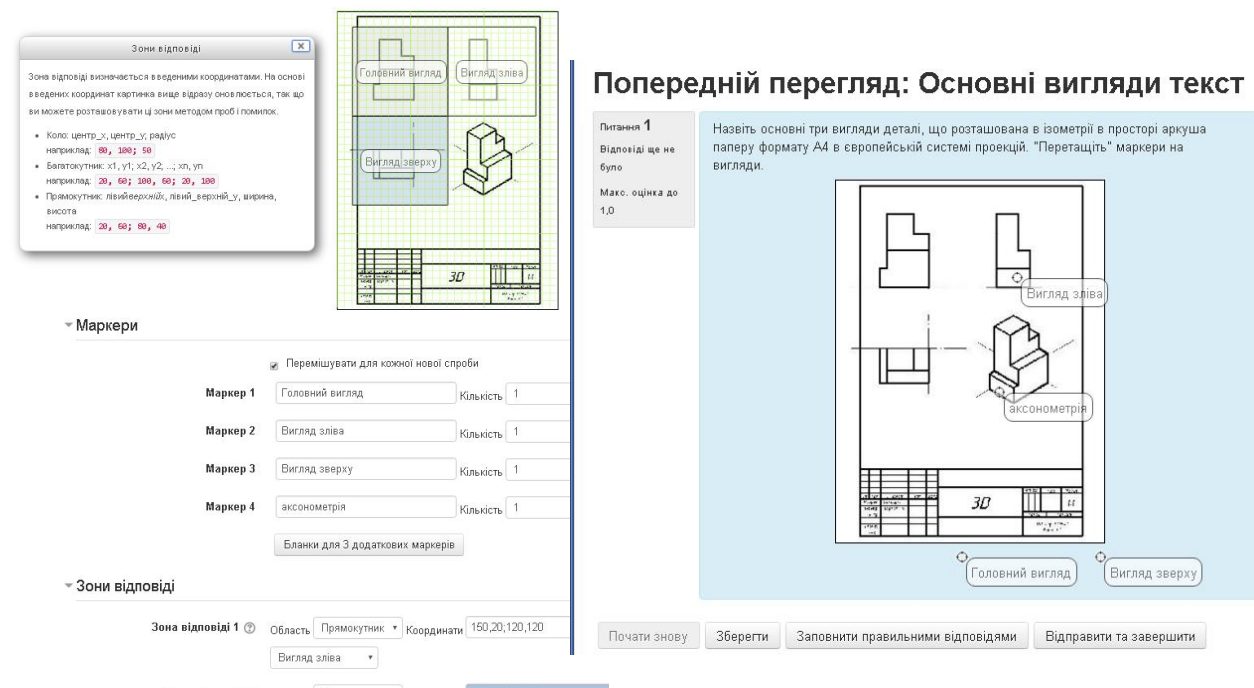
**Аналіз останніх досліджень та публікацій.** У роботах, розташованих на цьому ж сайті, зокрема [3], приведено детальний опис для користувачів LMS (системи управління навчанням) Moodle (рис.2). За допомогою цієї системи, використовуючи тільки стандартні модулі, можна: використовувати



Інтернет для комунікації та закачування матеріалів, надсилати повідомлення студентам, тестувати студентів, вести електронні журнали обліку оцінок та відвідування, налаштовувати різноманітні ресурси курсу, тощо [5]. Але у деяких випадках використання тільки стандартних модулів (плагінів) є недостатнім для підтримки навчального процесу в освітньому закладі, зокрема при проведенні тестів із графічних дисциплін. Тому для впровадження системи MOODLE у навчальний процес на кафедрі Інженерної та комп'ютерної графіки Луцького НТУ було впроваджено додаткові модулі (рис.3).

**Постановка завдання.** Завданням даної роботи є дослідження додаткових модулів, які дають змогу використати нові (нестандартні) сценарії тестування студентів з точки зору використання графічних зображень та трудоемкості (часу) виготовлення таких питань. Для поліпшення якості навчального процесу чи полегшення підготовки викладачами тестів, розглянуто можливість застосування вже готових додаткових сценаріїв тестів, які можна встановити (добавити) до існуючих – стандартних.

**Основна частина.** Розглянемо на прикладах два типи додаткових сценаріїв тестування студентів – "перетягування маркерів"  та "перетягування на картинку" .



**Зони відповіді**

Зона відповіді визначається введеними координатами. На основі введених координат картинка зображення оновлюється, так що ви можете розташувати ці зони методом проб і помилок.

- Коло: центр\_x, центр\_y, радіус  
наприклад: 80, 100, 50
- Багатокутник: x1, y1, x2, y2, ..., xn, yn  
наприклад: 20, 60; 100, 60; 20, 100
- Прямокутник: лівий\_верхній\_x, лівий\_верхній\_y, ширина, висота  
наприклад: 20, 60; 80, 40


**Маркери**

Перемішувати для кожної нової спроби

Маркер 1	Головний вигляд	Кількість	1
Маркер 2	Вигляд зліва	Кількість	1
Маркер 3	Вигляд зверху	Кількість	1
Маркер 4	аксонометрія	Кількість	1

Бланки для 3 додаткових маркерів

**Зони відповіді**

Зона відповіді 1  Область: Прямокутник Координати: 150,20;120,120

Вигляд зліва

**Попередній перегляд: Основні вигляди текст**

Питання 1  
Відповіді ще не було  
Макс. оцінка до 1,0

Назвіть основні три вигляди деталі, що розташована в ізометрії в просторі аркуша паперу формату А4 в європейській системі проєкцій. "Перетягніть" маркери на вигляди.

Головний вигляд Вигляд зліва аксонометрія Вигляд зверху

Почати знову Зберегти Заповнити правильними відповідями Відправити та завершити

Рисунок 4. Назви маркерів, зони відповідей, попередній перегляд запитання

### Перетягування маркерів – дуже подібне на використання навігаційних карт в HTML.

Вибираємо тип запитання, заповнюємо завдання: "Назвіть основні три вигляди деталі, що розташована в ізометрії в просторі аркуша паперу формату А4 в європейській системі проєкцій. для цього "перемістіть" відповідні маркери на вигляди", вибираємо фоновий рисунок. Розмір растрового рисунка, який використовується, як фон не повинен перевищувати 600x400 пікселів (рис.4). І у секції маркери задаються слова, які ми хочемо щоб студенти розмістили у створених зонах на зображенні. Якщо вказується у вікні "кількість" – 1, то кожен із заданих маркерів може бути використаний лише один раз. Далі переходимо до зон, на які розбивається фонове зображення. Зони, подібно до HTML, можуть бути заданими трьома способами – багатокутником, колом, прямокутником. Секція зони відповіді містить у собі довідку для полегшення задання зони у координатах (рис.4 – зліва). Також на фоновий рисунок наноситься сітка з коміркою 10x10 пікселів. Відразу після задання – зони виділяються та підсвічуються відповідним кольором та маркери розташовуються у відповідних зонах. Попередній перегляд тесту приведений на рис.4, справа. Маркери легко переміщуються по всій зоні рисунка і не мають прив'язки. Такий тип запитання є дуже зручним для студентів, однак для викладачів у процесі підготовки він є досить трудомістким. Більш зручним у підготовці є тип перетягування на картинку, однак і надписи і зображення при перетягуванні мають об'єктну прив'язку.

**Перетягування на картинку** – сценарій запитань, який надає можливість використовувати та переміщати не тільки текстові надписи, але і зображення на визначені зони фонового рисунка (рис.5). Цей тип запитань може використовувати і змішані переміщення зображень і тексту. Розмір фонового рисунка, як і в попередньому варіанті – обмежений 600x400 пікселів. Обмеження також мають графічні зображення, які переміщуються – їх розмір не повинен перевищувати 150x100 пікселів. Для прикладу ми взяли ту ж саму роботу і в растровому редакторі із кресленика вирізали три основних вигляди. Тепер запитання звучить так: "Виберіть та розташуйте основні види деталі, яка зображена в аксонометрії на аркуші паперу формату А4, назвіть їх - для цього "перемістіть" відповідні маркери на вигляди". Надалі завантажуюмо фоновий рисунок та долучаємо в розділ "елементи перетягування" зображення для переміщення (рис.5).

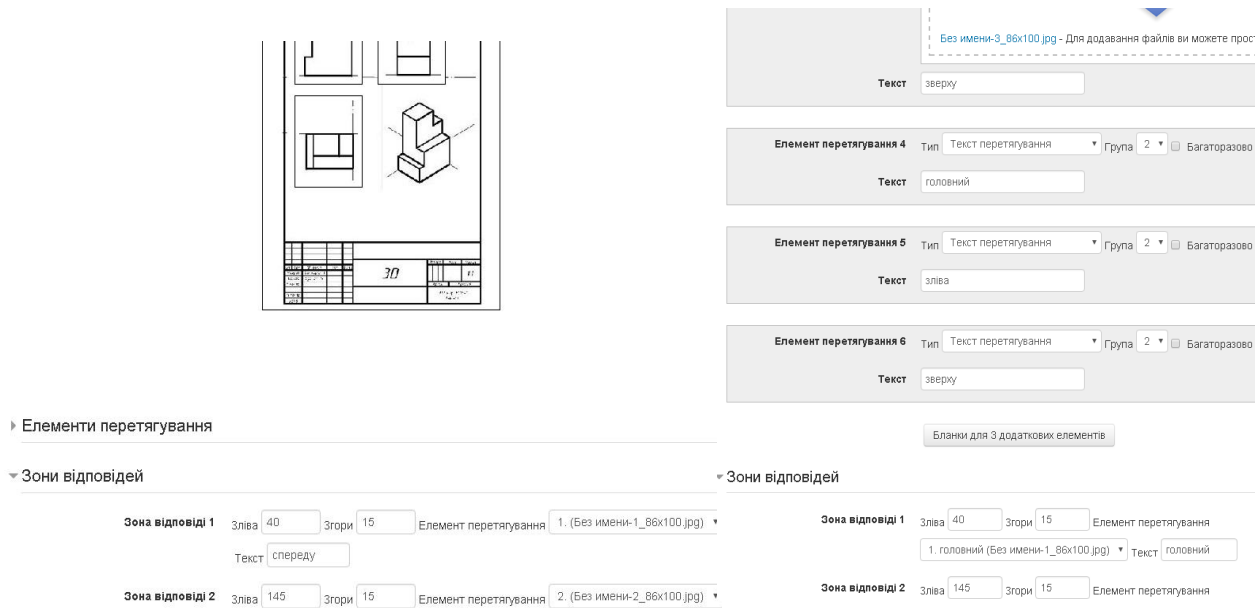


Рисунок 5. Розміщення графічних елементів та додавання другої групи – текстових елементів для перетягування.

Розміщення графічних елементів на тлі рисунку автоматично формує координати зон відповідей, рекомендується їх тільки коригувати. Справа на рисунку показано задання другої групи відповідей – тексту для перетягування. Текстові елементи аналогічно переміщуються на фоновий рисунок і автоматично формують координати нових зон відповідей. Приклади попереднього перегляду тесту та правильно розташовані відповіді приведені на рисунку 6. При використанні даного алгоритму тестового питання, як графічні так і текстові елементи розташовуються на фонному рисунку з меншими трудозатратами, проте вони мають прив'язку до фіксованого положення. Але і при різній величині розміру елемента, який переміщається, вони можуть один одного заміщати, що полегшує роботу з ними та скорочує процес підготовки тестових запитань.

**Висновки та перспективи.** Результатом даної роботи є дослідження зручності та трудомісткості додаткових сценаріїв тестування для потреб кафедри Інженерної та комп'ютерної графіки. Адже кафедра використовує величезну кількість зображень у процесі підготовки фахівців. Креслення є мовою будь-якого інженера і процес його створення є дуже трудомістким. Таким же трудомістким є і процес підготовки електронних навчальних засобів. Пошук, розробка, використання та тестування таких сценаріїв значно покращує якість навчального процесу.



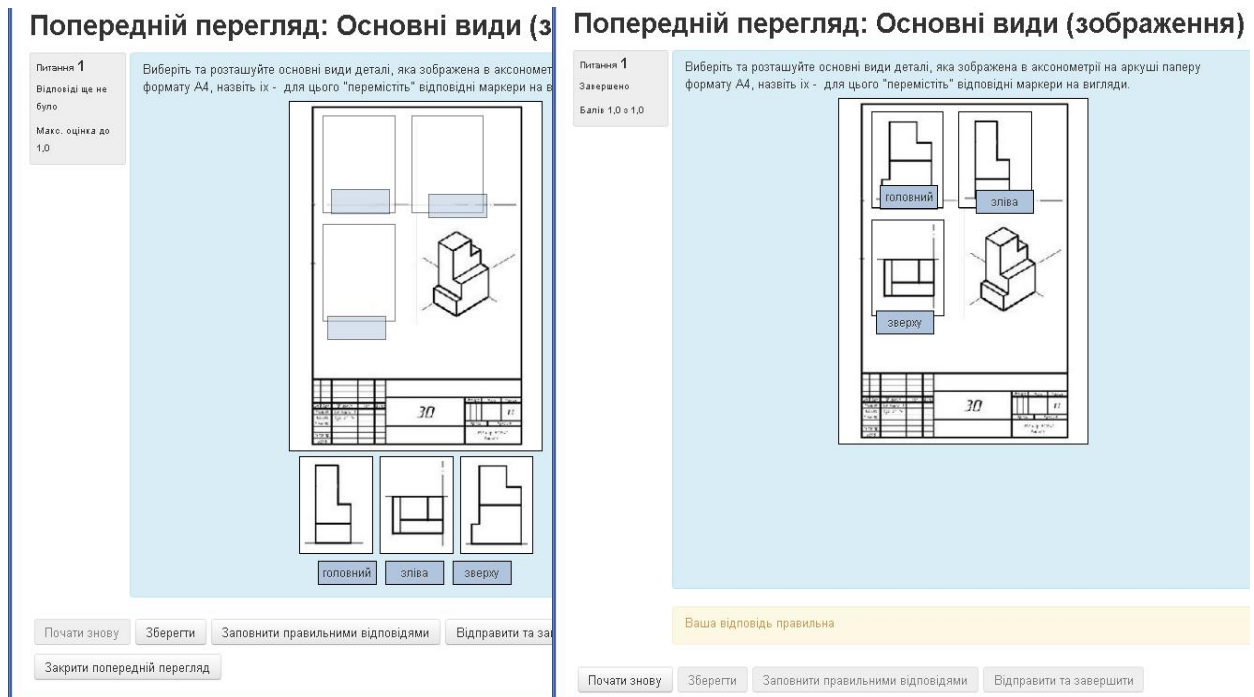


Рисунок 6. Попередній перегляд тесту

1. Про затвердження Положення про дистанційне навчання МОН України; Наказ, Положення від 25.04.2013 № 466 [Електронний ресурс] режим доступу: <http://zakon4.rada.gov.ua/laws/show/z0703-13> . – (дата звернення: 03.12.2016). – Назва з екрана.
2. Ресурс української спільноти користувачів Moodle [сайт] режим доступу: <https://moodle.org/mod/page/view.php?id=8174> . – (дата звернення: 03.12.2016). – Назва з екрана.
3. Система електронного навчання ВНЗ на базі MOODLE: Методичний посібник / Ю. В. Триус, І. В. Герасименко, В. М. Франчук // За ред. Ю. В. Триуса. – Черкаси. – 220 с. [Електронний ресурс] режим доступу: [https://moodle.org/pluginfile.php/1968620/mod\\_resource/content/1/Триус СЕН ВНЗ Moodle 2013.pdf](https://moodle.org/pluginfile.php/1968620/mod_resource/content/1/Триус%20СЕН%20ВНЗ%20Moodle%202013.pdf) – (дата звернення: 03.12.2016). – Назва з екрана.
4. Бурчак І. Н. Розвиток засобів дистанційної освіти кафедрою інженерної та комп'ютерної графіки ЛНТУ./ Бурчак І. Н., Бойко Л.С. // Наукові записки Тернопільського національного педагогічного університету. Серія: Педагогіка.-2009.-№3.-325с.
5. Бурчак І. Н. Методика організації курсу з дисципліни "НГІтаКГ" у середовищі дистанційного навчання "MOODLE"/ Бурчак І.Н., Величко В.Л., Ройко О.Ю.// Матеріали Всеукраїнської науково-методичної конференції "Викладання дисципліни "Нарисна геометрія, інженерна і комп'ютерна графіка" в умовах кредитно-модульної системи навчання м. Севастополь, 7 – 11 лютого 2011 року. – СевНТУ. – С.42-44

УДК 004.4:621.395:336

Єрмейчук С. Ю., Федонюк А.А. к.ф-мат.наук, доцент  
СНУ імені Лесі Українки

## РОЗРОБКА АГРЕГАТОРА КУРСА ВАЛЮТ ДЛЯ ОС ANDROID

**Єрмейчук С. Ю., Федонюк А.А. Розробка агрегатора курса валют для ОС Android.** Програмування для мобільних пристроїв – важлива галузь інформаційних технологій. Для цього сегмента прогнозують швидкий ріст у міру того, наскільки мобільні програми будуть підтримані технічними можливостями мобільних пристроїв. Інформаційні технології (ІТ) зазнали глобального поширення у всіх галузях: від освіти і до медицини. Однією з найбільш популярних та таких, що швидко зростає в ІТ є галузь *wearable technology*, що в перекладі на українську мову – носимі технології. Зазвичай це предмети одягу або аксесуари (все те, що людина може носити на собі), що є одночасно і електронними пристроями. Зазвичай такого роду пристрої називають носимими комп'ютерами (*wearable computers, wearables*).

**Ключові слова:** **Android; віджети; мобільний додаток; Android Studio; агрегація; бібліотека Volley; Android SDK.**

**Єрмейчук С. Ю., Федонюк А.А. Разработка агрегатора курса валют для ОС Android.** Программирование для мобильных устройств - важная отрасль информационных технологий. Для этого сегмента прогнозируют быстрый рост по мере того, как мобильные приложения будут поддержаны техническими возможностями мобильных устройств. Информационные технологии (ИТ) получили глобального распространения во всех областях: от образования и к медицине. Одной из наиболее популярных и таких быстро растущих в ИТ является отрасль *wearable technology*, что в переводе - носимые технологии. Обычно это предметы одежды или аксессуары (все то, что человек может носить на себе), что является одновременно и электронными устройствами. Обычно такого рода устройства называют носимый компьютер (*wearable computers, wearables*).

**Ключевые слова:** **Android; виджеты; мобильное приложение; Android Studio; агрегация; библиотека Volley; Android SDK.**

**Yermeychuk S.Y, Fedonyuk A.A Development aggregator exchange rates for Android OS.** Programming for mobile devices – is an important area of informational technology. This segment predicts rapid growth as how mobile applications will be supported by the technical capabilities of mobile devices. Informational technology (IT) has global distribution in all areas, from education to medicine. One of the most popular and the fastest growing in the IT industry is a *wearable technology*, which is translated into Ukrainian as the technology you can take with yourself. Usually it is a part of clothes or accessories (everything that people can carry with themselves), which is an electronic device at the same time. Usually this type of device is called as a *wearable computer (wearable computers, wearables)*.

**Keywords:** **Android; widgets; mobile app; Android Studio; aggregation; Library Volley; Android SDK.**

**Постановка наукової проблеми та її значення.** У сучасному світі мобільні електронні пристрої поширюються дуже швидко. Потужності збільшуються в геометричній прогресії. Ринок техніки все більше тяжіє до портативних пристроїв, тому програмісти розробляють для них велику кількість професійних додатків. Раніше мобільні додатки використовувалися для швидкої перевірки електронної пошти, але їх високий попит призвів до розширення їхніх призначень і в інших областях, таких як ігри для мобільних телефонів, GPS, спілкування, перегляд відео та користування інтернетом.

Мобільний додаток – це програмне забезпечення, призначене для роботи на смартфонах, планшетах та інших мобільних пристроях. Багато мобільних додатків вже встановлені на самому пристрої або можуть бути завантажені на нього з додатків онлайн-магазинів, таких як App Store, BlackBerry App World та інших. Тому створення мобільних додатків на ОС Android - популярний напрям сучасних ІТ-технологій. ОС Android встановлена на мільйонах девайсів: смартфонах, комунікаторах, планшетах, нетбуках та ін. Її особливість в тому, що це відкрита та безкоштовна платформа, на яку кожен може встановити різноманітні ігри, додатки для бізнесу, спілкування та відеозв'язку, медіа ресурсів та ЗМІ, онлайн-ресурсів.

Метою нашого дослідження є створення Android-додатку для перегляду курсів валют через встановлене з'єднання з мережею Internet У відповідності до поставленої мети визначено такі завдання дослідження: ознайомитись з операційною системою Android та її специфікаціями; розглянути процес моделювання роботи Android-додатку перегляду курсів валют; організувати збір статистичних даних з банків; розглянути механізм передачі даних через мережу Інтернет; розглянути і дослідити роботу стандарту передачі даних JSON; розробити програмний додаток на базі ОС Android; забезпечити зручне подання даних.

**Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.** Android – операційна система від Google для мобільних пристроїв, таких як планшетні

комп'ютери, комунікатори, мобільні телефони, камери та багато інших. Вона створена на базі ядра Linux. Назва «Android» в перекладі з англійської означає «людиноподібний робот». Спочатку операційна система розроблялася компанією Android Inc., яку згодом купила Google. Потім Google створила Open Handset Alliance - організацію, яка займається підтримкою і розвитком платформи. Розробка додатків для Android здійснюється, як правило, на мові Java, але можна використовувати і інші мови програмування. З моменту виходу першої версії у вересні 2008 року відбулося 40 оновлень системи. Ці оновлення, як правило, стосуються виправлення виявлених помилок і додавання нової функціональності в систему. Остання версія, на момент написання роботи — 6.0. В таблиці 1 наведені дати виходу основних версій Android.

Платформа Android представляє собою програмний стек для мобільних пристроїв, який включає операційну систему, програмне забезпечення проміжного шару (middleware), а також основні користувальницькі додатки, що входять до складу мобільного телефону, календар, карти, браузер, бази даних контактів, повідомлень SMS та ін.

Архітектура Android складається з чотирьох рівнів:

1. рівень ядра;
2. рівень бібліотек і середовища виконання;
3. рівень каркаса додатків (application framework);
4. рівень додатків.

Таблиця 1 - Список версій OS Android

Версія	Версія ядра Linux	Дата виходу	Кодова назва
1.0	2.6.25	23 вересня 2008 року	Apple Pie(API 1)
1.1	2.6.25	09 лютого 2009 року	Banana Bread (API 2)
1.5	2.6.27	30 квітня 2009 року	Cake(API 3)
1.6	2.6.29	15 вересня 2009 року	Donut(API 4)
2.0.x/2.1	2.6.29	27 жовтня 2009 року 12 січня 2010 року	Eclair(API 5,6,7)
2.2.x	2.6.32	20 травня 2010 року	Froyo(API 8)
2.3.x	2.6.35	06 грудня 2010 року	Gingerbread(API 9,10)
3.0	2.6.36	03 лютого 2011 року	Honeycomb(API 11,12,13)
4.0	3.0	19 жовтня 2011 року	Ice Cream Sandwich (API 14,15)
4.1 4.2 4.3	3.0.8-3.4 (4.2.1+)	27 червня 2012 року 29 жовтня 2012 року 24 липня 2013 року	Jelly Bean(API 16,17,18)
4.4	3.4	31 жовтня 2013 року	KitKat(API 19)
5.0	3.4	03 листопада 2014 року	Lollipop(API 21)
5.1	3.4	9 березня 2015 року	LolliPop(API 22)
6.0	3.10	28 травня 2015 року	Marshmallow(API 23)

На рисунку 1. показані основні компоненти операційної системи Android і їх взаємодія.

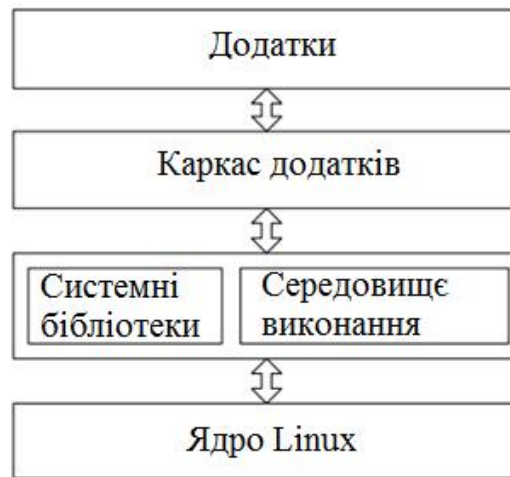


Рисунок 1 - Архітектура системи Android

На рівні ядра розташовуються основні служби типу управління процесами, розподілу пам'яті і управління файловою системою. Android заснований на ядрі Linux версії 2.6, але сама система Android не є Linux-системою в «чистому вигляді». Система Android має деякі відмінності і містить додаткові розширення ядра Linux, специфічні для Android, - свої механізми розподілу пам'яті, взаємодія між процесами і ін.

Наступний рівень над ядром Linux включає набір бібліотек C/C++, використовуваних різними компонентами ОС. Бібліотеки цього рівня за своїм функціональним призначенням можна розділити на дві групи:

- системну бібліотеку C;
- функціональні бібліотеки C/C ++.

Системна бібліотека базується на Berkeley Software Distribution (BSD). Компанія Google розробила власну версію системної бібліотеки libc – Bionic спеціально для мобільних пристроїв на основі Linux. Це було необхідно для забезпечення швидкого завантаження бібліотеки в кожен процес, і отже, бібліотека повинна була мати маленький розмір. Бібліотека Bionic має розмір близько 200 Кбайт, що в два рази менше розміру стандартної бібліотеки Linux glibc. Даний рівень представлений бібліотеками libc (в Android вона називається Bionic), OpenGL (підтримка графіки), WebKit (відображення Web-сторінок), FreeType (підтримка шрифтів), SSL (зашифровані з'єднання), SGL (2D-графіка), бібліотеки підтримки SQLite, Media Framework (підтримка мультимедіа). На цьому ж рівні працює Dalvik Virtual Machine - віртуальна машина Java, яка надає необхідну функціональність для Java-додатків.

Рівень каркаса додатків знаходиться на вершині системних бібліотек, функціональних бібліотек і Dalvik VM. На цьому рівні знаходяться основні служби Android для управління життєвим циклом додатків, пакетами, ресурсами і т.д.

Мобільний пристрій Android поставляється з набором основних додатків, включаючи поштового клієнта, програму для роботи з SMS, календар, навігаційні карти, браузер, контакти та ін [1].

Неможливо створити осмислений додаток, не вивчивши внутрішню організацію, властиву додаткам, які працюють на певній платформі. В першу чергу, необхідно проаналізувати структуру та основні компоненти додатків, що розробляються для роботи на пристроях під управлінням ОС Android. Від типу мобільного пристрою внутрішня організація додатків не залежить, тому Android-додатки, що розроблені для смартфонів, цілком зможуть виконуватися і на планшетах. При проектуванні та розробці програмного забезпечення має велике значення його швидкість та ефективність роботи. Дуже важливо постійно приділяти увагу оптимізації програмного забезпечення на всіх етапах його розробки. Для цього потрібно вимірювати час виконання елементів програмного коду, аналізувати хід роботи програмного забезпечення, виділяти обмежені ділянки програмного коду, слідкувати за виділенням пам'яті та часом життя об'єктів.

Приступаючи до розробки мобільних додатків, якщо вдасться визначити до якого типу належить додаток, то стане зрозуміло на які моменти в процесі його розробки необхідно звертати основну увагу. Наступні види додатків:

1) Змішані додатки більшу частину часу працюють у фоновому режимі, проте допускають взаємодію з користувачем після. Зазвичай взаємодія з користувачем зводиться до повідомлення про які-небудь події. Прикладами таких додатків можуть служити мультимедіа-програвачі, програми для обміну текстовими повідомленнями (чати), поштові клієнти. Можливість реагувати на користувача введення і при цьому не втрачати працездатності у фоновому режимі є характерною особливістю змішаних додатків. Такі програми зазвичай містять як видимі активності, так і приховані (фонові) сервіси, і при взаємодії з користувачем повинні враховувати свій поточний стан.

2) Фонові програми після настройки не припускають взаємодії з користувачем, більшу частину часу перебувають і працюють в прихованому стані. Прикладами таких додатків можуть служити, служби екранування дзвінків, SMS-автовідповідачі. У більшості фонові програми націлені на відстеження подій, породжуваних апаратним забезпеченням, системою або іншими додатками, працюють непомітно.

3) Додатки переднього плану виконують свої функції тільки, коли видимі на екрані, в іншому ж випадку їх виконання призупиняється. Такими додатками є, наприклад, ігри, текстові редактори, відеопрогравачі.

4) Віджети - невеликі додатки, які відображаються у вигляді графічного об'єкта на робочому столі. Прикладами можуть служити, додатки для відображення динамічної інформації, такої як заряд батареї, прогноз погоди, дата і час.

Як вже було зазначено, програми під Android розробляються на мові програмування Java, компілюються у файл з розширенням .apk. Після цього цей файл використовується для установок програм на пристрої, що працюють під управлінням Android. Після установки кожен Android-додаток функціонує у своїй власній безпечній області. Розглянемо, як це виглядає:

- операційна система Android є багатокористувацькою ОС, в якій кожен додаток розглядається як окремий користувач;
- за замовчуванням, система призначає кожному з додатків унікальний користувальницький ID, який використовується тільки системою і невідомий самому додатку;
- система встановлює права доступу до всіх файлів додатка наступним чином: доступ до елементів додатку має тільки користувач з відповідним ID;
- кожному додатку відповідає окремий Linux процес, який запускається, як тільки це необхідно хоча б одному компоненту додатку. Процес припиняє роботу, коли жоден компонент програми не використовує його або ж системі потрібно звільнити пам'ять для інших (можливо, більш важливих) додатків;
- кожному процесу відповідає окремий екземпляр віртуальної машини Dalvik. У зв'язку з цим код програми виконується ізольовано від інших додатків.

Перераховані ідеї функціонування програм в ОС Android реалізують «принцип мінімальних привілеїв». Кожному додатку за замовчуванням, дозволений доступ тільки до компонентів, необхідних для його роботи і ніяким більше. Таким чином забезпечується дуже безпечне середовище функціонування додатків. Однак, у випадку необхідності додатки можуть отримати доступ до даних інших додатків і системних сервісів (послуг). У випадку, коли двом додаткам необхідно мати доступ до файлів один одного, їм присвоюється один і той же користувальницький ID. Для економії системних ресурсів такі додатки запускаються в одному Linux процесі і ділять між собою один і той же екземпляр віртуальної машини. В цьому випадку додатки також повинні бути підписані одним сертифікатом. У разі ж, коли додатку потрібно доступ до системних даних, наприклад, контактів, SMS повідомленнями, картам пам'яті, камері, Bluetooth і т.д., користувачеві необхідно дати додатком такі повноваження під час встановлення його на пристрій [2].

Програмний інструментарій Java на базі платформи Android. Для даної розробки вирішено зупинитися на мові Java, як офіційній та найбільш поширеній мові для Android. Для створення додатків за допомогою Java існує докладна документація на офіційному сайті Android розробки, також більшість навчальних матеріалів створено з розрахунком саме на неї.

**Java** — об'єктно-орієнтована мова програмування, випущена компанією Sun Microsystems у 1995 році як основний компонент платформи Java. Зараз мовою займається компанія Oracle, яка придбала Sun Microsystems у 2009 році. Програми на Java транслюються в байт-код, що виконується віртуальною машиною Java, яка обробляє байтовий код і передає інструкції обладнанню як інтерпретатор. Перевага подібного способу виконання програм полягає в повній

незалежності байт-коду від операційної системи і устаткування, що дозволяє виконувати Java-додатки на будь-якому пристрої, для якого існує відповідна віртуальна машина. Іншою важливою особливістю Java є система безпеки завдяки тому, що виконання програми повністю контролюється віртуальною машиною. Будь-які операції, які перевищують встановлені повноваження програми (наприклад, спроба несанкціонованого доступу до даних або з'єднання з іншим комп'ютером) викликають негайне переривання. Слід зауважити, що фактично, більшість архітектурних рішень, прийнятих при створенні Java, було продиктовано бажанням надати синтаксис, схожий з C/C++.

Огляд існуючих середовищ програмування Java. В якості інтегрованого середовища розробки в основному виступає Android Studio. Воно було представлено 16 травня 2013 року на конференції Google I/O, довго перебувало на стадії бета-тесту, а перша стабільна версія вийшла 8 грудня 2014 року. На цей час для завантаження доступні версії Android Studio 1.2 та Android Studio 1.3 Preview [6].

Android Studio прийшов на зміну плагіну ADT для платформи Eclipse. Середовище побудоване на базі продукту IntelliJ IDEA Community Edition, що розробляється компанією JetBrains. Android Studio розвивається в рамках відкритої моделі розробки та поширюється під ліцензією Apache 2.0. Бінарні складання підготовлені для Linux (для тестування використаний Ubuntu), Mac OS X і Windows. Середовище надає засоби для розробки програм не тільки для смартфонів і планшетів, але і для носимих пристроїв на базі Android Wear, телевізорів (Android TV), окулярів Google Glass і автомобільних інформаційно-розважальних систем (Android Auto). Для додатків, спочатку розроблених з використанням Eclipse і ADT Plugin, підготовлений інструмент для автоматичного імпорту існуючого проекту в Android Studio.

Середовище розробки адаптоване для виконання типових завдань, що вирішуються в процесі розробки додатків для платформи Android. У тому числі у середовище включені засоби для спрощення тестування програм на сумісність з різними версіями платформи та інструменти для проектування додатків, що працюють на пристроях з екранами різної роздільності (планшети, смартфони, ноутбуки, годинники, окуляри тощо). Крім можливостей, присутніх в IntelliJ IDEA, в Android Studio реалізовано кілька додаткових функцій, таких як нова уніфікована підсистема складання, тестування і розгортання додатків, заснована на складальному інструментарії Gradle і підтримуюча використання засобів безперервної інтеграції.

Для прискорення розробки додатків представлена колекція типових елементів інтерфейсу і візуальний редактор для їхнього компонування, що надає зручний попередній перегляд різних станів інтерфейсу програми (наприклад, можна подивитися як інтерфейс буде виглядати для різних версій Android і для різних розмірів екрану). Для створення нестандартних інтерфейсів присутній майстер створення власних елементів оформлення, що підтримує використання шаблонів. У середовище вбудовані функції завантаження типових прикладів коду з GitHub. До складу також включені пристосовані під особливості платформи Android розширені інструменти рефакторингу, перевірки сумісності з минулими випусками, виявлення проблем з продуктивністю, моніторингу споживання пам'яті та оцінки зручності використання. У редактор доданий режим швидкого внесення правок. Система підсвічування, статичного аналізу та виявлення помилок розширена підтримкою Android API. Інтегрована підтримка оптимізатора коду ProGuard. Вбудовані засоби генерації цифрових підписів. Надано інтерфейс для управління перекладами на інші мови.

Також можна використовувати Eclipse, IntelliJ IDEA чи NetBeans з відповідними плагінами та за наявності встановленого Android SDK, наприклад для середовища Eclipse випускається плагін Android Development Tools (ADT), що дозволяє створювати додатки та тестувати їх у вбудованому емуляторі операційної системи.

На сьогодні найбільш зручним та логічним рішенням є використання саме Android Studio. Це єдине середовище, що розроблялось спеціально для створення додатків для платформи Android. Воно включає в себе весь необхідний інструментарій для розробки, включаючи емулятор Android Virtual Device та віртуалізатор Intel Hardware Accelerated Execution Manager (HAXM) для прискорення роботи емулятора не потребує жодних плагінів чи налаштувань для роботи та використовує Gradle. До того ж це повністю офіційне середовище, яке розробляється і підтримується компанією Google. Android Studio дозволяє імпорт проектів, які розроблялися за допомогою IDE Eclipse з плагіном ADT [3].

В об'єктно-орієнтованому програмуванні під агрегуванням розуміють методику створення нового класу з уже існуючих класів шляхом включення.

Вкладені об'єкти нового класу зазвичай оголошуються закритими, що робить їх недоступними для прикладних програмістів, які працюють з класом. З іншого боку, творець класу може змінювати ці об'єкти, не порушуючи при цьому роботи існуючого клієнтського коду. Крім того, заміна вкладених об'єктів на стадії виконання програми дозволяє динамічно змінювати її поведінку. Механізм успадкування такою гнучкістю не володіє, оскільки для похідних класів встановлюються обмеження, що перевіряються на стадії компіляції.

На базі агрегування реалізується методика делегування, коли поставлено перед зовнішнім об'єктом завдання, яке передається внутрішньому об'єкту, що спеціалізується на вирішенні завдань такого роду.

Агрегація (агрегування за посиланням) – відношення «частина-ціле» між двома рівноправними об'єктами, коли один об'єкт (контейнер) має посилання на інший об'єкт. Обидва об'єкти можуть існувати незалежно: якщо контейнер буде знищений, то його вміст – ні.

Android система реалізує принципи найменших привілеїв. Тобто, кожен додаток, за замовчуванням, має доступ тільки до компонентів, які він вимагає, але не більше того. Це створює дуже безпечне середовище, в якій додаток не може отримати доступ до частини системи, до якої не дозволено торкатись.

Тим не менш, є шляхи для обміну даними з іншими додатками і для доступу до системних послуг:

— Можна розділити Linux Id між двома додатками, щоб вони були в змозі отримати доступ до файлів один одного. Щоб заощадити системні ресурси, додатки з одним і тим же ідентифікатором можуть запускатись в одному процесі й ділити між собою одну й ту ж віртуальну машину (англ. Virtual Machine, VM).

— Додаток може запитати дозволу на доступ до персональних даних пристрою, таких як контакти користувача, SMS повідомлень, розділ монтування (SD карту), камеру, Bluetooth і багато іншого. Всі дозволи програмі мають бути надані користувачем під час встановлення. При розробці всі привілеї, які використовує додаток декларуються в маніфест файлі. Маніфест файл – це файл в який інкапсулюється вся архітектура додатку. Він має розширення XML. Наприклад, щоб надати додатку дозвіл на використання Internet, потрібно в маніфесті написати наступні рядки коду:

```
<uses-permission android:name="android.permission.INTERNET" />
```

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Бібліотека Volley це HTTP бібліотека, яка робить мережевий обмін Android додатків більш легким і, найголовніше, більш швидким. Volley доступний для використання через відкрите AOSP сховище.

Volley надає наступні переваги:

- автоматичне планування мережевих запитів;
- кілька паралельних мережевих підключень;
- прозоре кешування на диску і в пам'яті відповідей, використовуючи стандартну

HTTP когерентність кеш;

- підтримка пріоритетів запитів;
- API для скасування запиту;
- простота настройки, наприклад, для повторних спроб і відстрочки;
- строгий порядок, який дозволяє легко і правильно заповнити призначений для користувача інтерфейс даними отриманих з мережі асинхронно;
- інструменти для налагодження і трасування.

Volley виділяється серед RPC операцій, використовуваних для заповнення призначеного для користувача інтерфейсу, таких як отримання сторінки результатів пошуку у вигляді структурованих даних. Вона легко інтегрується з будь-яким протоколами і поставляється з підтримкою сирих рядків, зображень і JSON. Надаючи вбудовану підтримку функції, які вам необхідні, Volley звільняє від написання стандартного коду і дозволяє зосередитися на логіці, яка специфічна для застосування.

Volley не підходить для скачування великих даних або потокових операцій, так як Volley зберігає всі відповіді в пам'яті під час аналізу і розбору даних.

Ядро бібліотеки Volley розробляється у відкритому AOSP сховище в frameworks / volley і містить основу диспетчеризації запитів, а також набір найбільш часто вживаних утиліт, доступних в папці "toolbox". Найпростіший спосіб додати Volley в проект це клонувати репозиторій Volley, і встановити її в якості бібліотеки проекту:

Git клонує репозиторій, використовуючи наступну команду в командному рядку:

```
git clone https://android.googlesource.com/platform/frameworks/volley
```

Після чого імпортують завантажені файли з вихідним кодом в додаток проекту в якості бібліотеки Android проекту.

Існує два основних класи:

— Request queue: використовується для відправки мережових запитів, можете створювати клас request queue, де хочете, але, як правило його створюють під час запуску і використовують як Singleton;

— Request: він містить всі необхідні деталі для створення виклику Web API. Наприклад: який метод використовувати (GET або POST), дані запиту, response listener, error listener.

Стандарт JSON є частиною стандарту ECMAScript починаючи з 1999 року, коли ECMA визначила функцію eval () сприймає формат. Він став набирати популярність разом з наростаючим успіхом Ajax. Слово JSON часто з'являється коли хто-небудь починає розмову про Ajax. Відомо, що JSON є іншим форматом даних, що він заміщає XML, і що багато програмісти активно підтримують його.

Перевага JSON полягає в тому, що він розпізнається в JavaScript. Немає необхідності розбирати документ, як це відбувається з XML, для передачі даних через інтернет.

Переваги JSON:

–Простий для читання і розуміння.

–Простий для використання.

–Має можливості розширення

І XML і JSON мають можливість інтегрувати великий обсяг даних в бінарній формі. Файл JSON дозволяє завантажувати дані з сервера або на сервер. Наприклад, збереження вмісту форми, яка була тільки що заповнена користувачем. Процес включає три фази: обробку браузером, обробку сервером, і обмін даними між ними.

Файл JSON використовується різними програмними мовами, включаючи PHP і Java, завдяки наявності парсерів, які дозволяють отримувати вміст і навіть конвертувати його в класи і атрибути мови. На сайті JSON можна знайти парсери для багатьох мов програмування [4].

Зокрема, бібліотека Volley містить в собі JsonObjectRequest, що слугує для відправки запитів на віддалений сервер через URL-адресу. Синтаксис метода GET , що використовувався в нашому додатку приведено нижче:

```
JsonObjectRequest jsonObjectRequest = new JsonObjectRequest (Request.Method.GET,
urlFinanceUajs,null,
    new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject s) {
            try {
            }catch (JSONException e){}
            try {}catch (JSONException ex){
                ex.printStackTrace();
                Toast.makeText(getApplicationContext(),"Error: connection to server failed! Try again later",
Toast.LENGTH_LONG).show();
            }
        }, new Response.ErrorListener() {
    }
}
```



Для розробки мобільних додатків під Android необхідним є досить широкий інструментарій розробника. Він включає в себе необхідні бібліотеки, від Java, Android SDK, середовище програмування, а також набір емуляторів, програм налагоджувачів, різноманітних екранів стану операційної системи та систем моніторингу та логування.

Можна виділити необхідні інструменти, без яких розробка мобільних додатків під Android просто неможлива. З іншого боку, існує велика кількість допоміжних систем, що в деякій мірі спрощують процес розробки.

До обов'язкових інструментів відноситься *Android SDK* - набір засобів програмування, який містить інструменти, необхідні для створення, компіляції і збірки мобільного додатку [9].

Розглянемо коротко найбільш важливі інструменти, що входять до складу Android SDK:

*SDK Manager* - інструмент, що дозволяє завантажити компоненти Android SDK. Він відображає пакети Android SDK і їх статус: встановлений (Installed), не встановлено (Not Installed), доступні оновлення (Update available).

*Debug Monitor* - самостійний інструмент, що надає графічний інтерфейс для декількох інструментів, призначених для аналізу та налагодження Android додатків:

- *DDMS (Dalvik Debug Monitor Server)* - надає послуги перекидання портів, захоплення екрану пристрою, інформацію про потоки і динамічну пам'ять пристрою, виведення інформації про дії Android в реальному часі (logcat) і багато іншого.

- *Hierarchy Viewer* дозволяє налагоджувати і оптимізувати користувацький інтерфейс Android програми.

- *Tracer for OpenGL ES* - інструмент для аналізу OpenGL | ES коду, що використовується в мобільному додатку дозволяє захоплювати команди OpenGL | ES і демонструвати їх окремими кадрами, що допомагає зрозуміти як виконуються графічні команди.

- *Android Emulator (emulator)* - віртуальний мобільний пристрій, який створюється і працює на комп'ютері розробника. Він використовується для розробки і тестування мобільних додатків без залучення реальних пристроїв.

- *AVD Manager* - надає графічний інтерфейс для створення віртуальних Android пристроїв (AVDs), передбачених Android Emulator, і управління ними.

*Android Debug Bridge (adb)* - гнучкий інструмент, що дозволяє керувати станом емулятора або реального Android пристрою, підключеного до комп'ютера. Також може використовуватися для установки Android програми (.apk файл) на реальний пристрій [5].

**Висновки й перспективи подальших досліджень.** В результаті дослідження була створена програма для Android, яка взаємодіє з віддаленим сервером, та обмінюється з ним інформацією у *GET*-запитах. Запити відбуваються у єдиному потоці. У разі виникнення помилок програма повідомляє про це користувача за допомогою впливаючих повідомлень. Програма протестована на планшетному комп'ютері Ainol Novo 9 Spark (Android 4.3), смартфоні Sony Ericsson Xperia c2305 (Android 5.1) та ще одному смартфоні Lenovo p780 (Android 4.2). На всіх пристроях програма коректно встановлювалась та працювала. За рахунок використаних алгоритмів збереження мережного трафіку, програма швидко отримувала потрібну інформацію від серверу навіть при повільній швидкості підключення.

В нашій статті ми розглянули операційну систему Android, розглянули процес моделювання роботи Android-додатку перегляду курсів валют, організували збір статистичних даних з банків, розглянули механізм передачі даних через мережу Інтернет, розглянули і дослідили роботу стандарту передачі даних JSON.

У програмі може бути покращений інтерфейс користувача - можна додати більше функції у меню. Також можна максимально ефективно використовувати екран якщо використовується пристрій з великим екраном. Для цього потрібно додати декілька макетів та задати їм відповідну поведінку в залежності від розміру екрану. Також замість стандартних макетів можна використовувати фрагментацію інтерфейсу. Ця технологія дозволяє більш гнучко керувати розміщенням елементів на екрані користувача.

Для збільшення швидкості роботи у програму можна додати кешування. Кешування у Android відбувається за допомогою запису потрібної інформації до бази даних *SQLite*. *SQLite* - база даних, яка для зберігання інформації використовує текстові файли.

Вони будуть кожний раз завантажуватися з серверу у *JSON* - масиві, після чого користувач зможе їх переглянути.

1. Дейтел П., Дейтел Х., Дейтел Э., Моргано М. Android для програмістів. Створюємо програми на Android. «Питер». – 2012.
2. Комагинени С., Маклин Д., Хашими С. Розробка програм для Android. «Питер». – 2011.
3. Медник З., Дорнин Л. Программирование под Android. Издательство Питер, 2012.
4. Роджерс Р., Ломбардо Д. Android. Разработка приложений. — М.: ЭКОМ Паблшерз,
5. Фелкер Д. Android: разработка приложений для чайников. - М.: Диалектика, 2012.
6. Х. М. Дейтел, П. Дж. Дейтел, С. И. Сантри Програмування на Java - ДияСофтЮП, 2010.

УДК 621.58:681.32 (004)

Каганюк О.К., Поліщук М.М., Гаджемурра А.М.  
Луцький національний технічний університет

## РОЗРОБКА РАЦІОНАЛЬНОЇ МОДЕЛІ ДЖЕРЕЛА ЖИВЛЕННЯ ДЛЯ ПЛІС

**Каганюк О. К., Поліщук М.М., Гаджемурра А. М. Розробка раціональної моделі джерела живлення для ПЛІС.** У даній статті розглядається аналіз розробки пристроїв на базі ПЛІС і дається обґрунтування, а також пропонується раціональна модель будови структурної схеми живлення і особливості проектування джерел електроживлення ПЛІС. Показані способи реалізації блоків випрямлення, стабілізації напруги і забезпечення необхідної черговості подачі напруги на ПЛІС.

**Ключові слова:** джерело живлення, випрямляч напруги, стабілізатор напруги, ПЛІС, напруги живлення ПЛІС.

**Каганюк А. К., Полищук Н.Н., Гаджемурра А. М. Разработка рациональной модели источника питания для ПЛИС** В Данной статье рассматривается анализ разработок на базе ПЛИС и даются обоснование, а так же предлагается рациональная модель построения структурной схемы питания и особенности проектирования питания для ПЛИС. Показаны способы реализации блоков выпрямления, стабилизации напряжения и обеспечения необходимой очередности подачи напряжения на ПЛИС.

**Ключевые слова:** источник питания, выпрямитель напряжения, стабилизатор напряжения, ПЛИС, напряжение питания ПЛИС.

**Kaganyuk A, Polishchuk M, Gadzhemura A. M. Development of a rational model power supply for FPGA** This article discusses the analysis of developments on the PLYS and provides the rationale, as well as offered a rational model of the block diagram the power and features of the design power for PLYS. Showing how to implement rectification units, voltage regulation, and provide the necessary prioritization supply voltage at the . PLYS

**Keywords:** power supply, voltage rectifier, voltage regulator, PLYS supply voltage.

Запровадження електроніки й автоматики в усі сфери людської діяльності, що спостерігається в даний час є **актуальним** і пред'являє все більш жорсткі вимоги до виробів електронної техніки. Це пов'язано, з одного боку, зі зростанням важливості і складності розв'язуваних задач, а, з іншого боку, необхідністю поліпшення таких характеристик, як швидкість, надійність, споживана потужність, габарити, вартість та інше. Для вирішення певних задач почали широко використовувати ПЛІС. Доказом перспективності нової елементної бази служить щорічна поява нових поколінь ПЛІС, а також постійно зростаючий обсяг випуску вже розроблених ПЛІС. Результати досліджень, що опубліковані в сучасній літературі, свідчать про те, що істотно (на 2-3 порядки) підвищується швидкість моделювання без втрати точності можна досягти реалізацією потактового симулятора на програмованих логічних інтегральних схемах (ПЛІС) або перетворювача двійкових кодів.

Розробка пристроїв на основі ПЛІС являє собою нову технологію проектування електронних схем, включаючи їх виготовлення і супроводження. Істотною перешкодою широкого практичного використання ПЛІС є відсутність ефективних методів синтезу. Справа в тому, що в основу архітектури сучасних ПЛІС покладена структура програмувальних логічних матриць (ПЛМ), що являє собою пари матриць: І й АБО, у якій програмується матриця І, а матриця АБО має фіксоване положення. Методи й алгоритми синтезу на такій структурі одержали назву двохрівневого синтезу і бурхливо розвивалися в 80-х роках. З часом інтерес дослідників на багато років залучив багаторівневий синтез, використовуваний при проектуванні цифрових систем на основі FPGA (Field Programmable Gate Array). В останні роки спостерігається явне протиріччя: архітектури ПЛІС бурхливо розвиваються й удосконалюються, а методи проектування на їх основі залишаються без зміни.

Відносно низьку популярність даного інтегрованого середовища серед вітчизняних проектувальників можливо пояснити відсутністю детальної документації українською мовою про можливості та властивості ПЛІС. Також в Україні немає розвиненої галузі виробництва з комп'ютерної електроніки, відповідно, якщо українці і досліджують ПЛІС, то лише для закордонних виробників

В зв'язку ускладненням цифрових мікросхем і на базі їх створення багатofункційних блоків, які потребують джерела живлення різної конфігурації, виникає **проблема** створення та уніфікація джерела живлення, здатного забезпечити необхідною лінійкою напруги.

Багато складних цифрових мікросхем використовують по декілька джерел живлення електричної напруги, які повинні подаватися на окремий блок, модуль або мікросхему в строго визначеній послідовності і з необхідною швидкістю наростання цієї напруги. В зв'язку з тим, стає необхідним розробити функціонує схему джерела живлення програмованої логічної схеми (далі ПЛІС), яка б давала можливість формувати вхідну напругу для вторинного джерела електроживлення і секвенсора. Напруга від первинного джерела надходить на кілька вторинних розподільчих навитків, що дозволяє підвищити якість постачання живлення, за допомогою яких формуються всі необхідні напруги електроживлення. Секвенсор формує необхідну послідовність подачі напруги і/або швидкість їх наростання. Слід особливо зазначити, що одним з ключових компонентів інтерфейсу електроживлення ПЛІС є ланцюжок шунтуючих конденсаторів навколо ПЛІС. Вони дозволяють розподілити робочий струм між споживачами, використовуючи низькоімпедансний шлях проходження струму, тим самим знижуючи рівень високочастотного шуму, що дозволяє підвищити працездатність даної схеми.

Перетворювач двійкових кодів буде базуватися на логічних елементах І та АБО. Принцип схожий на мікропроцесорний. Побудова ефективних методів розробки і тестування апаратних потактових моделей мікропроцесорів на ПЛІС, що дозволяють домогтися без втрати точності істотного підвищення швидкості моделювання при низькій складності розробки. Для досягнення мети дослідження вирішуються наступні завдання:

- Дослідження існуючих способів підвищення швидкості потактових моделей мікропроцесорів і їх класифікація;
- Дослідження особливостей застосування ПЛІС для потактового моделювання мікропроцесорів, що дозволяють знизити трудомісткість в порівнянні з розробкою прототипу;
- Розробка ефективного методу розробки апаратних потактових симуляторів мікропроцесорів на ПЛІС;
- Розробка методу і системи автоматизованого тестування апаратних потактових симуляторів мікропроцесорів на ПЛІС;

Суть роботи ПЛІС-орієнтованого функціонального перетворювача двійкового коду у тому, що він містить вхідні та вихідні лінії, логічні елементи "І" та "АБО", лінії зв'язку. Пристрій виконаний у вигляді комбінаційної схеми з подвійною матричною структурою, яка складається з  $n$  рядків та  $t$  стовпчиків  $b$ -входових логічних елементів "І", що утворюють вузли матриці 1 та з  $n$  рядків та  $t$  стовпчиків  $b$ -входових логічних елементів "АБО", що утворюють вузли матриці 2.

Корисна модель належить до галузі обчислювальної техніки і являє собою комбінаційну схему, яка може бути використана як функціональний модуль, що входить до складу спеціалізованої системи [1] та призначена для автоматичного визначення кількісного значення рівня збігів у двох символічних послідовностях із довжиною не менше за  $d(2 \leq d)$  символів (деякий параметр).

Вхідною послідовністю для цього перетворювача є двійковий вектор  $S$  довжиною  $n$ , що являю собою результат порівняння символів в іншому блоці спеціалізованої системи. Дана корисна модель забезпечує виявлення всіх збігів шляхом заміни у вхідній двійковій послідовності на "0" груп поруч розташованих "1" в кількості менше ніж  $d(2 \leq d)$ .



Дослідження в даному напрямку потребують удосконалення, а ще краще, створення ефективних і на наш погляд, раціональних джерел живлення для ПЛІС

**Виклад основного матеріалу.** На ПЛІС можна реалізувати необмежену кількість всіляких конфігурацій схем, що працюють на різних тактових частотах і, отже, споживають різну потужність. А так як початковими даними для проектування системи електроживлення ПЛІС є споживана потужність, то в результаті треба мати максимально чітке уявлення про споживану систему та умови її функціонування. Для цього необхідно визначити:

- тактову частоту ПЛІС (споживана потужність пропорційна частоті);
- кількість задіяних ресурсів ПЛІС;
- число необхідних для живлення електричних напруг та окремих джерел електроживлення;
- діапазон робочих температур.

Для електроживлення різних функціональних блоків ПЛІС використовуються різні рівні напруги. Для забезпечення одного рівня напруги необхідний одне вторинне джерело електроживлення.

Всі ПЛІС потребують джерело напруги, яку подається на ядро, але більшості складних ПЛІС необхідне окреме джерело напруги електроживлення для блоків введення-виведення, джерело опорних напруг, джерело напруги для термінальних резисторів і джерело напруги для додаткових функцій.

Розглянемо більш докладно типи напруг електроживлення, необхідні для ПЛІС Xilinx:

- $V_{CCINT}$  - зовнішня напруга електроживлення ядра – є основною напругою, яка живить ПЛІС і, як правило, забезпечує більшу частину потужності, що витрачається в ПЛІС. Основна напруга джерела залежить від сімейства ПЛІС. ряд основних напруг електроживлення включає: 1,2; 1,5; 1,8; 2,5; 3,3 і 5,0 В. Напруга електроживлення ядра може використовуватися і для роботи конфігурацій ROM.

- $V_{CCO}$  - напруга електроживлення блоків введення-виведення. ПЛІС від Xilinx містять блоки введення-виведення, які можуть бути сконфігуровані для роботи за різними стандартами. Залежно від того, який стандарт вибраний розробником, використовуються різні рівні напруг (1,5; 1,8; 2,5; 3,0 і 3,3 В). Конфігурація блоків введення-виведення можлива в одній і тій ж ПЛІС, але має на увазі наявність декількох джерел напруги електроживлення пристрою введення-виведення ( $V_{CCO}$ ).

- $V_{CCAUX}$  - допоміжна напруга електроживлення, що необхідна для різних допоміжних функцій ПЛІС, наприклад для пристрою управління тактовою частотою або інтерфейса JTAG. Допоміжні пристрої більш чутливі до шуму, ніж інші, і тому  $V_{CCAUX}$  має більш високі вимоги

щодо розв'язки для мінімізації наведеного шуму. Напруги VCCAUX можуть бути 2,5 і 3,3 В. У разі якщо джерело електроживлення ядра напругою VCCO володіє низьким шумом, від нього можна живити блоки допоміжних функцій.

- $V_{\text{CCAUXTX}}$  - допоміжна напруга електроживлення для RocketIO-трансиверів, якими володіє родина ПЛІС Virtex-II Pro. Ця напруга живлення має особливі вимогами до шуму. Пристрій повинен бути живитися окремим стабілізатором напруги, підключеним до окремої заземленої поверхні на друкованій платі (GNDA).

- $V_{\text{REF}}$  - опорна напруга електроживлення, що необхідна для диференціального порту, який використовує різні стандарти введення-виведення, такий як LVDS.

- $V_{\text{TT}}$  - напруга електроживлення термінальних резисторів.

Залежно від сімейства ПЛІС може знадобитися визначена послідовність подачі необхідної напруги. Для сімейства Spartan-III потрібна одночасна подача напруг VCCINT і VCCO. Якщо напругу ядра VCCINT подати раніше, ніж напруги електроживлення блоків введення-виведення, то струм може різко вирости через те, що лінії введення-виведення не перебували в високоімпедансному стані. Проте, якщо струм ICSP0 забезпечується джерелом електроживлення VCCINT, процес включення буде успішним незалежно від послідовності подачі напруг живлення. Для сімейства Virtex-II Pro черговість ввімкнення джерел не важлива, за винятком джерела VCCAUX, який повинен вмикатися раніше VCCO, щоб забезпечити мінімальний струм від джерела електроживлення. [2]

Для електроживлення розроблюваної ПЛІС потрібні дві напруги живлення: 2 В для електроживлення блоків введення-виведення і 3,3 В для електроживлення ядра. Необхідно забезпечити якомога менший інтервал ( $0 \div 30$  мс) між включенням вихідних сигналів напруг джерела електроживлення і їх певну черговість: на початку потрібно подати напругу 3,3 В, а потім 2 В. Напруги повинні подаватися в такій послідовності або одночасно.

Для кожної напруги, що подається на ПЛІС, використовується своє власне джерело електроживлення (рис. 1). Кожне з них складається з трансформатора (Тр1 і Тр2), що живиться від трьохфазної мережі напругою 220 В і частотою 400 Гц, випрямлячів напруги ( $V_{1\text{осн}}$ ,  $V_{1\text{дод}}$  і  $V_{2\text{осн}}$ ,  $V_{2\text{дод}}$ ), і стабілізатора напруги ( $CH2$  і  $CH3,3$ ).

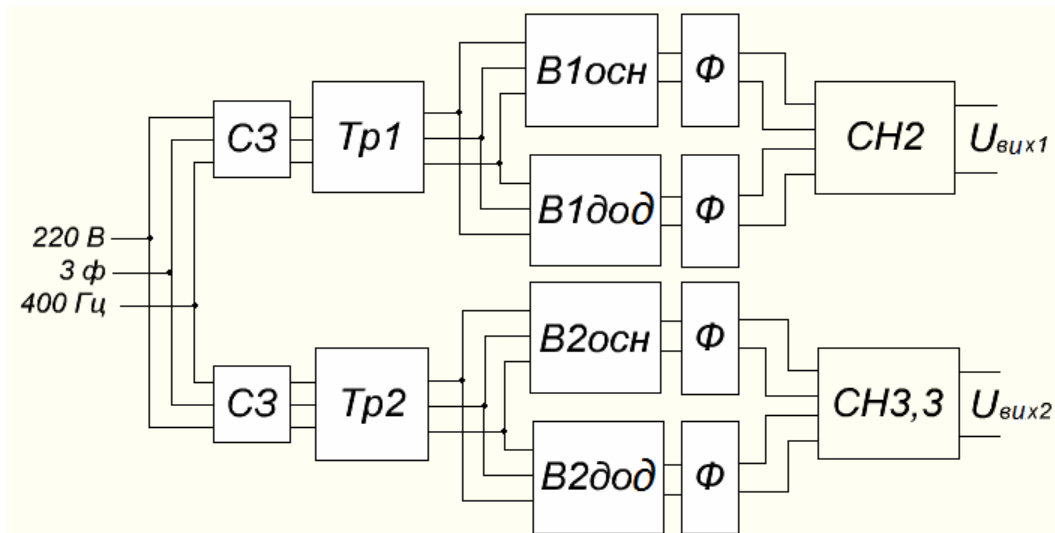


Рисунок 1 – Структурна схема джерела електроживлення

Випрямлячі напруги  $V_{1\text{осн}}$ ,  $V_{1\text{дод}}$  і  $V_{2\text{осн}}$ ,  $V_{2\text{дод}}$  забезпечують на виході два необхідних випрямлених напруги постійного струму: основний і додатковий. На вході джерел вторинного електроживлення встановлено вузли захисту від перенапруження на основі плавкого запобіжника (С3).  $CH2$  і  $CH3,3$  мають вихідні параметри 2 В 4 А і 3,3 В 6 А. Виконання стабілізаторів відрізняються, в основному, кількістю транзисторів регулюючого компонента, тому розглянемо тільки один з них –  $CH3,3$ .

Даний компенсацийний стабілізатор напруги безперервної дії окрім основного стабілізатора напруги містить також додатковий параметричний стабілізатор напруги, схему захисту від перенапруги на виході, від перевантажень по струму і коротких замикань у навантаженні. Його

структурна схема приведена на рис. 2. До складу стабілізатора напруги входять наступні функціональні вузли:

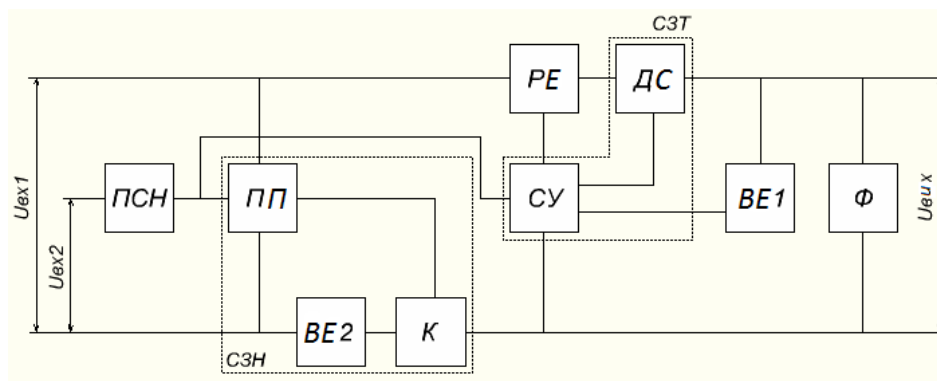


Рисунок 2 – Структурна схема стабілізатора напруги

- основний вузол компенсаційної стабілізації,
- додатковий вузол параметричної стабілізації (ПСН),
- пристрій захисту від перевантажень по струму і коротких замикань в навантаженні,
- пристрій захисту від перевищення напруги в вихідних ланцюгах.

Основний вузол компенсаційної стабілізації напруги складається з регулюючого елемента (РЕ), вимірювального елемента та (ВЕ1), схеми управління (СУ), що включає в себе му порівняння і підсилювач постійного струму.

Схема захисту від перенапруги на виході (СЗН) включає в себе вимірювальний елемент (ВЕ2), граничний пристрій (ПП) і ключ (К).

Схема захисту від перевантажень по струму і коротких замикань в навантаженні (СЗТ) складається з датчика струму (ДС) і частини схеми управління (СУ).

Один із способів забезпечення необхідної черговості включення напруг – підбір відповідної ємності конденсаторів в фільтрах випрямлячів і стабілізаторах напруги. Інший метод – використання електричного ключа та пристрою управління. При використанні даного методу і правильному підборі елементів схеми затримка між появами напруги на ПЛІС буде визначатися часом включення ключа.

**Висновки.** На основі проведення досліджень, які передбачались науково-дослідною роботою, можна зробити наступні висновки:

1. Проведений аналіз розробок на базі ПЛІС потребує створення раціональної моделі джерела живлення дл ПЛІС;
2. Розроблене джерело електроживлення може застосовуватися у ПЛІС, для забезпечення роботи блоку живлення з різними форматами напруги для необхідного ядра
3. Алгоритми розробки джерела електроживлення можуть застосовуватися при вирішенні таких завдань, як розробка блоків живлення, так і при вирішенні завдань забезпечення селективності каналів електроживлення

1. Антонов А.П. Язык описания цифровых устройств AlteraHDL. М.: РадиоСофт, 2001. – 221 ст.
2. Баранов С.И. Синтез микропрограммных автоматов. Л.: Энергия, 1979. 232 ст.
3. Девятков В.В. Методы реализации конечных автоматов на сдвиговых регистрах. М.: Энергия, 1974. – 80 ст.
4. Денисенко Е.Л. Иерархический синтез асинхронных автоматов на программируемых логических интегральных схемах (ПЛИС) с учетом ограничений. М.: УСИМ, 1997. – 476 ст.
5. Закревский А.Д. Алгоритмы синтеза конечных автоматов. М.: Наука, 1971. – 502 ст.
6. Алимов С. Применение стабилизаторов Micrel для питания ПЛИС и DSP//Мир электронных компонентов, 2008, №3 – С. 37 – 42.
7. Шаропин Ю., Будаев В. Основы построения систем электропитания ПЛИС//Компоненты и технологии, 2006, №8 – С. 144-151.
8. Гусев В. Г., Гусев Ю. М. Электроника, 2 издание – М.: Высшая школа, 1991. – С. 617.
9. Найвельт Г. С., Мазель К. Б., Хусаинов Ч. И. и др, под редакцией Найвельта Г. С. Источники электропитания радиоэлектронной аппаратуры: справочник – М.: Радио и связь, 1985. – С. 576.



УДК 629.113(071):004.01:004.04

Каганюк О.К., Панчук Р.Я.

Луцький національний технічний університет

## ДОСЛІДЖЕННЯ ШВИДКОДІЇ РОБОТИ З БАЗАМИ ДАНИХ РІЗНИХ ТИПІВ ПРОГРАМ, ТА ВИБІР РАЦІОНАЛЬНИХ ПАРАМЕТРІВ АПАРАТНИХ РЕСУРСІВ ПРИ ОДНАКОВОМУ ФУНКЦІОНАЛІ

**Каганюк О.К., Панчук Р.Я.** Дослідження швидкодії роботи з базами даних різних типів програм, та вибір раціональних параметрів апаратних ресурсів при однаковому функціоналі. В даній статті розглядається питання ефективності використання баз даних на ОС Android з різною структурою роботи з даними та пам'яттю. Дослідження проводиться методом аналізу часових характеристик запису даних у БД.

**Ключові слова:** Android, Бази даних, Android Studio, SQLite, sqlite3

**Каганюк О.К., Панчук Р.Я.** Исследование быстродействия работы с базами данных различных типов программ, и выбор рациональных параметров аппаратных ресурсов при одинаковом функционале. В данной статье рассматривается вопрос эффективности использования баз данных на ОС Android с различной структурой работы с данными и памятью. Исследование проводится методом анализа временных характеристик записи данных в БД.

**Ключевые слова:** Android, Базы данных, Android Studio, SQLite, sqlite3

**Kaganyuk OK, Panchuk RJ.** "Study of performance working with databases of different types of programs and selection of rational parameters of hardware resources with the same functionality. This article examines the efficiency of databases on Android OS with a different structure of data and memory. Research conducted by analysis of time characteristics data records in the database.

**Keywords:** Android, Databases, Android Studio, SQLite, sqlite3

Однією з основних платформ, яка використовується в мобільних телефонах, є система Андроїд. Дана система характерна відкритим кодом ядра, та своїм розповсюдженням на безкоштовній основі та можливістю реалізації на різноманітних апаратних платформах для різних пристроїв. Дана операційна система охоплює дуже широкий сектор ринкових послуг, що дає можливість затвердити **актуальність** даного напрямку досліджень.

На даний момент програмне забезпечення для різноманітних пристроїв виросло у дуже різноманітні програмні комплекси. Перед початком розроблення будь якого додатку потрібно спочатку провести аналіз доцільності в виборі структуру майбутньої програми. Постає питання, яким чином необхідно провести аналіз структур програмного забезпечення? І стає очевидним, що необхідно проводити аналіз по характеристиці їхньої швидкодії та раціонального використання апаратних ресурсів при однаковому функціоналу. Саме для цього і проводиться аналіз швидкодії роботи з базами даних двома різними методами.

Головне питання усіх сучасних програм це їхня оптимізація при запуску та прискорення їхньої роботи. Практично усі додатки на сьогодні містять у собі десятки тисяч строк програмного коду і тут постає **проблема** в оптимізації даного коду та структури програми для пришвидшення її роботи та покращення читабельності самого програмного коду для розробників. Це дозволяє зменшити час компіляції та оптимізувати роботу готового кінцевого продукту. Тому, питанням даної статті, полягає в аналізі та продуктивності читання – запису даних у додатках з різними методами взаємодії з базами даних.

Було поставлено завдання по розробці двох програмних додатки які б працювали з базами даних студентів.

При роботі з невеликими програмними додатками приріст швидкодії при переході з лінійної структури на модульну є невеликим і не являється критично важливим. Недоліком лінійної програми є пов'язаність з обмеженими ресурсами, які пов'язані з формуванням наступних питань:

- необхідність в формуванні точної цілі;
- задача повинна бути однорідною та поділюваною;
- результати та використовувані ресурси повинні бути розподілені на окремі долі.

Головна **ціль** даної статті полягає в математично обґрунтованому доведення переваги модульного програмування над лінійним. Використовуючи засоби середовища Eclipse нами створюються дві аналогічні програми для платформи Android. Дані програми призначені для передачі



та обробки даних, роботи з базою даних використовуючи можливості SQL Lite. Також проводиться тестування швидкості кодування інформації програмами з використанням апаратних ресурсів системи.

Перевага модульної програми полягає у тому що модульне програмування, або ще по іншому структурне програмування, це організація програми у вигляді взаємопов'язаних структурних блоків що пов'язуються між собою визначеними правилами взаємодії.

Модульне програмування дозволяє визначити наступні переваги у порівнянні з лінійним, адже саме для модульного програмування найкраще підходить вислів «розділяй та володій»:

- спрощується процес тестування програмного додатка та його налагодження;
- виникає можливість розподілити створення готової програми між кількома виконавцями по блоках;
- скоротити час розробки та знизити загальну вартість робіт.

Можна строго відокремити апаратно-залежні задачі від інших під задач, що дозволить поліпшити мобільність створюваних програм.

Програму можна вважати модульною якщо вона відповідає наступним принципам:

- модулі є незалежними програмними блоками, код яких логічно і фізично відокремлений від коду інших модулів;
- модулі є неподільними блоками програми, які можна використовувати повторно;
- розмір модуля обмежується розміром сегмента коду, який виділяється під час компіляції модуля.

При переході на модульну програму з лінійної, виникає необхідність в використанні метода структурування. Даний метод призначений для перетворення неструктурованої програми на еквівалентну їй структуровану, тобто таку, що складається з обмеженого набору керуючих алгоритмічних структур. На наведеному нижче рисунку показано блок схему перетворення в структуровану програму методом дублювання.

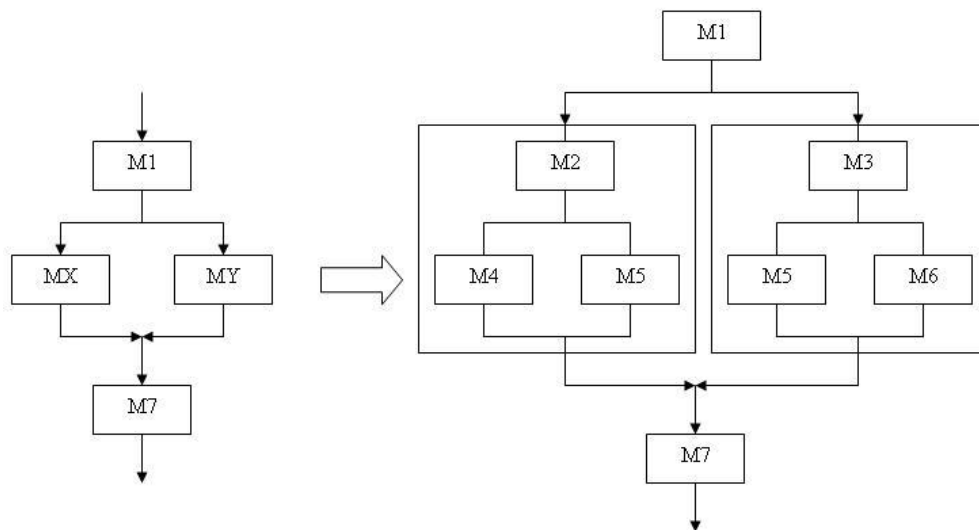


Рис.1 – перетворення неструктурованої програми в структуровану

Ще цікавим є метод булевої ознаки для перетворення неструктурованих програм, що містять цикли. В програму вводиться додаткова змінна, яка набуває значення true чи false.

**Застосування:** Дані методи покликані оптимізувати роботу з програмою для покращення її швидкодії та взаємодії з даними. Ще однією дуже важливою характеристикою модульних програм є можливість виконання модулів програми паралельно використовуючи максимально апаратні можливості пристрою.

В основному виділяють такі критерії оптимізації коду як:

- мінімізація часу виконання програми;
- мінімізація розміру коду;

– мінімізація енергоспоживання.

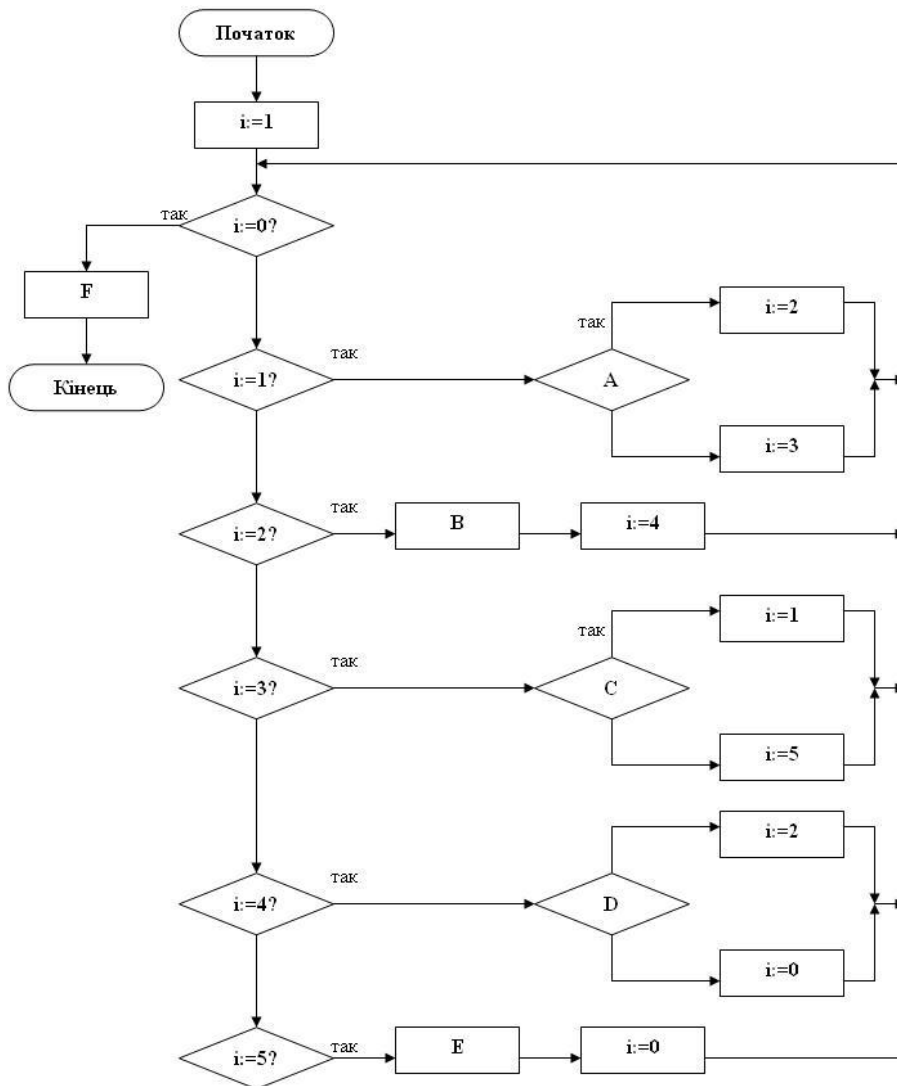


рис.2. – Швидкодія модульного додатку

Останній критерій особливо важливий якщо наш додаток розробляється для мобільної платформи, адже саме в них і потрібна висока оптимізація по енергоефективності для збереження енергії пристрою.

Враховуючи те що більшість сучасних мобільних платформ є багатопроесорними та багато потоковими, дуже важливим чинником що впливає на продуктивність є можливість використання додатків усіх можливостей апаратної платформи. Одним з методів що дозволяють це зробити є метод глобального планування.

Перевагою даного методу глобального планування є те, що він застосовується до великих фрагментів програми, що складається з декількох лінійних ділянок. В результаті з'являється можливість більш ефективного використання апаратних засобів паралельного виконання. Для того щоб ця перевага реально працювала, необхідно вміти переміщувати команди навколо точок розгалуження. Важливою перевагою моделі захищеного планування є підтримка адекватної діагностики виняткових ситуацій із зазначенням адреси фактичного виникнення, навіть якщо команда-джерело виконувалася в попереджувальному режимі, що наведено на рис.1. – Швидкодія лінійного додатку, та на рис.2. – Швидкодія модульного додатку.

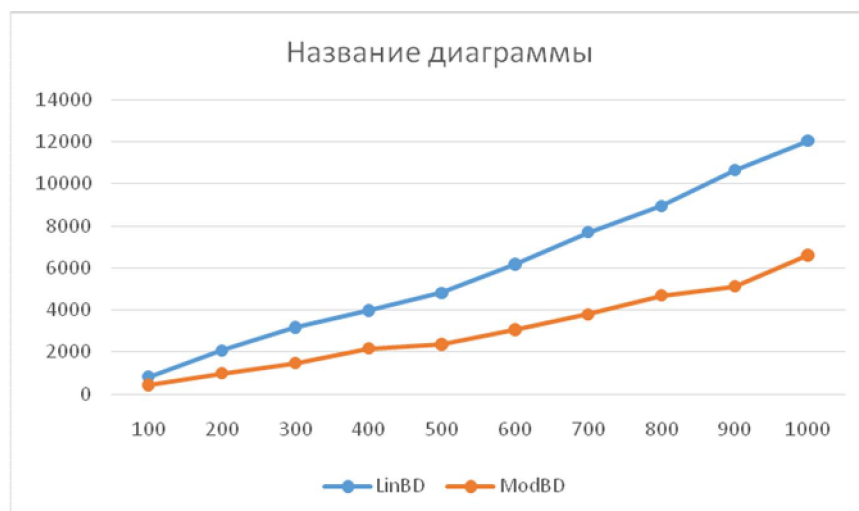


Рис. 1 – лінійний додаток

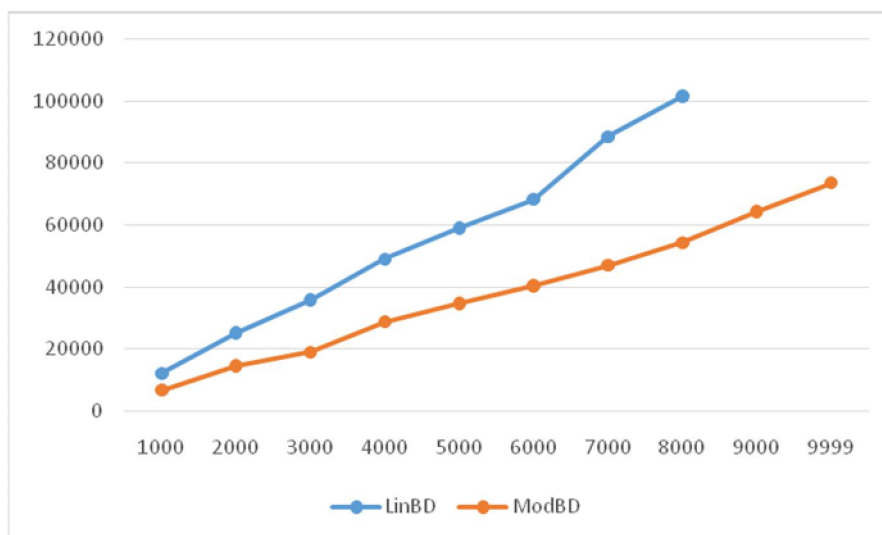


Рис. 2 – модульний додаток

На основі проведених експериментів, які розглядаються в науково – дослідній роботі, можна зробити наступні висновки:

- модульне програмування має ефект лише при розробці великих програмних додатків;
- дана структура надає приріст в швидкодії;
- полегшує читабельність програмного коду;
- надає можливість працювати над одним додатком команді програмістів;
- надає можливість працювати з окремими модулями для пришвидшення розробки, та оптимізації програмного коду.

Використовуючи метод глобального планування ми також отримуємо можливість багато потокової обробки даних, що дозволяє з максимальною ефективністю використовувати апаратні засоби.

1. "Автоматический поиск ошибок синхронизации в приложениях на платформе Android" В.П. Иванников, С.П. Варганов, М.К. Ермаков/
2. «Выработка и анализ требований к защищенной мобильной операционной системе» О.И. Бокова, Д.М. Михайлов, М.И. Фроимсон.
3. «Исследование уязвимости мобильных устройств системах apple и google» МИХАЙЛОВ1 Дмитрий Михайлович, кандидат технических наук.
4. «Разработка приложения для android на java» Елюбаева Ж.Ж. Кандидат технических наук, Евразийский национальный университет имени Л.Н.Гумилева

УДК 004.738:004.9

Коцюба А.Ю., Цяпич Я.П., Лавренчук С.В.

Луцький національний технічний університет

## ПРО МЕТОДИКУ ОПТИМІЗАЦІЇ ВІДМОВСТІЙКОСТІ ВЕБ-СЕРВЕРІВ НА ОДНОЧАСНУ КІЛЬКІСТЬ ЗАПИТІВ

**Коцюба А.Ю., Цяпич Я.П., Лавренчук С.В. Про методику оптимізації відмовстійкості веб-серверів на одночасну кількість запитів.** Проведено дослідження ефективності та відмовстійкості програмної частини веб-сервера. Було проведено навантажувальне тестування веб-серверів синхронної та асинхронної моделі обробки запитів. Окрім тестування та порівняння веб-серверів, було здійснено пошук та налаштування більш оптимізованого конфігураційного рішення для роботи веб-сервера, що дає можливість витримувати більшу кількість запитів.

**Ключові слова:** відмовстійкість, веб-сервер, асинхронна модель опрацювання запитів, навантажувальне тестування, Node.js, Nginx, Apache, Siege.

**Коцюба А.Ю., Цяпич Я.П., Лавренчук С.В. О методике оптимизации отказоустойчивости веб-серверов на одновременное количество запросов.** Проведено исследование эффективности и отказоустойчивости программной части веб-сервера. Было проведено нагрузочное тестирование веб-серверов синхронной и асинхронной модели обработки запросов. Кроме тестирования и сравнения веб-серверов, было осуществлено поиск и настройка более оптимизированного конфигурационного решения для работы веб-сервера, что позволяет выдерживать большее количество запросов.

**Ключевые слова:** отказоустойчивость, веб-сервер, асинхронная модель обработки запросов, нагрузочное тестирование, Node.js, Nginx, Apache, Siege.

**Kotsyuba A.Y., Tsyapych Y.P., Lavrenchuk S.V., On the method of optimizing the web-server failover for simultaneous amount of requests.** The research of the web-server efficiency and resiliency is made in the article. The web-servers testing with loading them have been made in agreement of synchronous and asynchronous query processing model. Furthermore the Web-servers testing and comparison, there have been carried out some search and setting of configuration to get more optimized solutions for the web server that allows to withstand the more number of queries.

**Keywords:** fault tolerance, web-server, an asynchronous request processing model, stress testing, Node.js, Nginx, Apache, Siege.

**Вступ.** На сьогоднішній день ми спостерігаємо значний ріст інтернет-технологій та комунікацій, які все більше і більше масштабуються та нарощуються в своїй масі високими темпами. Зокрема така тенденція відчутна при розробці веб-додатків чи веб-сайтів із великою кількістю користувачів. При проектуванні багатокористувацьких веб-проектів слід враховувати сам підхід і технологію вибрану для їх реалізації, адже від цього залежить, яким чином відбуватиметься функціонування, масштабування та відмовстійкість даного веб-рішення в майбутньому.

**Актуальність проблеми.** Розглянемо основні аспекти та проблематику, що пов'язана з розробкою веб-проектів із великою кількістю одночасних запитів та їх безвідмовною обробкою. Як правило, професійні веб-додатки використовують клієнт-серверну програмно-архітектурну модель. Клієнтом виступає браузер користувача, а сервером – наприклад, одна з машин в дата-центрі деякого хостера. Браузер запитує якийсь ресурс у сервера, а той, у свою чергу, його віддає клієнту. Таким чином, у нас є модель найпростішого веб-сервера – програмна модель, яка приймає запити від браузера, обробляє їх і повертає відповідь. Звучить зрозуміло, але такий найпростіший сервер вміє одночасно спілкуватися лише з одним користувачем. Якщо в момент обробки запиту до нього звернеться ще один клієнт, то користувачу доведеться чекати поки сервер відповідь першому. А в разі одночасного звернення до сервера, наприклад, десятків користувачів, може відбутися збій у його роботі і, як результат, ніхто не отримає відповідь на свій запит. Для того, щоб уникнути цього, необхідно налаштувати обробку запитів між користувачами так, щоб кожен з користувачів безвідмовно одержав відповідь на свій запит. Очевидне рішення: обробляти запити користувачів в окремих потоках або процесах операційної системи. На сьогоднішній день у деякому розумінні даного підходу дотримуються майже всі сучасні веб-сервери. Найбільш відомими серед них є такі, як, наприклад, Apache або IIS.

Але якщо виникає необхідність обробляти одночасно тисячі запитів, а той більше, то для ефективності необхідно повністю змінити ідеологію обробки запитів. І причин для цього є декілька. По-перше, як уже згадувалось вище, у синхронному середовищі запити та їх обробка відбувається послідовно. І навіть перерозподіл цієї послідовності на велику кількість потоків не дасть очікуваної ефективності на таку кількість одночасних запитів. Це буде призводити до неефективного використання серверних ресурсів. І як результат, під час передачі даних по мережі, веб-сервер більшу частину свого часу буде простоювати. По-друге, якщо пробувати вирішувати дану проблему створенням розподілених потоків, то доведеться зустрітися з проблемою неефективного використан-

ня оперативної пам'яті (тобто, якщо створювати окремий потік для кожного з тисячі користувачів, то незабаром можемо опинитися в ситуації, коли на сервері пам'яті просто не залишиться).

Рішенням в даній ситуації – це використовувати асинхронну обробку запитів на сервері. Для того, щоб реалізувати такий підхід, необхідно організувати усе введення-виведення і роботу з базою даних за допомогою неблокуйочної архітектури, побудованої на подійно-орієнтованій парадигмі, так званій “evented I/O”. Одною із не багатьох платформ, які пропонують такі можливості є “Node.js”.

**Аналіз досліджень.** Дослідження проблематики асинхронності обробки запитів на сервері залишаються актуальними на сьогоднішній день. Результатом пошуку компромісу між багатопоточністю та однопоточністю обробки даних на стороні сервера стала розробка web-платформи під назвою “Node.js”, переваги якої наведені у статті [1].

У публікації [2] наведено модель реалізації такого типу сервера в поєднанні з CMF Drupal та короткий опис процесу його розробки, що у свою чергу було підкріплено результатами тестування даної платформи.

У статті [3] наведено приклад порівняння двох серверів, а саме Apache і на базі платформи Node.js. Було проведено дослідження відмовостійкості серверів та здійснена порівняльна характеристика, яка виявила перевагу “Node.js” над “Apache”. Під час спроб взлому (DOS-атаки) web-сервера Apache ситуація виглядала наступним чином: початок атаки сервер не відчуває ніякої завантаженості, але після великої кількості запитів у розмірі 1000000 було виявлено зниження працездатності. Тест показав, що завантаження жорсткого диска дорівнює 97%, але сервер Apache через короткий проміжок часу (7-10 с) знову нормалізовується. Далі відбувався збій пов'язаний з блокуванням (перевантаженням) каналу і маршрутизатор не відповідав, відповідно до сервера не приходили запити і він був не доступний. Стандартний захист маршрутизатора пропускав 100% пакетів протоколу TCP/IP і відповідно канал перевантажувався помилковими запитами, після чого пристрій перестав функціонувати і починав перезапускатися. Як висновок – атака пройшла успішно. В той час як у Node.js ситуація більш оптимістична. Початок атаки проходить доволі успішно, потік пакетів транслюється на сервер, але протягом 5-10 с пакети починають блокуватися маршрутизатором і вже пропускати лише 5-10% всіх вхідних пакетів на сервер. У підсумку сервер обробляє запити на достатньому по надійності рівні. Підсумок – атака пройшла безуспішно.

В дослідженні встановлено, що успішність захисту від DOS і DDOS залежить не тільки від поставленого серверного обладнання а й від серверної програми. Не маючи потужного устаткування, у разі використання сервера на базі платформи Node.js спостерігається висока працездатність навіть при навантаженні вище середнього на відміну від сервера Apache.

Опис способу використання хмарних обчислень з використанням асинхронних web-серверів наведено у праці [4], де було описано алгоритм та методи реалізації асинхронної моделі, що використовують асинхронний ввід/вивід за допомогою зворотніх викликів, в результаті якої було побудовано навчальний чат-сервер.

**Виклад основного матеріалу та обґрунтування отриманих результатів.** Практична частина дослідження відмовостійкості веб-серверів розпочалось із навантажувального тестування. Тестуючим інструментом виступала утиліта під назвою “Siege”. За допомогою цієї утиліти відбувалось генерування запитів до сервера та формувались дані про проведене навантажувальне тестування, серед яких:

- загальна кількість запитів;
- надійність обробки запитів;
- час потрачений обробку усіх запитів;
- кількість даних передана сервером для усіх запитів;
- середній час відклику сервера на запит;
- середнє число запитів, яке сервер зумів обробити за секунду;
- середнє число даних передане від сервера до клієнта за секунду;
- кількість конкурентних запитів, які сервер зумів обробити без затримок;
- кількість вдало оброблених запитів;
- кількість невдало оброблених запитів;
- найдовший час обробки запиту;
- найменший час обробки запиту.

За допомогою отриманої інформації, побудуємо графічні залежності та обґрунтуємо висновки щодо роботи веб-серверів під навантаженням.

Для початку слід зазначити, що тестування відбувалось із використанням локального веб-сервера під управлінням операційної системи Ubuntu 16.04. Практична частина навантажувального тестування складалась із трьох тестів:

1) Навантажувальне тестування веб-серверів, які генеруватимуть і віддаватимуть html-сторінку, на якій буде виводитись лише привітання.

2) Навантажувальне тестування веб-серверів, які обраховуватимуть 20-й член послідовності Фібоначчі та віддаватимуть результат клієнту.

3) Навантажувальне тестування конфігураційного рішення, яке складалось із Nginx та Node.js, які разом реалізовували один веб-сервер з покращеними навантажувальними характеристиками.

Процес тестування відбувався ітеративно. За допомогою тестувальної утиліти, було згенеровано 100 одночасних запитів 100 разів, що у кінцевому результаті дало 100×100 запитів. Протягом усього тесту число одночасних запитів збільшувалось на 50 одиниць. Дана навантажувальна процедура відбувалась до того часу, доки сервер не перестав відповідати на запити.

Першим навантажувальним тестуванням для веб-серверів була html-сторінка, яка виводила привітання, згенероване на серверній стороні. Результат тестування, який полягає у знаходженні усередненого часу обробки запитів у секундах, при 100-кратному повторенні експериментів наведено у таблиці 1 та у вигляді графіка на рис. 1.

Таблиця 1 – Результат тестування на основі сторінки-привітання веб-серверів (Тест 1)

Кількість запитів	Apache, c	Node.js, c
100	6.9	4.94
150	11.37	7.85
200	14.16	11.46
250	18.8	13.4
300	29.72	17.16
350	30.64	21.03
400	32.87	23.96
450	34.09	28
500	-	34.85
550	-	33.35
600	-	35.88
650	-	44.38
700	-	66
750	-	46.25
800	-	46.98
850	-	51.14

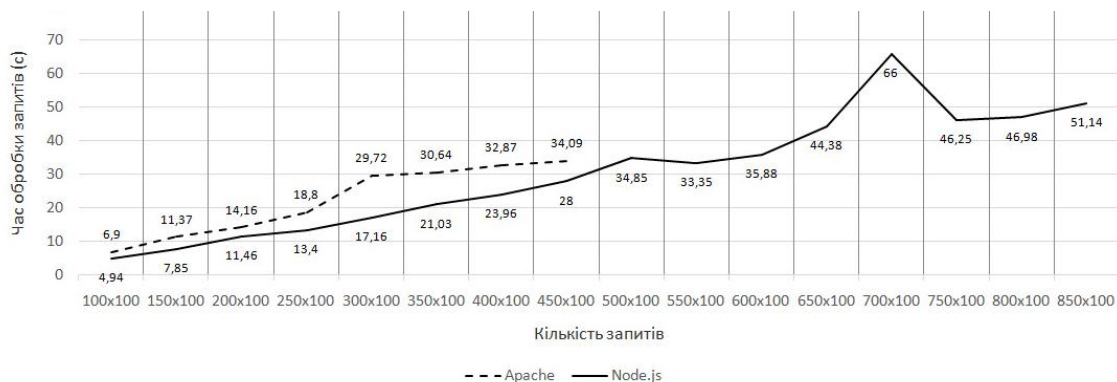


Рис. 1 – Залежність між кількістю запитів та усередненим часом затраченим на їх виконання (Тест 1)

У разі хоча б одного збою вважатимемо, що усередненого часу обробки запитів немає. Таким чином, з таблиці та рисунку видно, що веб-сервер на базі платформи Node.js витримує майже в два рази більше навантаження ніж Apache, а саме 850 одночасних з'єднань. Також даний веб-сервер зумів обробити аналогічну кількість запитів, в середньому, на 6.34 с швидше.

Другим навантажувальним тестуванням для веб-серверів була html-сторінка, згенерована на серверній стороні, яка виводила результат обчислення 20-го члену послідовності Фібоначчі. Результат тестування при аналогічних вищеописаним умовах наведено у таблиці 2 та у вигляді графіка на рис. 2.

Таблиця 2 – Результат тестування на основі сторінки, що нараховує 20-й член послідовності Фібоначчі, веб-серверів (Тест 2)

Кількість запитів	Apache, с	Node.js, с
100	34.43	5.29
150	43.91	8.14
200	58.84	10.58
250	73.63	13.68
300	102.43	17.99
350	103.76	20.52
400	132.95	24.94
450	-	34.14
500	-	34.53
550	-	32.59
600	-	39.1
650	-	41.01
700	-	47.48
750	-	46.66
800	-	70.39
850	-	51.39

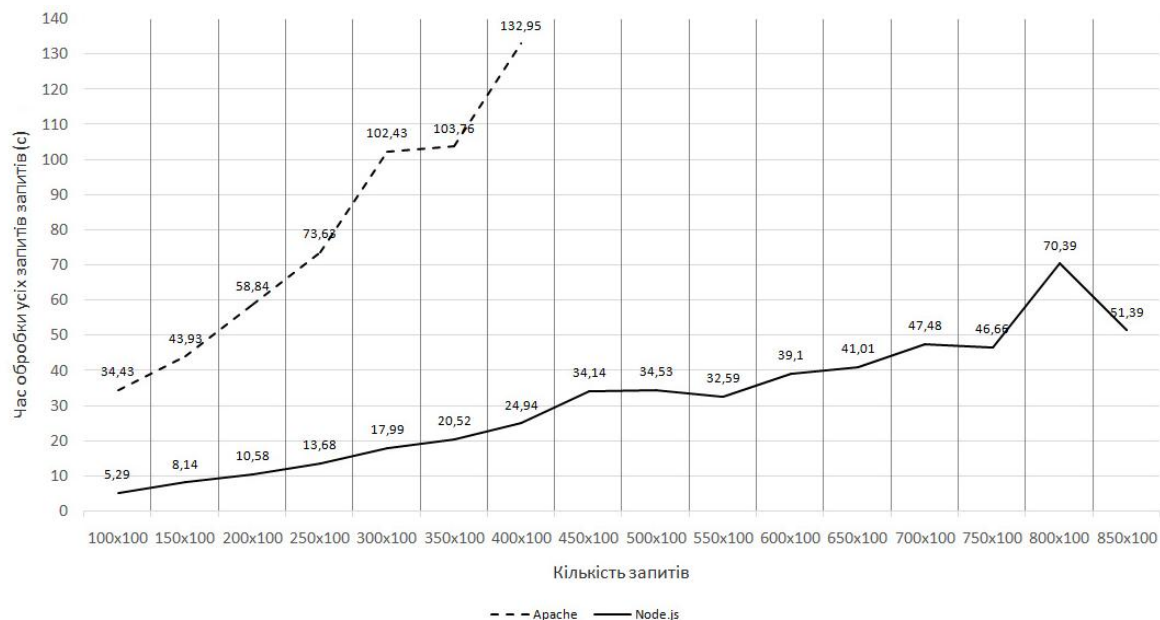


Рис. 2 – Залежність між кількістю запитів та усередненим часом затраченим на їх виконання (Тест 2)

Як видно із результату другого тесту, при виконанні більш трудомісткого процесу обчислення, сервер на базі платформи Node.js залишається в робочому стані і стабільно витримує високу кількість з'єднань, ніж веб-сервер Apache. Одержані з 2-го тесту результати демонструють значну перевагу роботи асинхронної архітектури обробки запитів. Щодо багатопоточної технології обробки запитів, то вона не лише дає збій при одночасній кількості запитів більшої за 400, але і демонструє набагато більший час обробки запитів, що пов'язані з незначними математичними розрахунками.

Для більш оптимізованої обробки запитів була налагоджена взаємодія веб-сервера Nginx, задача якого обробляти та віддавати статичні файли, з асинхронною технологією Node.js, на яку запропоновано покласти основну функцію роботи з базами даних, та інші динамічні операції. Спочатку було протестовано вищеописану конфігурацію сервера, який під час запиту зчитував дані із бази даних, та паралельно виводив статичний контент у вигляді картинок. Після цього були виконані певні налаштування Nginx для того, щоб він по вищеописаній схемі взаємодіяв з асинхронною технологією Node.js. В даному випадку Nginx виступав в ролі проксі-сервера, який обробляє запити, результатами яких є віддача статичних файлів клієнту. А технологія Node.js використовувалася для виконня певних обчислень (наприклад, реалізація асинхронності, взаємодія із базами даних тощо). Запропоновано "легенький" веб-ресурс, який має у контенті 10 статичних картинок та інформацію, що витягується з бази даних. Під поняттям "легенький" мається на увазі, що розроблено ресурс, у якому на обробку одного запиту, який пов'язаних із підтягуванням

картинок та взаємодією з БД, повинно витратитися якомога менше часу. Результат тестування при аналогічних вищеописаних умовах наведено у таблиці 3 та у вигляді графіка на рис. 3.

Таблиця 3 – Результат тестування веб-сервера оптимізованої конфігурації (Тест 3)

Кількість запитів	Apache, c	Node.js, c
100	9.97	7.1
150	15.4	9.64
200	19.57	13.37
250	25.48	16.32
300	30.21	20.02
350	37.3	23.39
400	66.12	27.33

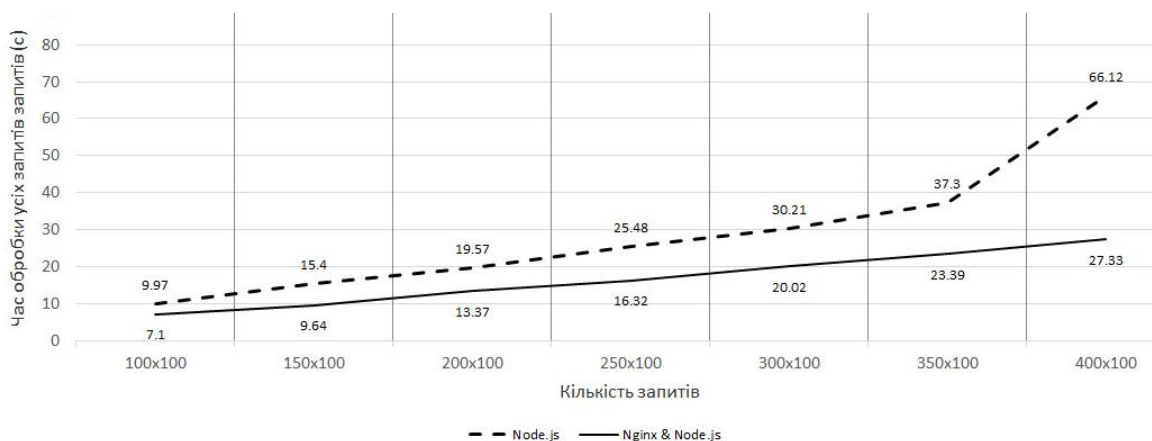


Рис. 3 – Залежність між кількістю запитів та усередненим часом затracеним на їх виконання (Тест 3)

Отриманий результат третього тесту демонструє приріст продуктивності роботи технології Node.js в парі з проксі-сервером Nginx. При кількості одночасних запитів меншій за 400 час їхньої обробки в даному програмно-архітектурному конфігураційному рішенні суттєво зменшився, що є дуже важливим для досягнення максимальної відмовитості в умовах великої кількості запитів та навантажень.

**Висновки.** Дослідивши та протестувавши роботу веб-серверів із синхронною та асинхронною моделлю обробки запитів, було отримано дані та залежності, з яких можна зробити наступні висновки. Спочатку було з'ясовано, що асинхронна технологія обробки запитів працює ефективніше (швидше і безвідмовніше), ніж багатопоточна. При цьому вона витримує більшу кількість одночасних запитів. Після цього було побудовано програмно-архітектурне конфігураційне рішення, яке полягає у перерозподілі часу обробки запитів між синхронною та асинхронною технологіями: на першу частину роботи, що пов'язана з підтягуванням статичної інформації, не було задіяно технологію асинхронності; а на другу, що пов'язана зі взаємодією з базами даних та з іншими динамічними операціями, відповідно задіяно дану технологію. Результат останнього тестування є позитивним і він показує, що розроблене в даній роботі конфігураційне рішення може використовуватися при побудові високонавантажувальних систем таких як: чат-сервери, соціальні мережі, багатопоточні веб-додатки, та інші веб-проекти, де необхідно опрацювати велику кількість запитів одночасно і в реальному часі.

1. Книга О.О. Аналіз переваг використання серверу Node.js для високонавантажених веб-додатків / Збірник тез Харківського національного економічного університету.
2. Пех П.А. Дослідження ефективності синхронних та асинхронних технологій програмування у процесі розробки веб-додатків / П.А. Пех, В.О. Бондарчук // Науковий журнал "Комп'ютерно-інтегровані технології: освіта, наука, виробництво". – Луцьк, 2015. – № 18 – С. 73-76.
3. <http://www.sworld.com.ua/index.php/ru/conference/the-content-of-conferences/archives-of-individual-conferences/march-2013>
4. <http://www.ibm.com/developerworks/ru/library/cl-nodejscloud>



УДК 004.023

Марченко О.О., аспірант

Марченко О.І., к.т.н., доцент,

Національний технічний університет України «Київський політехнічний інститут»

## КРИТЕРІЙ «ГЛИБИНА-ШИРИНА» ДЛЯ КОНТРОЛЮ ФОРМИ ДЕРЕВА ПОШУКУ ПРИ ВИКОРИСТАННІ МЕТОДУ МОНТЕ-КАРЛО

**Марченко О.О., Марченко О.І. Критерій «глибина-ширина» для контролю форми дерева пошуку при використанні методу Монте-Карло.** У статті, на основі аналізу методу пошуку в дереві методом Монте-Карло (MCTS) та особливостей його поведінки при використанні в різних випадках, запропоновано оригінальний критерій «глибина-ширина» дерева для контролю форми дерева пошуку, а також для оцінювання і вибору потенційно кращих варіантів продовження такого пошуку. Передбачається, що цей критерій може бути використаний для розробки нових способів покращення методу MCTS та створення більш швидких реалізацій MCTS для конкретних застосувань.

**Ключові слова:** задачі штучного інтелекту, дерева ігор, пошук в дереві, метод Монте-Карло, MCTS, методи покращення MCTS.

**Марченко А.А., Марченко А.И. Критерий «глубина-ширина» для контроля формы дерева поиска при использовании метода Монте-Карло.** В статье, на основе анализа метода поиска в дереве методом Монте-Карло (MCTS) и особенностей его поведения при использовании в различных случаях, предложен оригинальный критерий «глубина-ширина» дерева для контроля формы дерева поиска, а также для оценивания и выбора потенциально лучших вариантов продолжения такого поиска. Предполагается, что этот критерий может быть использован для разработки новых способов улучшения метода MCTS и создания более быстрых реализаций MCTS для конкретных приложений.

**Ключевые слова:** задачи искусственного интеллекта, деревья игр, поиск в дереве, метод Монте-Карло, MCTS, методы улучшения MCTS.

**Marchenko O.O., Marchenko O.I. “Depth-breadth” criterion for control of the search tree shape using Monte-Carlo method.** Basing on analysis of the Monte-Carlo tree search (MCTS) method and specific features of its behavior for various cases of usage, a new tree “depth-width” criterion for controlling search tree shape along with for estimation and choose of potentially better options for the search continuation. It is supposed that this criterion could be used for development of new techniques of MCTS method improvement and for creation of faster MCTS implementations for concrete applications.

**Keywords:** artificial intelligence tasks, game trees, tree search, Monte-Carlo method, MCTS, MCTS improvement methods.

*Рис. 3. Лім. 9.*

### Постановка наукової проблеми.

Розробка нових методів та алгоритмів, що можуть прискорити обчислення задач, час рішення яких зараз є занадто великим, була і залишається актуальною проблемою сьогодення. Особливо це важливо в галузі задач штучного інтелекту, яка відзначається надзвичайно великим обсягом даних і високими вимогами до швидкості їх обробки. Характерними і популярними задачами штучного інтелекту, на яких, як правило, виконується апробація нових методів, алгоритмів, структур даних є складні інтелектуальні ігри, наприклад, такі як ігри в шахи та Го.

Пошук по дереву з використанням методу Монте-Карло (Monte Carlo Tree Search – MCTS) [1, 2] є методом пошуку рішень у заданій області за допомогою випадкових значень із заданого простору значень і побудови дерева пошуку за отриманими результатами. Цей метод вже встиг суттєво вплинути на розв'язання задач штучного інтелекту, а особливо складних ігрових задач. За час існування методу MCTS було створено багато його варіантів та модифікацій з метою прискорення процесу пошуку та покращення рішень, що приймаються на його основі, але потужність концепції цього методу залишає ще дуже великий простір для нових досліджень для покращення пошуку в задачах штучного інтелекту.

### Аналіз досліджень.

Методу MCTS присвячено вже багато досліджень і публікацій, на декілька з них автори посилаються в даній роботі. Найбільш фундаментальною узагальнюючою роботою з методу MCTS є

огляд 2012 року, підготовлений десятима провідними вченими під керівництвом Камерона Броуне, що досліджують цей метод [2], але, незважаючи на її ґрунтовність, вона не є строгою класифікацією. Крім того, після 2012 року з'явилися не тільки нові модифікації та покращення існуючих способів, але й принципово нові варіанти реалізації MCTS.

Структура та критерії класифікації різних способів реалізації методу MCTS, його покращення та паралелізації були розглянуті авторами в [1]. Автори вважають, що на основі систематизованої інформації про способи та підходи до покращення пошуку методом MCTS можна визначити нові критерії та місця в схемі роботи MCTS для покращення процесу пошуку.

**Метою даної роботи** є аналіз процесу пошуку в дереві методом Монте-Карло (MCTS) і особливостей його поведінки при використанні для різних задач та в різних випадках пошуку, а також визначення нового критерію для оцінювання і вибору потенційно кращих варіантів продовження такого пошуку.

### Пошук в дереві з використанням методу Монте-Карло (MCTS)

Дерева ігор використовуються комп'ютерними гравцями для вибору ходу гри для таких ігор, як шахи чи Го. Використання повного дерева гри (того, яке містить всі можливі варіанти і послідовності гри) дозволяє реалізувати сильного гравця при застосуванні для пошуку у дереві методів Minimax та Negamax [3]. Однак ці методи потребують достатньо великого об'єму пам'яті та обчислювальної потужності для більшості ігор і тому прямі алгоритми Minimax та Negamax, як правило, не використовуються на практиці на повному дереві. Замість цього використовуються їх варіанти з Alpha-Beta-відсіканням, що дозволяє зменшити необхідні обчислювальні ресурси та зробити дерево більш підходящим для пошуку [3]. Однак для багатьох ігор, дерева яких мають велику степінь розгалуження (наприклад гра Го), використання методів з Alpha-Beta-відсіканням також є недостатнім для того, щоб створити сильного гравця [4].

Кожне піддерево дерева пошуку методом MCTS відповідає поточній позиції гри як коренева вершина, а кожна вершина в цьому піддереві зберігає інформацію про кількість вигравів і загальну кількість разів виконання гри через цю вершину, які позначаються у вершинах через дробову риску (кількість вигравів / кількість ігор) [2]. На рисунку 1 наведено приклад дерева MCTS, де в кожній вершині показано результат багатократного моделювання гри від цієї вершини. Всі термінальні (листові) вершини і нетермінальні вершини, до яких зберігається доцільність додавання нових листів-нащадків, формують межу (передній фронт) дерева MCTS.

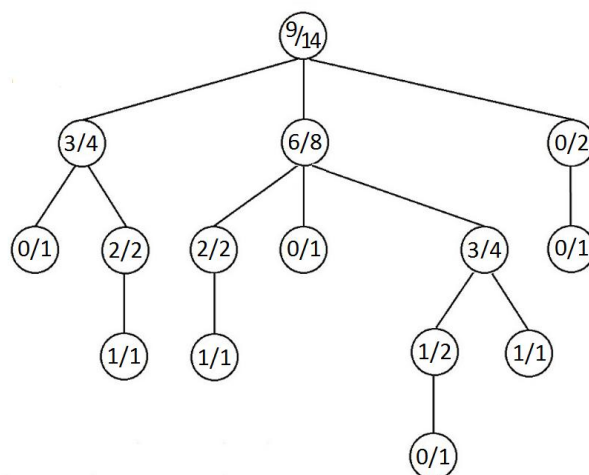


Рис.1. Приклад дерева MCTS

Кожне моделювання гри методом Монте-Карло в стандартному варіанті реалізації є виконанням процесу гри від деякої поточної позиції до кінця гри, вибираючи ігрові ходи випадковим чином [2]. Процес моделювання гри, починаючи від однієї й тієї самої вершини дерева (позиції гри), виконується багато разів, після чого відношення перемог до загальної кількості виконання гри формує сумарну оцінку цієї позиції. Після цього виконується переобчислення балів для кожної вершини дерева пошуку MCTS на шляху від поточної вершини до кореня дерева.

Загальний підхід методу MCTS [2] складається з наступних чотирьох послідовних етапів:

- етап вибору;
- етап розширення;
- етап моделювання;
- етап переобчислення.

Розглянемо більш детально етап вибору. На цьому етапі виконується процес спуску по дереву вниз з метою знаходження найкращого ходу для продовження гри. При цьому кожна наступна вершина, через яку йде спуск, вибирається згідно з певною політикою вибору, яка називається політикою дерева (Tree Policy).

Основна мета цієї політики полягає в тому, щоб зменшити час, витрачений на розгляд недоцільних ходів. Ця мета досягається за допомогою балансування між дослідженням нових, ще не змодельованих, ходів по відношенню до використання вже змодельованих наступних ходів. Використання – це зосередження на кращій в даний час вершині (ході), в той час як дослідження – це розглядання іншої, на даний момент не найкращої вершини (ходу), але яка, можливо, в кінцевому результаті, стане кращою.

Наприклад, однією з найбільш популярних політик дерева, яка використовується, в тому числі, для комп'ютерної гри Го, є політика з використанням методу верхніх довірчих границь – Upper Confidence Bounds (UCB) – або варіант на його основі. Згідно одному з таких варіантів, що називається UCB1 [2], відбирається вершина, для якої рівняння, показане на рисунку 2, дає максимальне значення.

$$\text{рейтинг (вершина)} = \frac{\text{виграші (вершина)}}{\text{кількість ігор (вершина)}} + C * \sqrt{\frac{\log(\text{кількість ігор (предок)})}{\text{кількість ігор (вершина)}}$$

Рис.2. Рівняння UCB1

Це рівняння складається з двох доданків (умов): перший доданок – рейтинг виграшу вершини, в той час як другий доданок – апроксимація верхньої границі на довірчому інтервалі. Іншими словами, максимізація цього рівняння вибирає вершину, яка, можливо, в кінцевому результаті є кращою. Константа C підбирається таким чином, щоб оптимізувати баланс між процесами дослідження і використання (при більшому значенні C буде виконуватись більше досліджень). Спосіб реалізації MCTS, який для обчислення рейтингу вершин (ходів) використовує метод UCB, називається методом «верхніх довірчих границь, застосованих до дерев» – “Upper Confidence bounds applied to Trees” (UCT).

Коли процес вибору закінчився, і отримано конкретну вершину, виконується крок розширення – процес додавання нової вершини до цієї вершини дерева. Таким чином межа дерева постійно розширюється і стає в подальшому «більш глибокою». Процес розширення збільшує точність і відповідність дерева поточній ситуації гри, роблячи його більш реалістичним представленням можливих наслідків. По суті, розширюючи дерево ми робимо наступний крок у гри.

Після того, як ми додали нову вершину до дерева, тобто, зробили наступний можливий хід, починається крок моделювання – процес виконання моделювання гри від цієї вершини до закінчення гри. При моделюванні ми розглядаємо додану до дерева нову вершину як початковий стан гри, після

чого відбувається гра відповідно до її правил. Процес моделювання можна абстрагувати функцією, яка поверне одне з двох можливих значень: виграш чи програш (в деяких варіантах MCTS повертається також третє значення – нічия). Мета даного етапу – зібрати інформацію про те, скільки зіграних від цієї вершини партій було виграно, тобто, наскільки ефективним був цей крок, обраний на етапах вибору та розширення.

Після того, як моделювання закінчується і результат отриманий, починається крок переобчислення – процес розповсюдження цього результату по дереву від початкової вершини моделювання до кореня. Як правило, новий результат об'єднується з попередніми результатами вважаючи, що всі попередні результати мають однакову вагу із отриманим, але можуть бути використані й інші функції переобчислення [5, 6].

Процес моделювання від вершини з подальшим переобчисленням результату подібний до взяття вибірки із певного ряду даних. Але оскільки при цьому змінюється дерево, то змінюється і ряд, з якого робиться вибірка, і тому цей ряд є нестационарним, тобто таким, який містить основну тенденцію розвитку подій.

Загальний підхід методу MCTS можна записати у вигляді алгоритму верхнього рівня наступним чином:

```
while ЗаданийОбчислювальнийБюджет ще не закінчився do  
Вершина ← ВибірВершиниДерева()  
Листок ← РозширенняВідВершини(Вершина)  
Результат ← МоделюванняВідЛистка(Листок)  
ПереобчисленняВідЛистка(Листок,Результат)  
end
```

У якості обчислювального бюджету використовується або певний період часу (час роздумів на один хід гри), або задана кількість ітерацій MCTS. Після того, як відбулася визначена кількість моделювань, чи пройшов визначений для моделювання час, пошук MCTS зупиняється, після чого відбирається найкращий хід відповідно до деяких критеріїв, наприклад, обирається вершина із найбільшою кількістю успішних моделювань. При цьому, піддерево, в якому обрана вершиною є коренем, зберігається як результат у поточному дереві MCTS, в той час як інші вершини відкидаються. Це дозволяє при наступних пошуках, що будуть виконуватись на цьому піддереві, брати його як початкове дерево, і збільшує загальну кількість результатів моделювань в цьому піддереві.

### **Критерій «глибина-ширина» для контролю форми дерева пошуку MCTS**

Ігри та інші прикладення, де застосовується MCTS, можуть мати різну степінь розгалуження можливих дій (ходів) на кожній вершині дерева пошуку. Глибина дерева пошуку узгоджується з визначеннями глибини та ясності гри, які дав Томпсон у [7]. Якщо дерево пошуку для певної гри є глибоким і має низьку степінь розгалуження, то продумування на велику кількість ходів наперед стає нескладною задачею, оскільки дерево пошуку буде «вузьким». В такому випадку ясність гри буде високою і навіть недосвідчений гравець-людина зможе зрозуміти цінність подальших ходів і який з них є найкращим, в результаті чого вибір найкращого ходу з невеликої кількості варіантів може стати занадто простим рішенням. З іншої сторони, використання рівняння UCS1 для вибору ходів в грі за комп'ютер при малій степені розгалуження в дереві пошуку також буде вибирати найкращі ходи, оскільки вони будуть добре передбачувани, і гра стане повторюваною і нецікавою. З іншої сторони, якщо дерево пошуку MCTS для певної гри має високу степінь розгалуження, гравцю-людині буде дуже важко охопити і продумати багато варіантів продовження гри [8].

Накопичений досвід використання методу MCTS показав, що, в залежності від степені розгалуження можливих варіантів рішень на вершинах дерева, для одних прикладень (ігор) ефективність пошуку кращих ходів вище, якщо дерево зростає переважно в ширину, а для інших – в

глибину [2].

На процес більшого зростання дерева пошуку вглибину чи вширину впливають два фактори:

- значення константи  $C$  у рівнянні, показаному на рис.2, яка регулює баланс між кількістю досліджень (exploration) та кількістю використань (exploitation);
- відношення кількості виконання етапу вибору (етап 1) порівняно до кількості моделювань, виконаних на етапі 3.

Згідно першого фактору, було встановлено, чим більша константа  $C$  у рівнянні, показаному на рис.2, то тим більше буде виконуватись досліджень і тим більше дерево буде зростати вширину [2]. Згідно другого фактору, було встановлено, що чим більше виконується моделювань порівняно до кількості процесів вибору, то дерево більше зростає вглибину. Константа  $C$  фактично є параметром, який контролює форму дерева пошуку MCTS. При менших значеннях  $C$  створюється глибше і вужче дерево пошуку, в той час, як при більших значеннях  $C$  буде побудоване менш глибоке але більш широке дерево MCTS [9]. Рівняння UCB1 автоматично фокусує пошук на поточних найкращих варіантах подальших ходів одночасно з наданням можливості достатнього дослідження для того, щоб знайти потенційно найкращий хід, якого в дереві ще нема.

В роботі [8] для контролю за формою дерева пошуку, крім константи  $C$ , використовуються критерії глибини дерева та степені розгалуження кожної вершини (критерій ширини дерева).

Автори даної статті вважають, що критерій ширини дерева у вигляді степені розгалуження кожної вершини є недостатньо об'єктивним і змістовним для контролю за формою дерева пошуку, оскільки такий критерій показує тільки локальну ширину розгалуження для поточної вершини і потенційну (але не фактичну) ширину всього дерева, і пропонують замість степені розгалуження вершини використовувати відносний критерій «глибина-ширина», що обчислюється за певними правилами для кожної вершини дерева. Оскільки правила обчислення цього критерію є дуже простими, то його використання не вплине на час роботи пошуку.

Обчислення критерію «глибина-ширина» будемо виконувати за наступними правилами:

- у всіх термінальних вершин дерева (листіків) і глибина, і ширина їх піддерев дорівнюють одиниці;
- глибина піддерева нетермінальної вершини обчислюється як максимальне значення зі значень глибин піддерев її нащадків плюс одиниця;
- ширина піддерева нетермінальної вершини обчислюється як сума значень ширин піддерев її нащадків.

На рисунку 3 показаний приклад розгалуженого дерева з обчисленими значеннями критерію «глибина\ширина».

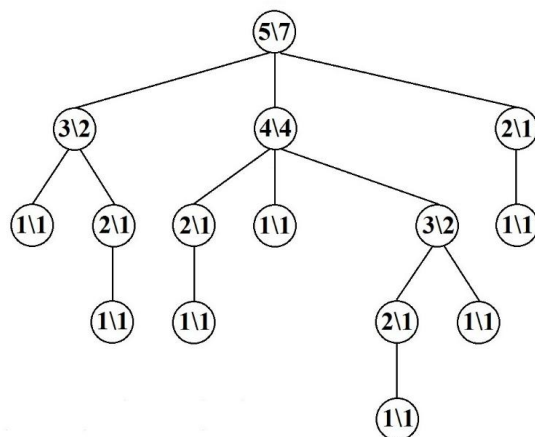


Рис.3. Розгалужене дерево з обчисленими значеннями критерію «глибина\ширина»

Перша цифра відповідає глибині піддерева, а друга цифра – ширині піддерева. Щоб відрізнити ситуацію, коли у вершині показаний критерій «глибина-ширина», а не критерій «кількість вигравів/кількість моделювань», на відміну від останнього, для розділу двох значень використовується символ «\» (back-slash), а не «/».

Якщо для певних задач відомо, яка форма дерева MCTS (більш глибока чи більш широка) є ефективнішою, то враховуючи таку наявну початкову інформацію, для цих задач з'являється можливість доцільного виконання динамічного контролю форми дерева за критерієм «глибина-ширина», направляючи подальшу побудову дерева пошуку у потрібному напрямку.

Розглянемо, які переваги при виконанні пошуку за методом MCTS можна отримати з допомогою запропонованого критерію «глибина-ширина». Цей критерій надає можливість без уповільнення роботи основної частини будь-якої реалізації методу MCTS, одночасно з роботою цієї основної частини, визначити нові перспективні вершини для продовження гри і виконати додаткове дослідження найбільш перспективних гілок дерева MCTS на вільних в цей час апаратних ресурсах.

Крім того, з'являється також можливість більш доцільно виконувати розпаралелення алгоритму на доступних в даний момент апаратних ресурсах. Якщо для виконання паралельної обробки доступна комп'ютерна система з невеликою кількістю комп'ютерів/процесорів/процесорних ядер, то можливості динамічної корекції критерію «глибина-ширина» дерева є обмеженими. З іншої сторони, зі зростанням доступної кількості та потужності вузлів комп'ютерної системи, з однієї сторони зростають можливості розпаралелення процесу пошуку, але, з іншої сторони, зростає також і складність організації ефективної паралелізації. Таким чином можна забезпечити коректне визначення потрібних апаратних ресурсів для ефективного розпаралелення на нижчих рівнях дерева гри, виходячи з інформації, що знаходиться у вузлах-предках без подальшого перерозподілу апаратних ресурсів (ядер, процесорів, потоків тощо) з можливим додатковим перерозподілом (перезаписуванням) інформації в різних типах та блоках пам'яті.

#### **Висновки**

Пошук у дереві з використанням методу Монте-Карло MCTS може показувати гарні результати у багатьох областях застосування, але не у всіх. Для одних задач MCTS виконує пошук краще, якщо форма дерева зростає переважно в ширину, а для інших – якщо в глибину. Паралелізація може бути виконана згідно декількох різних підходів. На основі аналізу методу MCTS та існуючих підходів контролю за процесом пошуку і формою дерева пошуку був запропонований оригінальний критерій «глибина-ширина» для такого контролю, який обчислюється за певними правилами. Відзначається, що підтримка такого критерію в процесі пошуку не призведе до суттєвих часових витрат. Подальшим напрямком досліджень можуть бути як деталізація застосування запропонованого критерію під час пошуку методом MCTS, так і розробка методів паралелізації на основі цього критерію.

1. Марченко О. І. Структура та критерії класифікації способів реалізації та покращення пошуку по дереву методом Монте-Карло / О. І. Марченко, О. О. Марченко, М. М. Орлова. – Комп'ютерно-інтегровані технології: освіта, наука, виробництво, 2015. – № 21. – С. 51–57.
2. Cameron Browne and others. A Survey of Monte Carlo Tree Search Methods // IEEE Trans. on Computational Intelligence and AI in Games. - Vol. 4. - No. 1. - March 2012, pp.1-48.
3. CS312 Recitation 21. Minimax search and Alpha-Beta Pruning [Електронний ресурс]. Режим доступу: <http://www.cs.cornell.edu/courses/cs312/2002sp/lectures/rec21.htm>
4. В. Bouzy and T. Cazenave, "Computer Go: an AI oriented survey" // Artificial Intelligence. Vol.132. – Issue 1, October 2001, pp.39-103.
5. Thomas Keller, Malte Helmert. Trial-based Heuristic Tree Search for Finite Horizon MDPs // [Електр. Ресурс]. Режим доступу: <http://www.aaai.org/ocs/index.php/ICAPS/ICAPS13/paper/view/6026>.
6. Zohar Feldman, Carmel Domshlak. On MABs and Separation of Concerns in Monte-Carlo Planning for MDPs // Proc. 24th Int. Conf. Automat. Plan. Sched., pp.120-127.
7. J. M. Thompson. Defining the abstract. The Games Journal, 2000. [Електронний ресурс]. Режим доступу: <http://www.thegamesjournal.com/articles/DefiningtheAbstract.shtml>
8. Gareth M. J. Williams. Determining Game Quality Through UCT Tree Shape Analysis. MSc. Thesis, Imperial College London, 2010.
9. Joel Veness, Kee Siong Ng, Marcus Hutter, William Uther, David Silver. A Monte-Carlo AIXI Approximation // Journal of Artificial Intelligence Research 40 (2011), pp.95-142.

УДК 514.18

Пех П.А., Бурчак І.Н., Тихомиров В.В.  
Луцький національний технічний університет**РОЗРОБКА МОДЕЛЕЙ В AUTODESK MAYA З МЕТОЮ  
ПОДАЛЬШОГО ВИКОРИСТАННЯ У ІГРОВОМУ РУШІІ UNREAL ENGINE 4**

**Пех П.А., Бурчак І.Н., Тихомиров В.В. Розробка моделей в Autodesk Maya з метою подальшого використання у ігровому рушії Unreal Engine 4.** У статті розкрито важливість застосування новітніх технологій сфери тривимірного моделювання для реалізації процесу віртуального відтворення історичних пам'яток та відтворення реальних споруд, що існують на даний час. Запропоновано спосіб моделювання та проектування об'єкту, коли зразком являється реальний прототип – його місцерозташування, розміри та ін.

**Ключові слова:** метод, моделювання, об'єкт, технології, проектування та моделювання, історична пам'ятка.

**Тихомиров В.В. Разработка моделей в Autodesk Maya с целью дальнейшего использования в игровом движке Unreal Engine 4.** В статье раскрыта важность применения новейших технологий сферы трехмерного моделирования для реализации процесса виртуального воссоздания исторических памятников и воссоздания в качестве 3D объекта реальных сооружений, существующих в настоящее время. Предложен способ моделирования и проектирования объекта, когда образцом является реальный прототип – его месторасположение, размеры и др.

**Ключевые слова:** метод, моделирование, объект, технологии, проектирование и моделирование, памятник истории.

**Pekh P.A., Burchak I.N., Tykhomirov V.V. Development of models to Autodesk Maya for later use in the Unreal Engine 4.** The article reveals the importance of the use of three-dimensional modeling of the latest in technology for the realization of process of virtual reconstruction of historical monuments and reconstruction of a 3D object as the real structures that exist at the moment. The method by which you can carry out the modeling and design of an object by taking a sample of a real prototype – its location, size, etc.

**Keywords:** method, modeling, object, technology, design and modeling, the historical monument.

**Постановка проблеми.** Факт ознайомлення людини з історичною спадщиною рідного краю, в тому числі архітектурних пам'яток, є одним із важливих аспектів її формування як особистості. Замок Любарта є однією з таких славетних архітектурних пам'яток. Як відомо, не усі пам'ятки минулого добре збереглися до наших днів. Часто вони потребують виконання великого обсягу робіт щодо відновлення їх початкового стану. Реконструкція цих об'єктів може зайняти багато часу, а витрати на ці роботи будуть значними. У даній статті розглядається проблема створення 3D моделей пам'яток архітектури, що пов'язане з вибором способу створення віртуального середовища, алгоритмів виконання роботи та подальшого перенесення моделей в ігровий рушій.

Виходячи з поставлених задач, виникають питання, які стосуються проблеми створення віртуального середовища як такого. Окрім аналізу раціональності будівництва та реальної реконструкції, необхідно проаналізувати способи, завдяки яким можна досягти бажаного результату без особливих затрат, як фінансових, так і людських ресурсів.

Перш, ніж розглянути питання моделювання навколишнього середовища, необхідно визначитися:

- з розміщенням об'єкту історичної будівлі, яка буде головною у проєкті;
- з програмним забезпеченням, завдяки якому можна буде здійснити моделювання;
- з плагінами, які необхідні для експортування у ігровий рушій

та обрати сам ігровий рушій із величезної кількості існуючих.

**Аналіз останніх досліджень та публікацій.** Побудова 3D моделей пам'яток архітектури розглядалася багатьма авторами [ ], однак в жодній з них не досліджувався замок Любарта. Створення 3D моделі для цього об'єкту має ряд особливостей, які і вирішуються у даній роботі. Вибір програмного забезпечення - це дуже важлива частина, яка реалізується на етапі підготовки. , а саме слід обирати опираючись на власний досвід та функції, які надає те чи інше рішення. До програм, які дозволяють працювати з 3D об'єктами, можна віднести найпопулярніші: 3Ds Max, Maya, Blender. Усі ці програмні продукти відносно однакові за функціоналом. Для виконання поставленої мети абсолютно достатньо тих інструментів, що вони пропонують. Слід зауважити, що 3Ds Max та Maya - це платні продукти, а Blender – безкоштовний програмний пакет, проте існує можливість отримання безкоштовної ліцензії терміном 3 роки на програми компанії Autodesk. Отже, у виборі між 3Ds Max`ом та Maya ми надаємо перевагу другій програмі, причому



вирішальним аргументом є наш особистий досвід створення просторових моделей. Якщо порівнювати ці два продукти з точки зору функціоналу, то особливу різницю складно встановити.

Після того, як вибрано моделюючу програму, необхідно перейти до наступного етапу планування проекту, а саме дуже важливим є отримання референсу, або зображення, на яке буде спиратись розробник, що моделює об'єкти, при створенні 3D моделі. Це можна зробити просто сфотографувавши будівлю, споруду, тощо. Також можна скористатись інтернет-ресурсами і завантажити необхідні фотографії. Отримана фотографія є чудовим прикладом, проте існує одна проблема. За відсутності будь-якого креслення, або плану будівлі, як це часто буває, необхідно власноруч визначити масштабування та розмітити усі розміри. Маючи необхідне зображення, знаючи висоту та ширину споруди, можна визначити розміри, які будуть потрібні для відтворення ключових вузлів у програмі моделювання 3D об'єктів. Прикладом накладання розмірів на картинку є рис. 1а, рис 1б. Він показує як використовуючи мінімальні знання про розміри будівлі можна виконати розмітку:

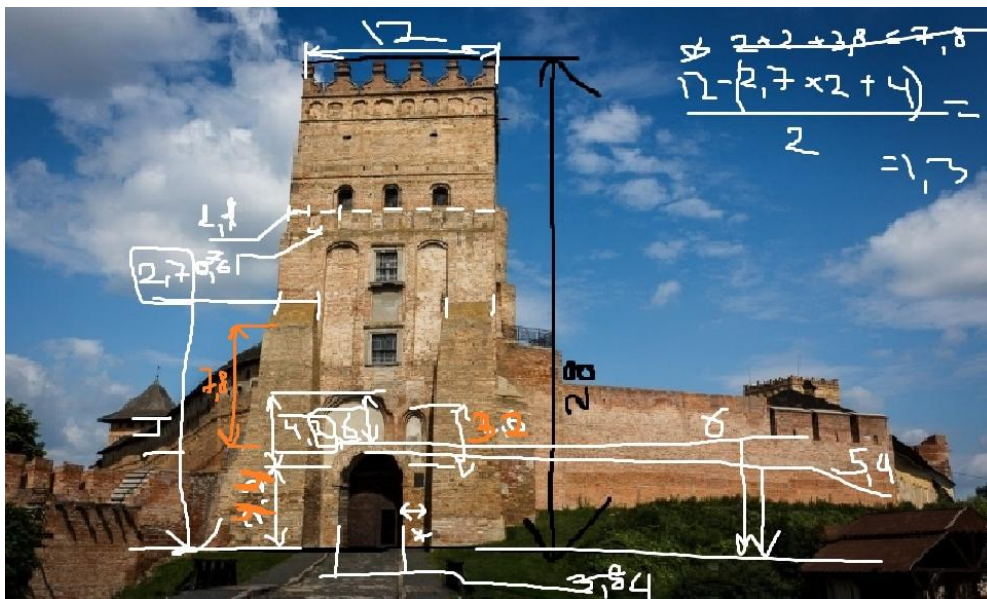


Рисунок 1а. Встановлення розмірів (мірило 1:200)

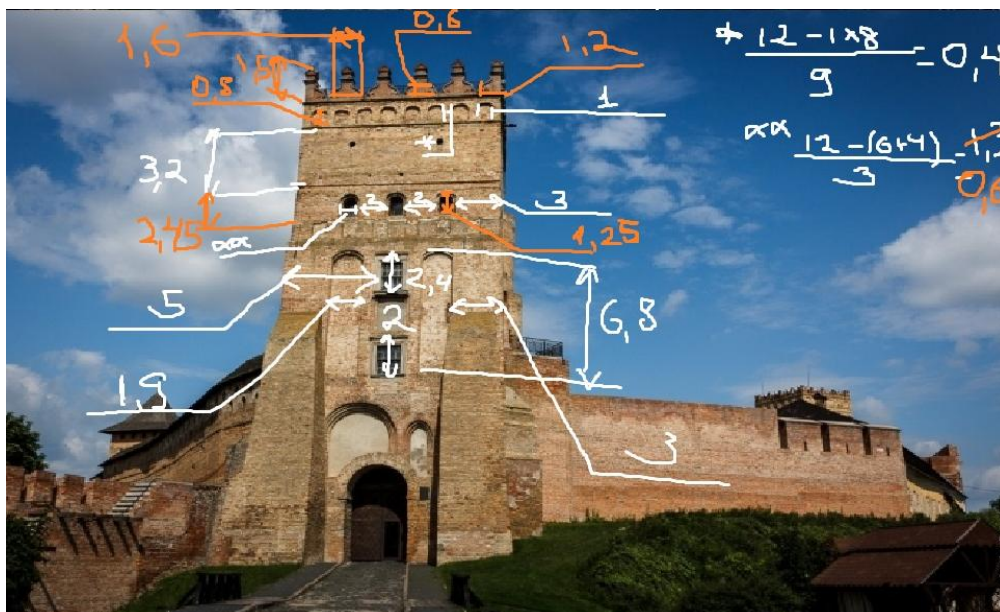


Рисунок 1б. Встановлення розмірів (мірило 1:200)



Отримавши розміри, які задовільняють розробника, варто перейти до наступного етапу створення проекту – розміщення споруд у програмі моделювання згідно географічного розташування. Це робиться для того, щоб під час імпортування в ігровий рушій масштаби та місцезнаходження всього, що є в проекті, відповідало дійсності, а відносність до сторін світу повинна бути збережена. Існує декілька способів, аби зробити це, проте найвідомішим та найлегшим є вирізання потрібного фрагменту – локації з різноманітних карт. В такому разі можна використовувати додатки Google Maps, Yandex Maps, Google Earth. Шляхом порівняння усіх програм було вирішено використати Google Earth. Завдяки своєму користувацькому інтерфейсу та зручності, Google Earth є найкращим рішенням для подібної роботи. Скориставшись інструментом для отримання зображень з програми було витягнуто наступний рисунок (рис .2):



Рисунок 2. Супутникове зображення замку Любарта та його околиць

Витягнувши супутниковий знімок з програми Google Earth, необхідно імпортувати його в програму для моделювання та розташувати умовні об'єкти по всьому периметру, а саме три головні вежі та стіни (рис. 3). Таким чином маючи масштаби моделей та основу для точного їх розташування можна отримати відповідну модель усього комплексу:

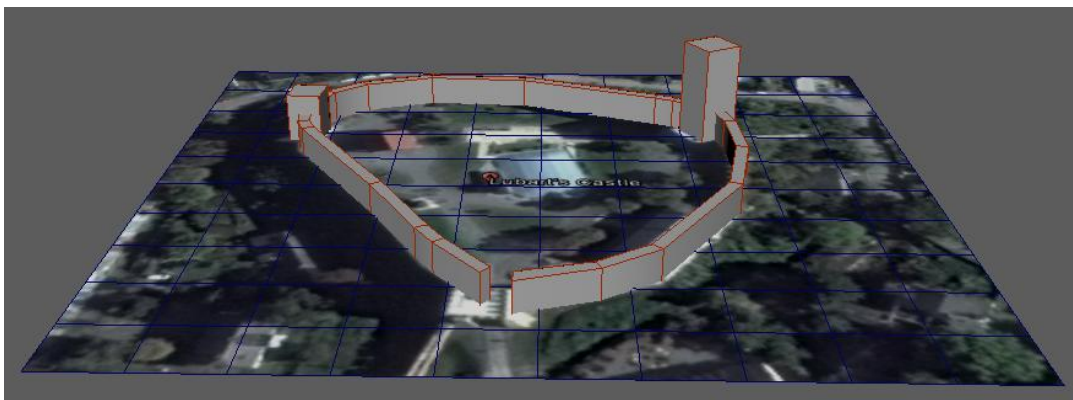


Рисунок 3. Розміщення об'єктів на фрагменті супутникової карти

**Основна частина.** На цьому кроці етап планування закінчений і необхідно переходити до наступного – створення власне моделі, яка складається з деталей, що були розмічені на рис. 1.а та рис1.б. Точне моделювання об'єкту є можливим завдяки стандартному набору інструментів,

запропонованих програмою Autodesk Maya, серед яких: створення примітивів (куб, сфера, циліндр), об'єднання границь, полігонів, вершин, різноманітні булеві функції, а саме: додавання множин точок, віднімання, визначення спільної області, та ін. Використовуючи їх своєчасно у потрібних ситуаціях можна досягти найбільш ефективних та реалістичних результатів, які задовільняють розробника та замовника. Після обробки моделі вищевказаними інструментами, зважаючи на попередні заміри було отримано наступний об'єкт (рис. 4):

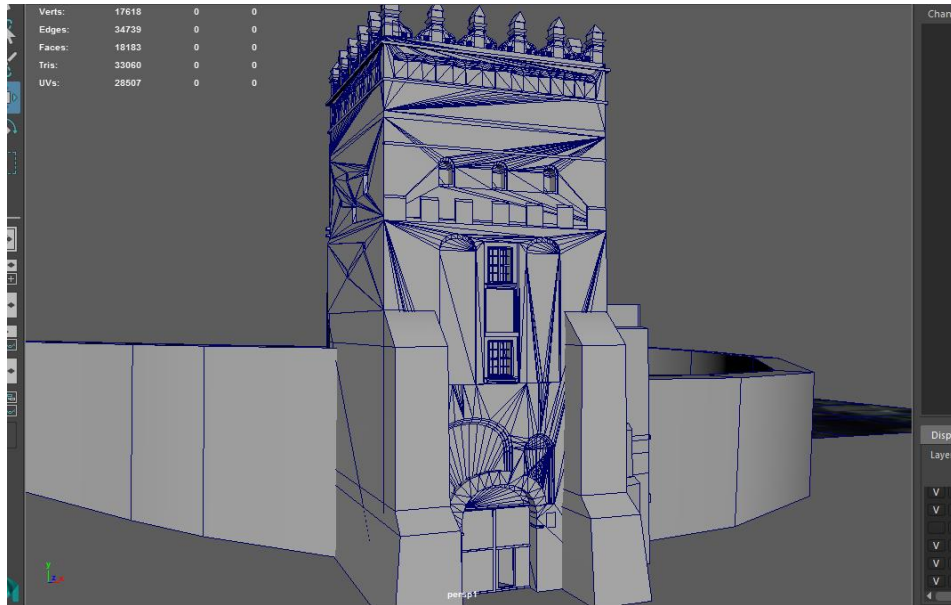


Рисунок 4. Модель в'їзної вежі замку Любарта

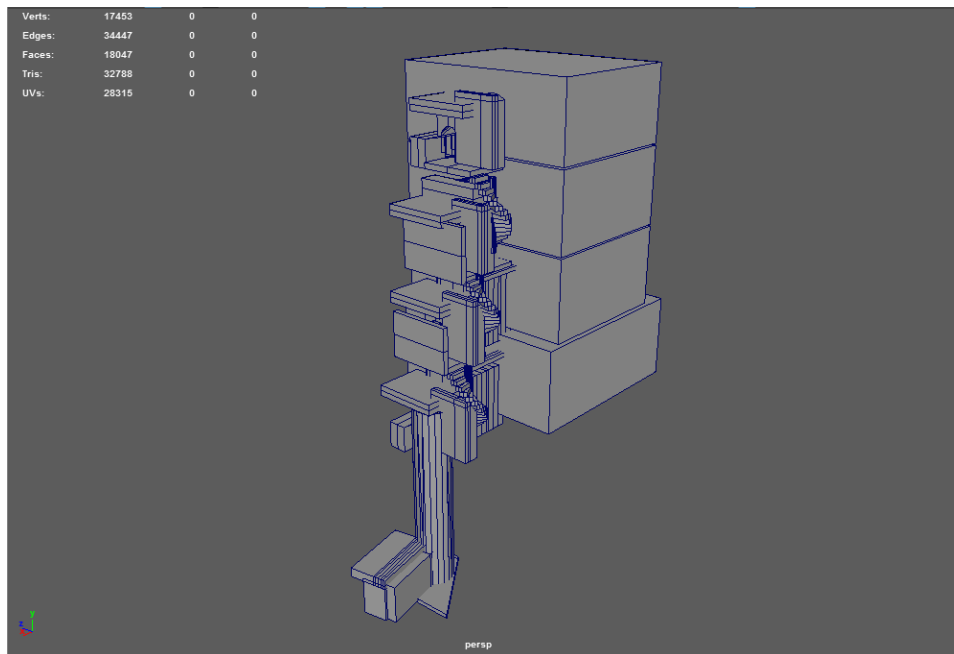


Рисунок 5. Внутрішня частина в'їзної вежі замку Любарта

Працюючи з примітивами та інструментами, немов архітектор, розробник створює і інші частини локації, які необхідні для досягнення повноти проекту. Розглядаючи конкретно даний приклад слід зауважити, що в'їзна вежа замку Любарта повинна бути змодельована не тільки ззовні а й зсередини, що робить проект більш комплексним і складнішим водночас, проте робота повинна бути виконана, адже відновлення такої історичної пам'ятки з усіма деталями є одним з головних пунктів даного проекту. Зауваживши цю річ, також варто було б наголосити на тому факті, що

внутрішня частина в'їзної вежі фізично буде знаходитись поза межами досяжності користувача. Тобто у ігровому рушії переходи здійснюватимуться природньо, проте насправді «нутрощі» будуть відділені від «тіла» вежі та перенесені у інше місце, недоступне користувачеві. Це робиться з наступних міркувань: покращення швидкодії самого проекту, раціональність створення оточення, а саме зручність моделювання ззовні набагато вища ніж зсередини. Як зазначалось вище, виготовлення внутрішньої частини (рис. 5) буде вестись окремо, але з урахуванням усіх масштабів та пропорцій. Використовуючи примітивні та стандартні інструменти для роботи з ними (переміщення, сполучення, та ін.) було змодельовано деякі частини внутрішніх приміщень замку Любарта, а саме передкімнатні коридори, гвинтові сходи та деякі інші деталі.

Після побудови в'їзної вежі зсередини та ззовні варто імпортувати ці моделі в ігровий рушій, перевіряючи тим самим працездатність спеціальних плагінів (рис. 6) для імпортування та експортування, коректність відображення моделі вцілому. Перехід між зовнішнім та внутрішнім рівнем здійснюватиметься за допомогою статичних об'єктів – тригерів, та відповідного скрипту, який створюється на візуальній мові програмування рушія unreal engine 4 BluePrint.

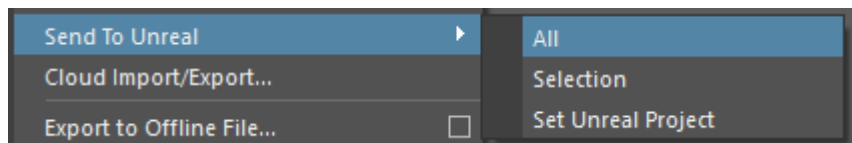


Рисунок 6. Плагін експортування об'єктів/сцени в Unreal Engine 4

У пункті «Файл», головного меню програми Maya є вкладка Send To Unreal, при наведенні курсором на яку впливає 3 різні режими експортування, а саме «експортувати усі об'єкти», «вибраний об'єкт», та «встановити проект Unreal», тобто перенести абсолютно всі об'єкти, навіть не 3D об'єкти такі як камери, світло і т.д. Обравши необхідну модель, натиснувши «Selection» її буде імпортовано в ігровий рушій, де далі над ним можна виконувати різноманітні перетворення, включаючи утворення власної колізії. Імпортувавши в'їзну вежу в Unreal Engine 4 були отримані наступні результати (рис 7.):

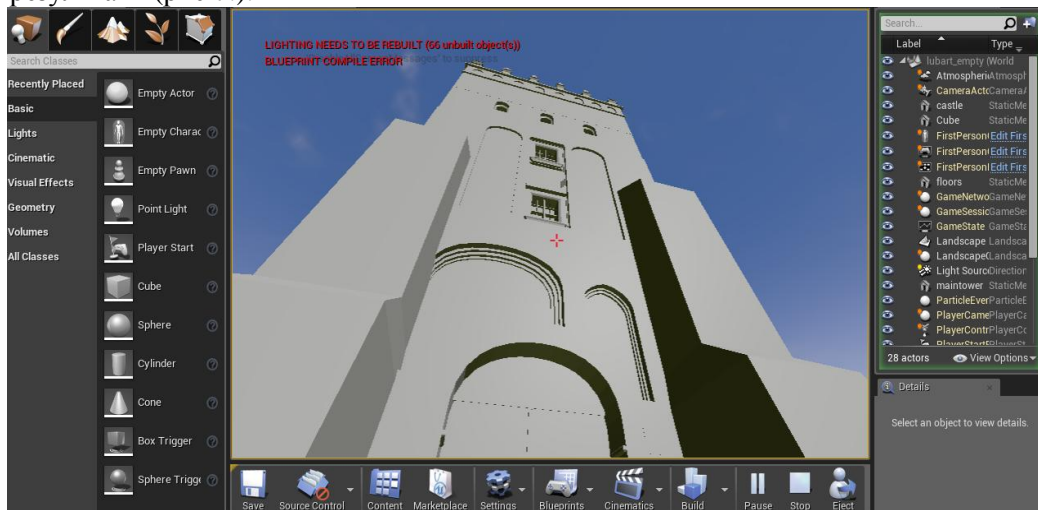


Рисунок 7. Модель в'їзної вежі в ігровому рушії

Разом з центральною вежею були імпортовані також її внутрішня складова та частина двору. Виконавши усі операції по перенесенню моделі з однієї програми в іншу можна побачити певний результат. Реалістичність даного проекту (рівень відповідності реаліям) залежить від того як старанно моделюються окремі частини, найменші дрібниці. Склавши разом все, що було зроблено можна зрозуміти, що певний рівень реалістичності – існує.

Нижче наведені рисунки, які дозволяють порівняти вежу у віртуальному та реальному виконанні:





Рисунок 8. Порівняння в'їзної вежі



Рисунок 9. Порівняння гвинтових сходів та передкімнатних коридорів

**Висновки та перспективи.** Таким чином одним із найдоступніших способів створення віртуального середовища з метою реконструкції історичних пам'яток є моделювання об'єктів з урахуванням географічного розташування, масштабу та інших факторів, подальше перенесення цих об'єктів у ігровий рушій та розміщення їх уже в середовищі даної програми. Запропоновано алгоритм процесу відтворення історичних пам'яток у віртуальному ігровому середовищі. Було проведено порівняння програмних засобів та запропоновано варіанти, виходячи з власного досвіду моделювання та міркувань.

Проект, що розглядався у даній статті не є завершеним, а отже потребує подальшого розроблення. Проте використання іншого програмного забезпечення не планується, робота продовжується згідно із заданим алгоритмом: об'єкти створюються у Maya та тестуються у рушії.

1. Інженерна та комп'ютерна графіка : підруч. для студ. ВНЗ / В. Є. Михайленко, В. В. Ванін, С. М. Ковальов. – 5-те вид. – К. : Каравела, 2010. – 360 с. – (Українська книга).
2. Дж. Лі, Б. Уер. Тривимірна графіка та анімація. — 2-е вид. — М. : [Вільямс](#), 2002. — 640 с.
3. Д. Херн, М. П. Бейкер. Комп'ютерна графіка й стандарт OpenGL. — 3-е вид. — М. : [Вільямс](#), 2005. — 1168 с.
4. Е. Енджел. Інтерактивна комп'ютерна графіка. Вступний курс на базі OpenGL. — 2-е вид. — М. : [Вільямс](#), 2001. — 592 с.
5. Г. Снук. 3D-ландшафти в реальному часі на C++ і DirectX 9. — 2-е вид. — М. : [Кудиц-прес](#), 2007. — 368 с. — ISBN 5-9579-0090-7.
6. В. П. Іванов, А. С. Батраков. Тривимірна комп'ютерна графіка / Під ред. Г. М. Поліщука. — М. : [Радіо та зв'язок](#), 1995. — 224 с. — [ISBN 5-256-01204-5](#).

УДК 004.65

Яцюк С.М., к.пед.н, доцент, Муляр В. П., к.пед.н, доцент  
Східноєвропейський національний університет імені Лесі Українки**ВИКОРИСТАННЯ МАКРОСІВ БАЗИ ДАНИХ ACCESS ПРИ ВИВЧЕННІ ІНФОРМАТИКИ**

**Яцюк С.М., Муляр В. П. Використання макросів бази даних Access при вивченні інформатики.** У статті розглянуто загальні відомості про макроси, їх типи та створення в базі даних Access 2010 і вище версій. Проаналізовано роботу в новому конструкторі макросів, описано застосування макросів даних, іменованих макросів, їх значення та використання.

**Ключові слова:** база даних, об'єкти бази даних, макрос, запит, панель керування, конструктор макросів, будівник виразів, макроси даних, іменовані макроси.

**Яцюк С.М., Муляр В. П. Использование макросов базы данных Access при изучении информатики.** В статье рассмотрены общие сведения о макросах, их типы и создание в базе данных Access 2010 и выше версий. Проанализирована работа в новом конструкторе макросов, описано применение макросов данных, именуемых макросов, их значение и использование.

**Ключевые слова:** база данных, объекты базы данных, макрос, запрос, панель управления, конструктор макросов, построитель выражений, макросы данных, именуемые макросы.

**Yatsyuk S.M., Mulyar V.P. Using Macros Access database in the study of science.** The article discusses general information about macros, their types and creating a database Access 2010 and later versions. The analysis work in the new Macro Designer, described the use of macro data, called macros, their meaning and use.

**Keywords:** database, database objects, macros, query the control panel, macro designer, builder expressions, macro data, called macros.

**Вступ.** З випуском Access 2010 і старших версій проектування і використання макросів збагатилося низкою нових можливостей. Вони включають покращений конструктор макросів, макроси даних, засновані на таблицях, і зміни в процесі створення виразів. Покращений конструктор макросів дозволяє створювати більш гнучкі, зрозумілі і зручні в супроводі макроси. Макроси тепер можна прив'язувати до таблиць, після чого кожен об'єкт, який створюється на основі відповідної таблиці, буде успадковувати пов'язаний макрос. Будівник виразів тепер підтримує технологію IntelliSense, що спрощує створення виразів.

**Постановка наукової проблеми.** Мета даної роботи – показати значення та переваги створення макросів у Microsoft Office Access 2010 і старших версій при вивченні дисципліни «Інформатика».

**Аналіз літературних джерел.**

У ряді відомих літературних джерел достатньо добре розглянуто використання системи управління базами даних (СУБД) «ACCESS» у цілому [1], для рішення облікових задач [4; 5; 6] та з маркетингу [7]. Окрім того, розкривається методологія формування балансу підприємства, його звітності про фінансові результати [5], застосування від повідних методик [3; 8]. Але, нажалю, мало уваги приділяється застосуванням макросів у професійній діяльності.

**Виклад основного матеріалу**

Макроси в програмі Access можна вважати спрощеною мовою програмування, яку можна використовувати для додавання функціональності до бази даних. Наприклад, можна вкласти макрос до кнопки форми, щоб запускати макрос у разі натискання цієї кнопки. **Макроси** містять дії, які виконують завдання, наприклад відкривають звіт, виконують запит або закривають базу даних.

Можна створити макрос для виконання певної низки дій, а також групу макросів для виконання пов'язаної низки дій [10].

У Microsoft Office Access 2010 – Українська версія макроси можуть міститися в об'єктах макросу (їх іноді називають ізольованими макросами) або бути вбудованими у властивості події форм, звітів або елементів керування. Вбудовані макроси стають частиною об'єкта або елемента керування, у який їх вбудовано. Об'єкти макросу відображаються в області переходів у розділі Макроси; вбудовані макроси не відображаються [9, ст.32-33].

У Access версії вище 2010 з'явився новий конструктор макросів, що спрощує створення складних макросів і дозволяє скоротити кількість помилок при кодуванні. Для цього в конструкторі застосовуються списки, що розкриваються, технологія IntelliSense, повторне використання існуючих макросів, перетягування, а також копіювання і вставка через буфер обміну.

Поліпшений користувальницький інтерфейс дозволяє створювати умовні блоки для формування логічної послідовності дій. Обмінюватися макросами з іншими користувачами можна по електронній пошті, через веб-сторінки, в блогах і групах новин.

В попередніх версіях Microsoft Access конструктор макросів складається з трьох стовпців (Рис. 1). Умовні оператори додавалися в стовпець **Умова**, макроси - в стовпець **Макрокоманда**, а зазначені параметри - в стовпець **Аргументи**.

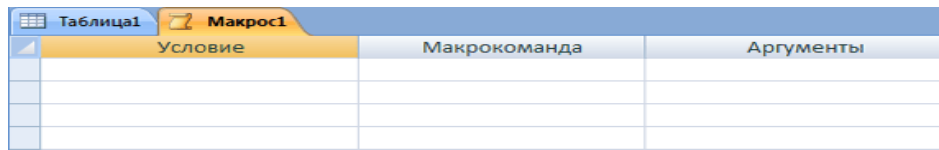


Рис 1. Минула версія конструктора макросів

Новий конструктор макросів для Access вище 2010-го більше схожий на текстовий редактор. Трьох стовпців більше немає. Замість них макроси і умовні оператори відображаються в списках, що розкриваються в звичному для програмістів форматі. Аргументи відображаються у вбудованому діалоговому вікні, як показано на рис. 2.

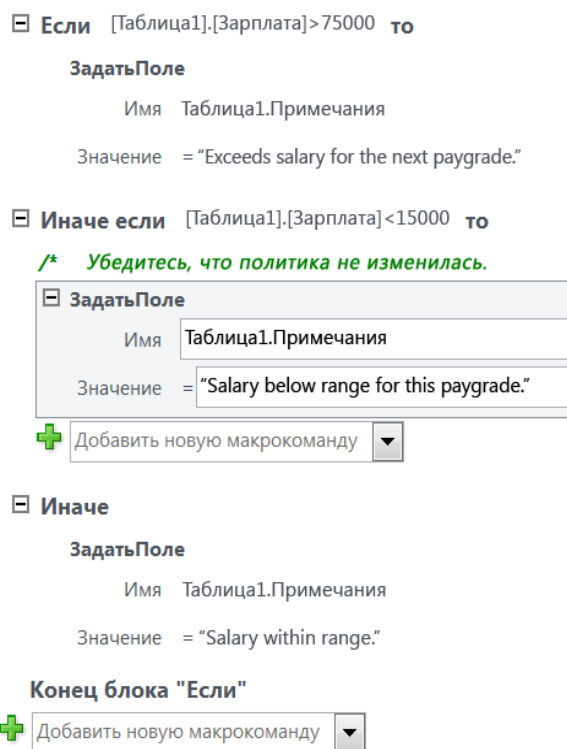


Рис. 2. Конструктор макросів

Додати нову макрокоманду або умовний оператор дуже просто. Макрокоманду можна вибрати в списку, що розкривається, з контекстного меню після клацання правою кнопкою миші або в області **Каталог макрокоманд** (Рис. 3), розташованої в правій частині конструктора макросів.

Щоб вивести всі макрокоманди, необхідно клацнути на панелі команду **Показати всі макрокоманди** (рис. 4).

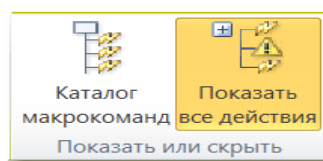


Рис. 4. Команда **Показати всі макрокоманди**

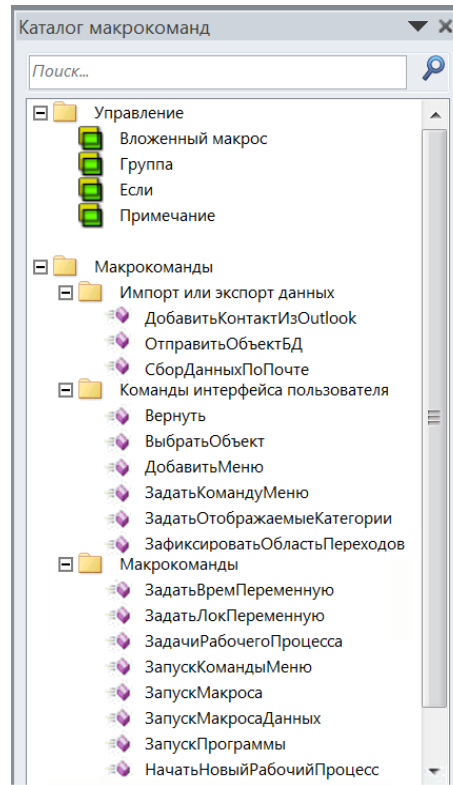


Рис. 3. Каталог макрокоманд

За допомогою елементів управління, розташованих зліва від блоків коду, можна згорнути деякі або навіть всі блоки коду. Для згортання і розгортання блоків коду також можна скористатися відповідними елементами управління в групі "Розгорнути / Згорнути" на панелі. На рис. 5 показаний макрос з рис. 2 в згорнутому вигляді.

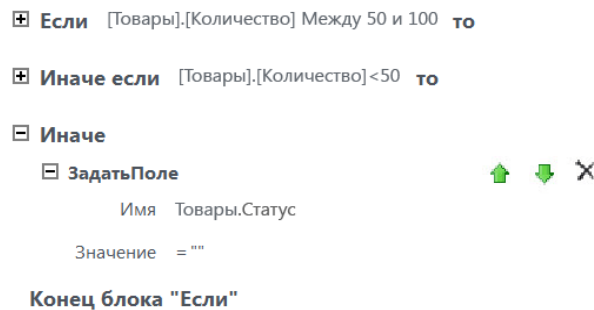


Рис. 5. Згорнуті блоки коду

**Каталог макрокоманд** підтримує можливість пошуку, що дозволяє ввести умови пошуку, застосувати необхідні фільтри Microsoft Access і вивести елементи, що задовольняють заданим умовам. При пошуку в **каталозі макрокоманд** враховуються не тільки їх імена, але й описи. Наприклад, якщо ввести в поле пошуку умову **Запит**, в результатах пошуку будуть виведені не тільки макрокоманди зі словом **Запит в імені**, а й макрокоманди **ApplyFilter**, **GoToRecord** і **ShowAllRecords**, в описі яких зустрічається слово *Запит*.

Щоб додати коментарі до макросу необхідно ввести // в поле зі списком **Додати нову макрокоманду** або перетягнути вузол **Коментарі** в **каталог макрокоманд**. Коментарі виділяються зеленим кольором, як показано на рис. 2. Це дозволяє легко знаходити коментарі і відокремлювати розділи процедури один від одного. Змінити порядок проходження коду в макросі також легко. Необхідно перетягнути блоки коду на нове місце або скористатися зеленими стрілками вгору і вниз (Рис. 5) для блоків в області макрокоманд.

У конструкторі макросів з'явилася нова конструкція для управління виконанням програми - блоки групування. Блоки групування дозволяють легко об'єднати декілька макросів в групу, яку можна розгортати і згорнути цілком для підвищення зручності читання.

У конструкторі макросів попередніх версій Microsoft Access в стовпці **Умова** можна було створювати прості умовні оператори. У конструкторі макросів Access 2010 і вище можна створювати більш універсальні оператори **If** шляхом додавання операторів **ElseIf** і **Else**. Щоб додати ці оператори, необхідно вибрати блок **If** і клацнути в правому нижньому кутку блоку коду тексту **ElseIf** або **Else**. Під час введення коду в поле умови в додатку Microsoft Access за допомогою технології IntelliSense будуть відображатися ідентифікатори, функції та інші елементи бази даних.

Конструктор макросів спільно з технологією IntelliSense також надає різні види довідки, які перераховані нижче.

Після введення користувачем достатньої кількості знаків для ідентифікації терміну функція автозаповнення виводить список слів, що відповідають об'єкту, функції або параметру. Користувач може або прийняти запропонований варіант, натиснувши клавішу **Enter** або **TAB**, або продовжити введення.

Функція "**Короткі відомості**" інформує про правильний синтаксис для кожного ідентифікатора в макросі або виразі. Функція "**Поради**" надає додаткові відомості про компоненти користувацького інтерфейсу при виборі значення за допомогою автозаповнення.

**Макроси даних** - це нова можливість в Access 2010 і вище версій. макроси даних дозволяють прив'язувати логіку до записів і таблиць (аналогічно триггерам в SQL Server). При цьому логіка пишеться один раз, а всі форми і код для додавання, оновлення та видалення даних в таблиці, успадковують цю логіку. Макроси даних дозволяють реалізовувати різні сценарії.

Перевірка значення поля, перед тим як дозволити додавання іншого запису

Збереження історії змін запису

Створення повідомлення електронної пошти при зміні значення поля

Перевірка точності даних в таблиці

Застарілі макроси (тут: макроси) як і раніше підтримуються, хоча викликати їх можна тільки з події форми, іншого макросу, події звіту або коду VBA.

Макроси даних бувають двох типів: макроси "подій", що спрацьовують при виконанні над даними в таблиці певної дії, і автономні "іменовані" макроси, які запускаються при їх виклику на ім'я. Макрос даних можна запрограмувати на запуск відразу ж після подій додавання, оновлення або вилучення даних або безпосередньо перед подією вилучення або зміни даних.

Макроси даних можуть повертати значення макросам через змінні ReturnVars. Змінні ReturnVars в макросах даних подібні значенням, які повертаються при виклику функції або методу в VBA та іншими мовами програмування. Це дозволяє виводити призначений для користувача інтерфейс у викликаному макросі в залежності від того, що сталося в макросі даних. У макросі даних змінні ReturnVars вказуються за допомогою команди SetReturnVar, як показано на рис. 6. Ці значення задаються в іменованих макросах даних.

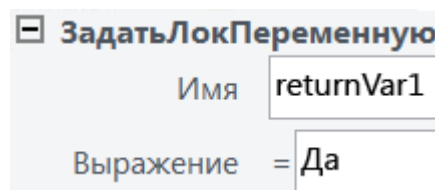


Рис. 6. SetReturnVar

Щоб посилатися на змінну в макросі, необхідно скористатися командою Return Vars, яканаведена в зазначеному нижче операторі.

VBA

```
=[ReturnVars]![retrunVar1]
```

Макроси даних дозволяють уникнути нагромадження бази даних завдяки відсутності необхідності пов'язувати один і той же макрос з декількома формами. При додаванні логіки в таблицю кожна форма, створена на основі таблиці, успадковує цю логіку. За допомогою макросів даних також можна забезпечити цілісність даних. Припустимо, що подія спрацьовує в формі, прив'язаною до таблиці без макросу даних. Якщо користувач має доступ до таблиць або може виконувати запити, він зможе обійти форму, порушивши таким чином логіку. Можна обмежити доступ до таблиць і заборонити виконання запитів, однак це можливо не завжди. При додаванні ж логіки безпосередньо в таблицю макрокоманда спрацьовує, навіть якщо користувач вносить зміни без використання форми.



Щоб створити макрос даних події, в області навігації зліва двічі клацніть ім'я таблиці, до якої необхідно прив'язати макрос. На вкладці **Таблиця** в групі **Події до** або **Події після** клацніть подію, в яку необхідно додати макрос. У Microsoft Access відкриється конструктор макросів. Якщо макрос для цієї події вже створений, конструктор макросів відкриється для цього макросу.

Щоб створити **іменованний макрос даних**, в області навігації зліва двічі клацніть ім'я таблиці, до якої необхідно прив'язати макрос. На вкладці **Таблиця** в групі **Іменовані макроси** натисніть кнопку **Іменованний макрос** і виберіть команду **Створити іменованний макрос**. У Microsoft Access відкриється конструктор макросів.

Розглянемо приклад. Припустимо, у нас є система обробки замовлень, що відслідковує кількість товарів на складі. Якщо кількість товару опускається нижче певного рівня, необхідно повідомити персонал про необхідність замовлення нової партії товару. Для цього в таблиці товарів ми створимо макрос даних. Таким чином, при створенні форми на основі цієї таблиці вся необхідна логіка вже буде готова.

Щоб створити макрос даних, створимо таблицю **Товари** з зазначеними нижче полями.

Таблиця 1. Зразок даних

Ім'я	Тип
Товар	Текст
Кількість	Число
Стан	Текст

Збережемо таблицю і додамо в стовпець Товар такі дані, залишивши інші стовпці порожніми.

Таблиця 2. Дані стовпця Товар

Coffe
Soda
Tea
Water

Щоб додати макрос, потрібно вибрати таблицю, в розділі Робота з таблицями на вкладці Поля натиснути розкриваючий список Уявлення і вибрати пункт Режим таблиці. У режимі таблиці відкрити в розділі **Робота з таблицями** вкладку **Таблиця** і натиснути кнопку події **До зміни**. Чому вибрано подія **Before Change**, а не **After Change**? Як правило, краще викликати подію **Before Change** для зміни даних і виконання додаткового коду, ніж дозволити таблиці оновити запис, а вже ПОТІМ внести додаткові зміни в той же запис, який тільки що був збережений в таблиці. Тригер події **After Change** краще використовувати для підведення підсумків і збереження підсумкових значень в іншій таблиці. Наприклад, можна перераховувати підсумкові значення по складських запасах при зміні кількості товару. У подібних випадках зміни, як правило, вносяться в іншу таблицю.

Далі необхідно створити оператор **If ... Then ... Else**, який буде спрацьовувати при кожному внесенні змін до таблиці. На вкладці Товари: до зміни натиснути розкриваючий список і вибрати оператор **If**. Щоб додати в оператор **If** умову, ввести після оператора **If** букву **T**. Далі вибрати таблицю Товари. Після ідентифікатора [**Товари**] ввести точку і вибрати поле **Кількість**. Після поля **Кількість** ввести від 50 до 100. При цьому буде задано умову, за якої якщо кількість товару виявиться в діапазоні від 50 до 100, спрацює гілка оператора **If**.

У правій частині екрана в каталозі макрокоманд виводиться список доступних макрокоманд і операторів управління виконанням програми для даної конкретної події. Виберіть макрокоманду **SetField** і перетягніть її в блок оператора **If**. У полі **Ім'я** введіть **Товари.Состояние**. У полі **Значення** ввести **"Брак товару"** (з лапками). Коли кількість товару виявиться в діапазоні від 50 до 100 одиниць, текст в полі **Стан** зміниться.

Клацнути оператор **If** ще раз і звернути увагу, що у правому нижньому кутку блоку з'явилися команди Додати **Else** і Додати **Else If**. Натиснути кнопку **Додати Else If**, щоб додати відповідний блок. В поле умови ввести **Товари.Количество <50**. Перетягнути з каталогу макрокоманд макрокоманду **SetField** в блок **Else If**. У полі **Ім'я** ввести **Товари.Состояние**, а в поле **Значення** - **"Замовити негайно!"** (з лапками). Кожен раз, коли товару залишиться менше 50 одиниць, текст в полі **Стан** буде змінюватися.

Клацнути оператор **Else If** і вибрати команду **Додати Else**. Перетягнути макрокоманду **SetField** в блок **Else**. У полі **Ім'я** ввести **Элементы.Состояние**, а в поле **Значення** - **""**, порожній рядок. Це вказує на те, що якщо кількість товару перевищує 100 одиниць, поле **Стан** повинно залишатися порожнім. Тепер макрос повинен бути подібний до наведеного на рис. 7.

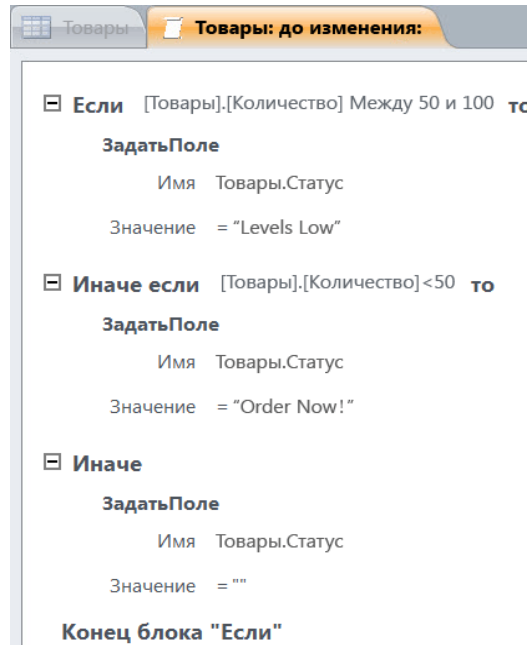


Рис 7. Макрос данных

Слід зберегти і закрити макрос, клацнувши вкладку **Товари:** для зміни правою кнопкою миші і вибравши команду **Закрити**. Щоб протестувати макрос, треба перейти в режим таблиці, змінити кількість товару "Кава" на 100, клацнути за межами записів і ви побачимо, що текст в полі Стан зміниться на Брак товару. Змініть кількість товару "Сода" на 49, і текст в поле Стан зміниться на Замовити негайно !. Змініть кількість товару "Чай" на 101. Поле Стан залишиться порожнім. Все це проілюстровано на рис. 8.

ИД	Товар	Количество	Статус
1	Coffee	100	Levels Low
2	Soda	49	Order Now!
3	Tea	101	
4	Water		

Рис. 8. Зміна поля Стан при зміні кількості товару

Ще одна можливість, підтримувана іменованими макросами даних - це можливість передачі параметрів. Параметри часто використовуються для підвищення продуктивності бази даних, оскільки вони дозволяють розробникам і користувачам примусово обмежити набори даних перед виконанням макросу. Це дозволяє підвищити швидкість роботи макросів, знизити навантаження на сервери баз даних і зменшити мережевий трафік. Параметри також забезпечують додаткову гнучкість, оскільки дозволяють повторно використовувати один і той же макрос без змін. Щоб додати параметри в іменованій макрос даних, необхідно двічі клацнути таблицю, до якої необхідно прив'язати макрос. На вкладці Таблица в групі Іменовані макроси натисніть розкриваючий список Іменованій макрос і вибрати команду Створити іменованій макрос. У верхній частині конструктора макросів (Рис. 9) натиснути кнопку **Створити параметр**.

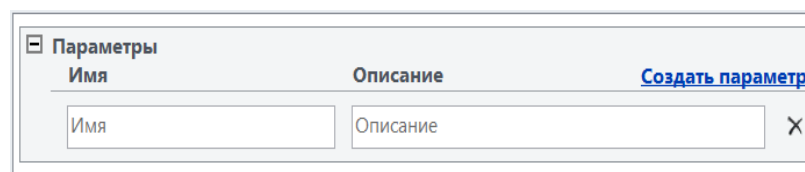


Рис. 9. Команда **Створити параметр** в конструкторі макросів

При налагодженні макросів даних є не так багато можливостей, як при налагодженні програмного коду. Покрокове виконання коду і використання вікна повідомлень не підтримується. Однак для пошуку

помилку в макросах даних можна скористатися таблицею "Журнал додатків" і макрокомандами **OnError**, **RaiseError** і **LogEvent**. Таблиця "Журнал додатків" є системною (**USysApplicationLog**) і за замовчуванням не відображається в області навігації. При виникненні помилки в макросі даних можна переглянути відомості в таблиці "Журнал додатків" і визначити, що сталося.

Щоб вивести таблицю "Журнал додатків" слід відкрити вкладку **Файл**, щоб переключитися у позицію **Backstage**. На вкладці **Відомості** натиснути кнопку Перегляд таблиці журналу додатків. Якщо кнопка **Перегляд таблиці журналу програми** не відображається, це означає, що таблиця "Журнал додатків" для поточної бази даних ще не створена. При запису в журнал якої-небудь події, наприклад за допомогою макрокоманд **LogEvent** або **RaiseError**, ця кнопка з'явиться на вкладці **Відомості**.

Будівник виразів отримав ряд нових можливостей і спрощень користувальницького інтерфейсу. Користувачам більше не потрібно запам'ятовувати синтаксис виразів і доступні функції або властивості. Новий будівник виразів підтримує технологію IntelliSense, яка надає всі необхідні відомості зі зворотним відліком вираження.

Крім того, новий призначений для користувача інтерфейс підтримує прогресивне розкриття. Відображення тільки функцій і властивостей, що існують в певному контексті. У попередніх версіях Microsoft Access для всіх контекстів, в яких використовувався вираз, застосовувався єдиний загальний обчислювач виразів. Це означає, що незалежно від того, де використовується вираз, доступні користувачеві функції і оператори будуть в більшості випадків одні і ті ж. У Access 2010 будівник виразів є контекстно-залежним. Наприклад, функція **FormatDateTime** недоступна в контексті таблиці, але доступна у всіх інших контекстах. Також слід звернути увагу, що кнопки операторів зникли. Оператори тепер доступні на панелі "Елементи виразів". Всі ці зміни дозволяють не захарашувати інтерфейс, надати користувачам більш осмислений вибір і збільшити область редагування.

У конструкторі макросів та інших місцях Access 2010 і старших версій, де необхідно створювати вирази, одним натисканням іконки, відкривається будівник виразів.

#### Висновки

Завдяки внесеним в новий конструктор макросів удосконаленням створення макросів стало простіше, ніж коли-небудь, як для новачків, так і для досвідчених розробників на Microsoft Access. Додавання конструкцій **Else** і **Else If** додало макросам ще більше гнучкості. А завдяки новому каталогу макрокоманд і підтримці технології IntelliSense тепер можна не запам'ятовувати часто використовувані команди і дії. Крім того, створення виразів в удосконаленому будівнику виразів стало більш зручним. В цілому можна сказати, що всі ці вдосконалення дозволяють ефективніше використовувати функції створюваних баз даних.

Напрямом подальшого дослідження є вдосконалення роботи з макросами Access для ефективного використання баз даних.

1. Довгань В. Access – Тольятти: Дока-Пресс, 2014. – 232 с.
2. Лобовко В. Нюансы работы с Access// Програмное обеспечение.– 2012.– № 7. – С. 12.
3. Мирончук Т. Вивчення робочого вікна Microsoft Access: Вісник Національного університету "Львівська техніка". – Л.: Вид-во НУ "Львівська техніка", – 2013. – № 527. – С. 44.
4. <http://www.mon.gov.ua/ua/activity/563/perelik-naukovikh-fakhovikh-vidan/6797/>
5. <http://sophus.at.ua>
6. <http://nauka.lp.edu.ua/index.php?id=8454>
7. <http://www-library.univer.kharkov.ua/ukr/>
8. [http://naps.gov.ua/ua/iccr/doc\\_list/](http://naps.gov.ua/ua/iccr/doc_list/)

[cit@lntu.edu.ua](mailto:cit@lntu.edu.ua).

УДК 621.317.7

Бурбело М. Й.<sup>1</sup>, Гадай А. В.<sup>2</sup>, к.т.н.

<sup>1</sup> - Вінницький національний технічний університет

<sup>2</sup> - Луцький національний технічний університет

## ВИЗНАЧЕННЯ ПОТУЖНОСТЕЙ НЕЛІНІЙНИХ НАВАНТАЖЕНЬ ТРИФАЗНИХ ЕЛЕКТРИЧНИХ МЕРЕЖ

**Бурбело М. Й., Гадай А. В. Визначення потужностей нелінійних навантажень трифазних електричних мереж.**

В статті проаналізовано можливість застосування класичної та миттєвої теорій потужностей для визначення потужностей нелінійних навантажень трифазних електричних мереж із заземленою нейтраллю. Показано, що найкращим є застосування класичної теорії потужностей. Введено поняття ефективних на періоді напруги живлення активної, реактивної та повної потужностей, а також їх середньоквадратичних відхилень. Показано, що останні є потужностями пульсацій, а ті, в свою чергу, є сумою потужностей небалансу та гармонічних спотворень. Запропоновано для визначення потужності спотворень використовувати середньоквадратичні значення різниці потужностей пульсацій та небалансу. Показано, що потужність спотворення, зумовлену наявністю канонічних гармонік, достатньо вимірювати на частоті шостої гармоніки напруги живлення з використанням миттєвих та комплексних потужностей.

**Ключові слова:** трифазні електричні мережі, нелінійні навантаження, визначення потужностей.

**Бурбело М. И., Гадай А. В. Определение мощности нелинейных нагрузок трехфазных электрических сетей. В**

статье проанализирована возможность применения классической и мгновенной теорий мощностей для определения мощностей нелинейных нагрузок трехфазных электрических сетей с заземленной нейтралью. Показано, что наилучшим является применение классической теории мощностей. Введено понятие эффективных на периоде напряжения питания активной, реактивной и полной мощностей, а также их среднеквадратичных отклонений. Показано, что последние являются мощностями пульсаций, а те, в свою очередь, являются суммой мощностей небаланса и гармонических искажений. Предложено для определения мощности искажений использовать среднеквадратичные значение разницы мощностей пульсаций и небаланса. Показано, что мощность искажения, обусловленную наличием канонических гармоник, достаточно измерять на частоте шестой гармоника напряжения питания с использованием мгновенных и комплексных мощностей.

**Ключевые слова:** трехфазные электрические сети, нелинейные нагрузки, определение мощностей.

**Burbelo1 M. J., Hadaĳ A. V. Determination of three-phase nonlinear loads electric.** The article possibility to application

of classic and instantaneous theories of capacity is analysed for determination of powers nonlinear loading of three-phase electric networks with the earthed neutral. It is shown that the best is to use the classic theory of capacity. The concept of effective voltage on the period of active, reactive and apparent power and their rms deviations. It is shown that the rms deviations are ripple capacity, and those, in turn, are the sum of unbalance powers and harmonic distortions. It is suggested for determination of distortions power to use the rms value of difference of ripple power and unbalance power. It is shown that distortion power, predefined by the presence of the canonical harmonics, it is enough to measure the frequency of sixth canonical voltage using instantaneous and complex power.

**Keywords:** three-phase electrical network, nonlinear load, power determination.

**Постановка наукової проблеми.** Робота трифазних електричних мереж з нелінійним навантаженням, призводить до значних спотворень синусоїдності струму і напруги. Ця проблема особливо стосується мереж обмеженої потужності з короткими лініями зв'язку. До таких мереж відносяться з напівпровідниковими перетворювачами. В таких умовах стає важливим питання правильного визначення активної, реактивної та повної потужностей.

**Аналіз досліджень.** Класичний підхід до визначення потужностей в синусоїдних режимах трифазних електричних мереж оснований на використанні комплексної повної потужності як скалярного добутку векторів трифазних комплексних напруг і спряжених струмів [1]

$$\underline{S} = P + jQ = \left( \left| \dot{\mathbf{U}} \cdot \dot{\mathbf{I}} \right| \right) = \left( \dot{U}_A \dot{I}_A^* + \dot{U}_B \dot{I}_B^* + \dot{U}_C \dot{I}_C^* \right), \quad (1)$$

де  $P, Q$  – активна та реактивна потужності;  $\dot{\mathbf{U}}, \dot{\mathbf{I}}$  – вектори комплексних напруг і спряжених струмів в фазних координатах, відповідно:  $\left[ \dot{U}_A, \dot{U}_B, \dot{U}_C \right]^T, \left[ \dot{I}_A, \dot{I}_B, \dot{I}_C \right]^T$ .

Реалізація вимірювальних перетворень полягає у формуванні миттєвих активної та реактивної потужностей як скалярного добутку векторів трифазних миттєвих напруг і струмів [2, 3]:

$$p = \left( \left| \mathbf{U} \cdot \mathbf{I} \right| \right) = (u_A i_A + u_B i_B + u_C i_C); \quad (2)$$

$$q = (|\mathbf{U}' \cdot \mathbf{I}|) = (u'_A i_A + u'_B i_B + u'_C i_C), \quad (3)$$

де  $\mathbf{U}, \mathbf{U}', \mathbf{I}$  – вектори напруг і струмів в фазних координатах  $[u_A, u_B, u_C]^T$ ,  $[u'_A, u'_B, u'_C]^T$ ,  $[i_A, i_B, i_C]^T$ , штрихом тут і нижче позначено фазовий зсув миттєвих величин на  $-90$  ел. градусів. За умов несинусоїдності фазовий зсув усіх гармонік напруги забезпечується за допомогою перетворення Гільберта. Однак реалізація такого перетворення є достатньо складною [4].

Активну та реактивну потужності вимірюють шляхом інтегрування відповідних миттєвих потужностей на періоді  $T$  напруги живлення

$$P = \frac{1}{T} \int_0^T p(t) dt; \quad Q = \frac{1}{T} \int_0^T q(t) dt. \quad (4)$$

За умов несинусоїдності класичний підхід відповідає теорії потужності К. Будяну, згідно з якою активна та реактивна потужності дорівнюють сумі, відповідно, активних та реактивних потужностей окремих гармонік:

$$P = \sum_{n=1}^{\infty} P_n; \quad Q = \sum_{n=1}^{\infty} Q_n, \quad (5)$$

а для виконання умови ортогональності вводиться поняття потужності спотворення  $D$  як третьої складової, що зумовлена гармонічними спотвореннями напруг і струмів

$$S^2 = P^2 + Q^2 + D^2. \quad (6)$$

Поряд з класичним визначенням реактивної потужності використовують поняття реактивної потужності за С. Фрізе, який запропонував визначати реактивну потужність в несинусоїдних режимах як ортогональну складову, прийняту для синусоїдних величин. Вимірювання реактивної потужності за С. Фрізе полягає у виконанні перетворень

$$Q_F = \sqrt{S^2 - P^2}; \quad S = U_A I_A + U_B I_B + U_C I_C, \quad (7)$$

де  $U_A, I_A; U_B, I_B; U_C, I_C$  – діючі значення напруг та струмів відповідних фаз, які за несинусоїдних напруг і струмів визначають за формулами, наприклад, для фази  $A$ :

$$U_A = \sqrt{\frac{1}{T} \int_0^T u_A^2 dt}, \quad I_A = \sqrt{\frac{1}{T} \int_0^T i_A^2 dt}. \quad (8)$$

Сучасний підхід до визначення потужностей полягає у використанні миттєвої повної потужності трифазної системи  $s(t)$  в якості базової величини, яку розкладають на ортогональні складники [7]. Вираз для квадрата миттєвої повної потужності можна подати у вигляді суми квадратів скалярного та векторного добутків векторів трифазних напруг і струмів в системі фазних координат

$$s^2 = p^2 + q_{\Sigma}^2 = (|\mathbf{I} \cdot \mathbf{U}|)^2 + (|\mathbf{I} \times \mathbf{U}|)^2, \quad (9)$$

де  $\mathbf{I}, \mathbf{U}$  – вектори струмів і напруг в системі фазних координат  $\mathbf{I} = [i_A, i_B, i_C]^T$ ,  $\mathbf{U} = [u_A, u_B, u_C]^T$ ;

У разі застосування системи фазних координат миттєву активну потужність визначають за формулою (2), а вираз для миттєвої реактивної потужності має такий вигляд:

$$q_{\Sigma} = (|\mathbf{I} \times \mathbf{U}|) = \sqrt{\begin{vmatrix} i_A & i_B \\ u_A & u_B \end{vmatrix}^2 + \begin{vmatrix} i_B & i_C \\ u_B & u_C \end{vmatrix}^2 + \begin{vmatrix} i_C & i_A \\ u_C & u_A \end{vmatrix}^2}. \quad (10)$$

Миттєву реактивну потужність  $q_{\Sigma}$  можна також визначити з умови ортогональності за виразом

$$q_{\Sigma} = \sqrt{s^2 - p^2} = \sqrt{u^2 \cdot i^2 - p^2}, \quad (11)$$

де  $u^2 = u_A^2 + u_B^2 + u_C^2$ ;  $i^2 = i_A^2 + i_B^2 + i_C^2$ .

Необхідно відмітити, що ортогональність складників миттєвої теорії потужностей справедлива лише для миттєвих значень і не виконується для усереднених величин. Якщо, наприклад, визначити

$$S = \frac{1}{T} \int_0^T s(t) dt; \quad P = \frac{1}{T} \int_0^T p(t) dt; \quad Q_{\Sigma} = \frac{1}{T} \int_0^T q_{\Sigma}(t) dt, \quad (12)$$

то умова  $S^2 = P^2 + Q_{\Sigma}^2$  не виконується.

Виконання умов ортогональності можна було б забезпечити у випадку формування ефективних інтегральних величин з використанням таких виразів

$$S_e = \sqrt{\frac{1}{T} \int_0^T s^2(t) dt}; \quad P_e = \sqrt{\frac{1}{T} \int_0^T p^2(t) dt}; \quad Q_{\Sigma e} = \sqrt{\frac{1}{T} \int_0^T q_{\Sigma}^2(t) dt}; \quad (13)$$

однак доцільність їх введення вимагає додаткових досліджень.

**Обґрунтування результатів дослідження.** Розглянемо несинусоїдний режим, за якого до мережі приєднаний шестипульсний випрямляч, який моделюється, за допомогою середовища Simulink 7.6 (R2010b), джерелом струму. Джерело містить першу, п'яту та сьому гармоніки, що становлять відповідно 20 % та 14,3 % від основної гармоніки, а напруги фаз крім першої гармоніки містять п'яту та сьому гармоніки, вміст кожної з яких становить 2 % від основної гармоніки (вказаний рівень напруг вищих гармонік відповідає відносному опору мережі, що визначається як відношення повних потужностей навантаження і короткого замикання, рівному 0,02).

Проаналізуємо спочатку симетричний по фазах режим навантажень. Потужність окремих гармонік становить:  $S_1 = 1500 e^{j30^\circ} = 1299 + j750 \text{ кВ}\cdot\text{А}$ ;  $\underline{S}_5 = -6 e^{j30^\circ} = -5,2 - j3 \text{ кВ}\cdot\text{А}$ ;  $\underline{S}_7 = -4,3 e^{j30^\circ} = -3,7 - j2,1 \text{ кВ}\cdot\text{А}$ . Результати вимірювань активної та реактивної потужностей відповідно будуть:  $P \approx 1290 \text{ кВт}$ ,  $Q \approx 744,9 \text{ квар}$ . Значення реактивної потужності за С. Фрізе  $Q_F \approx 850,9 \text{ квар}$ . Результати вимірювань ефективних повної, активної та реактивної потужностей відповідно будуть:  $S_e = 1545 \text{ кВ}\cdot\text{А}$ ,  $P_e \approx 1307 \text{ кВт}$ ,  $Q_e = Q_{\Sigma e} \approx 824,3 \text{ квар}$ .

Отже, в несинусоїдних симетричних режимах виміряні потужності, що відповідають класичній теорії потужностей, дають занижений результат, оскільки з мережі споживається активна потужність на основній гармоніці, що становить 1299 кВт, а генерується в мережу 5,2 кВт і 3,7 кВт, відповідно на п'ятій і сьомій гармоніках, спотворюючи форму напруги в мережі. При цьому споживач сплачує за сумарною активною потужністю на основній і вищих гармоніках, відповідно до показів лічильників електроенергії, тобто за 1290 кВт. Аналогічна ситуація спостерігається зі споживанням і оплатою за перетоки реактивної потужності.

Водночас, значення реактивної потужності за С. Фрізе та ефективних реактивних потужностей істотно завищують споживану реактивну потужність, оскільки включають в себе гармонічні спотворення як активної, так і реактивної потужностей.

В табл. 1. наведено результати розрахунків потужностей для шести режимів: 1) несинусоїдний симетричний режим (НСР), який було розглянуто вище; 2) несинусоїдний несиметричний режим (ННР), який відрізняється від розглянутого зниженням напруги та струму фази С на 10%; 3) синусоїдний несиметричний режим зі зниженням напруги та струму фази С на 10% (СНР); 4) несинусоїдний несиметричний режим з обривом фази С (ННРО); 5) несинусоїдний несиметричний режим з обривом фази С і 50% провалом напруги на фазі С (ННРОП); 6) синусоїдний несиметричний режим з обривом фази С (СНРО).

Таблиця 1. Результати розрахунків потужностей

Режим	$P_1$ , кВт	$Q_1$ , квар	$P$ , кВт	$Q$ , квар	$Q_F$ , квар	$P_e$ , кВт	$Q_e$ , квар	$Q_{\Sigma e}$ , квар
1) НСР	1299	750	1290	745	851	1307	824	824
2) ННР	1217	703	1208	698	797	1226	775	770
3) СНР	1217	703	1217	703	703	1219	706	701
4) ННРО	866	500	860	497	568	945	659	837
5) ННРОП	866	500	860	497	568	945	659	587
6) СНРО	866	500	866	500	501	936	613	791

Ефективні активна  $P_e$  та реактивна  $Q_e$  потужності завжди більші від потужностей на основній гармоніці, а також від усереднених на періоді потужностей. Водночас, ефективна реактивна потужність  $Q_{\Sigma e}$  в несиметричних режимах може бути як більшою, так і меншою від потужності на основній гармоніці, а також від усередненої на періоді потужності. Особливо неприємним є відмінність цієї потужності в режимах 4) і 5), які відрізняються значенням напруги на обірваній фазі. Фізична суть нестабільності потужності пояснюється використанням модуля трифазної напруги, який, наприклад, в режимі обриву фази не впливає на струм навантаження, проте його значення впливає на результат вимірювань потужності.

Проаналізуємо можливість визначення потужності гармонічних спотворень. Для цього пропонується використати поняття ефективних активної та реактивної потужностей, визначених на періоді напруги живлення:

$$P_e = \sqrt{\frac{1}{T} \int_0^T p^2(t) dt}; \quad Q_e = \sqrt{\frac{1}{T} \int_0^T q^2(t) dt}, \quad (14)$$

а також середньоквадратичні відхилення активної, реактивної та повної потужностей

$$\sigma_p = \sqrt{P_e^2 - P^2}; \quad \sigma_q = \sqrt{Q_e^2 - Q^2}; \quad \sigma_s = \sqrt{\sigma_p^2 + \sigma_q^2}. \quad (15)$$

Значення середньоквадратичних відхилень активної, реактивної та повної потужностей в режимі 1) будуть:  $\sigma_p = 210$  кВт;  $\sigma_q = 353$  квар;  $\sigma_s = 411$  кВ·А.

Потужності гармонічних спотворень  $s_p$  та  $s_q$  за відсутності несиметрії напруг та струмів можна записати через потужності пульсацій  $n_p$  та  $n_q$  у такому вигляді:

$$s_p = n_p = (p - P); \quad s_q = n_q = (q - Q). \quad (16)$$

Значення потужностей спотворень можна визначити шляхом інтегрування квадратів миттєвих потужностей пульсацій на періоді  $T$  напруги живлення

$$S_p = N_p = \sqrt{\frac{1}{T} \int_0^T n_p^2(t) dt}; \quad S_q = N_q = \sqrt{\frac{1}{T} \int_0^T n_q^2(t) dt}. \quad (17)$$

Значення  $S_p = 210$  кВт;  $S_q = 353$  квар;  $S_s = \sqrt{S_p^2 + S_q^2} = 411$  кВ·А.

На рис. 1 зображено залежності миттєвих величин  $p, q$  для режиму 1), з якого видно, що миттєві потужності мають рівномірний характер пульсацій, зумовлений лише несинусоїдністю і що частота коливань миттєвих потужностей в шість разів вища від частоти напруги мережі.

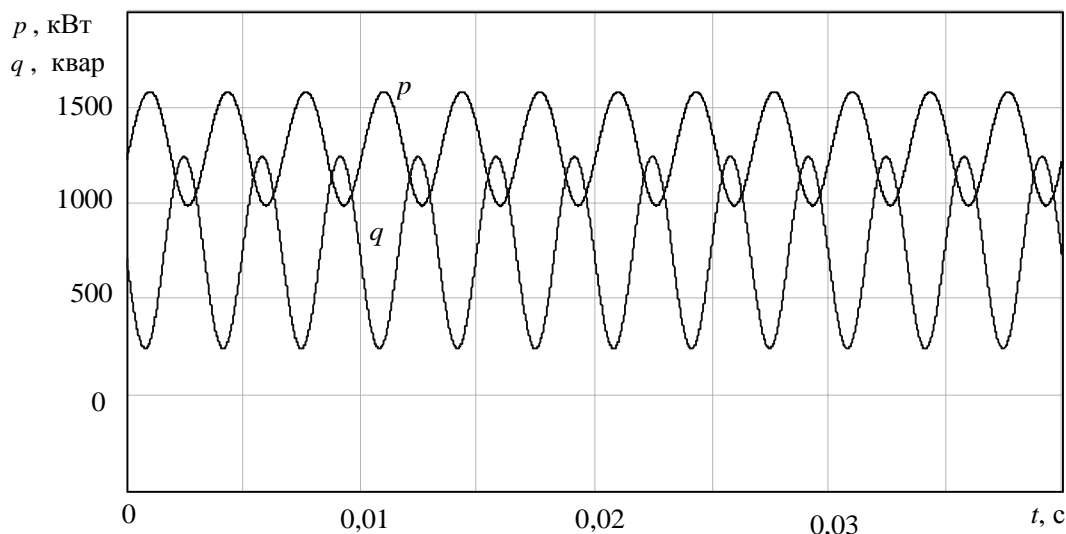


Рис. 1. Залежності миттєвих активної та реактивних потужностей за несинусоїдності

Потужності спотворень можна виміряти шляхом інтегрування відповідних миттєвих потужностей на періоді  $T$  напруги живлення

$$\operatorname{Re} \underline{S}_p = \frac{1}{\sqrt{2T}} \int_0^T s_p \sin 6\omega t dt; \quad \operatorname{Im} \underline{S}_p = \frac{1}{\sqrt{2T}} \int_0^T s_p \cos 6\omega t dt. \quad (18)$$

$$\operatorname{Re} \underline{S}_q = \frac{1}{\sqrt{2T}} \int_0^T s_q \sin 6\omega t dt; \quad \operatorname{Im} \underline{S}_q = \frac{1}{\sqrt{2T}} \int_0^T s_q \cos 6\omega t dt. \quad (19)$$

Комплексні значення потужностей спотворень, які визначені за миттєвими значеннями на частоті, що відповідає шостій гармоніці напруги, будуть:  $\underline{S}_p = 210e^{-j14,5^\circ}$  кВ·А,  $\underline{S}_q = 353e^{-j175^\circ}$  кВ·А.

Комплексні значення потужності спотворення можна визначити за векторним добутком векторів комплексних напруг і струмів окремих послідовностей, розмірність яких залежить від кількості врахованих гармонік. Шляхом комп'ютерного моделювання встановлено, що у разі врахування комбінаційних складників, що зумовлені першою, п'ятою та сьомою гармоніками прямої послідовності, комплексні значення потужності спотворення визначаються з таких виразів:

$$\underline{S}_{p,6} \approx \frac{3}{\sqrt{2}} \operatorname{Im} \left[ \left( \dot{U}_{11} \dot{I}_{15}^* - \dot{U}_{15} \dot{I}_{11}^* \right) + \left( \dot{U}_{11} \dot{I}_{17}^* - \dot{U}_{17} \dot{I}_{11}^* \right) \right] + j \frac{3}{\sqrt{2}} \operatorname{Re} \left[ - \left( \dot{U}_{11} \dot{I}_{15} + \dot{U}_{15} \dot{I}_{11} \right) + \left( \dot{U}_{11} \dot{I}_{17} + \dot{U}_{17} \dot{I}_{11} \right) \right]; \quad (20)$$

$$\underline{S}_{q,6} \approx - \frac{3}{\sqrt{2}} \operatorname{Re} \left[ \left( \dot{U}_{11} \dot{I}_{15} - \dot{U}_{15} \dot{I}_{11} \right) + \left( \dot{U}_{11} \dot{I}_{17} - \dot{U}_{17} \dot{I}_{11} \right) \right] + j \frac{3}{\sqrt{2}} \operatorname{Im} \left[ - \left( \dot{U}_{11} \dot{I}_{15} + \dot{U}_{15} \dot{I}_{11} \right) + \left( \dot{U}_{11} \dot{I}_{17} + \dot{U}_{17} \dot{I}_{11} \right) \right], \quad (21)$$

де перший індекс комплексних напруг і спряжених комплексних струмів вказує на використання комплексів напруг і струмів прямої послідовності, а другий – на номер гармоніки.

Розраховані за формулами (20) – (21) комплексні значення потужності спотворення відповідно будуть:  $\underline{S}_p = 210e^{-j14,5^\circ}$  кВ·А,  $\underline{S}_q = 353e^{-j175^\circ}$  кВ·А.

Отже, за усіх чотирьох способів оцінювання потужності гармонічних спотворень отримано однаковий результат.

Необхідно відмітити, що за відсутності несиметрії напруг і струмів потужність гармонічних спотворень можна визначити і через реактивні потужності:  $S_s = \sqrt{Q_F^2 - Q^2} = 411$  кВ·А.

За несиметрії напруг і струмів, наприклад, у разі провалу напруги фази С на 10 % від значення напруг інших фаз миттєві потужності  $p$  та  $q$  характеризуються змінним коливним процесом протягом пів періоду, що зумовлено несиметрією та несинусоїдністю. Обвідні цих потужностей характеризують пульсації, що зумовлені несиметрією режиму за зворотною послідовністю.

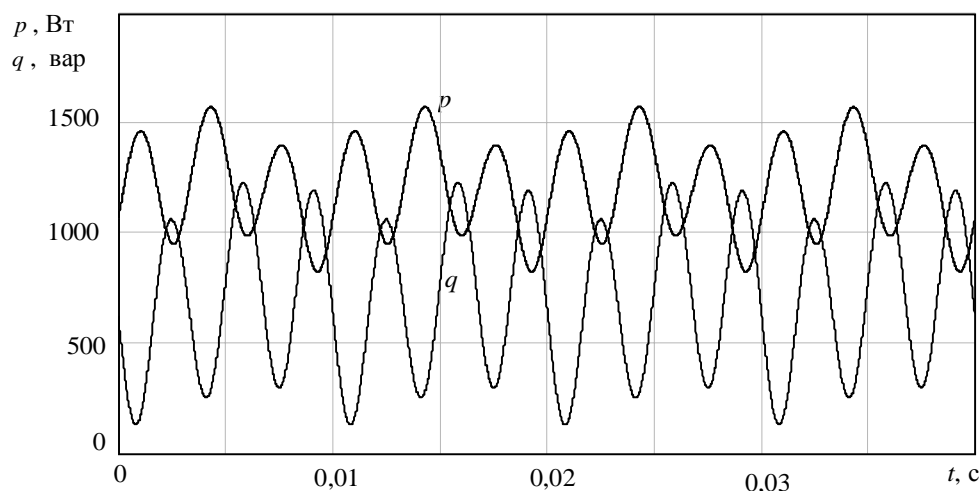




Рис. 2. Залежності миттєвих активної та реактивних потужностей за несиметрії та несинусоїдності

Складники потужності небалансу, що зумовлені несиметрією напруг і струмів, представимо миттєвими умовними потужностями зворотної послідовності [6, 7]:

$$p_2 = u_\alpha i_\alpha - u'_\beta i'_\beta; \quad q_2 = u'_\alpha i_\alpha + u_\beta i'_\beta, \quad (22)$$

де  $u_\alpha, u_\beta, i_\alpha, i_\beta$  – миттєві напруги та струми в системі  $\alpha\beta$ -координат; штрихом позначено фазовий зсув миттєвих величин на  $-90$  ел. градусів.

Несинусоїдний характер цих потужностей (рис. 3) свідчить про наявність комбінаційних складників, зумовлених взаємовпливом основної та вищих гармонік різних послідовностей.

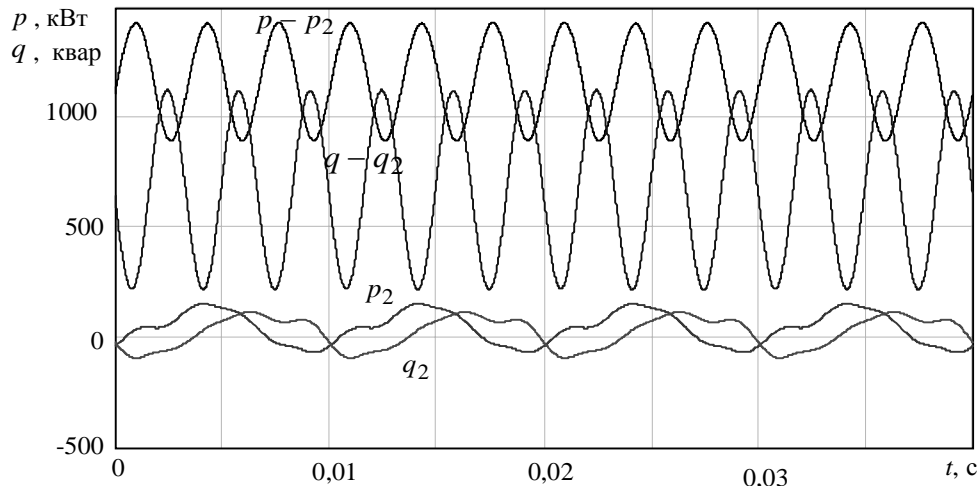


Рис. 3. Залежності миттєвих умовної потужності несиметрії та потужності спотворення

Умовні потужності зворотної послідовності визначають шляхом інтегрування відповідних миттєвих потужностей на періоді  $T$  напруги живлення

$$P_2 = \frac{1}{T} \int_0^T p_2(t) dt; \quad Q_2 = \frac{1}{T} \int_0^T q_2(t) dt. \quad (23)$$

Припустивши, що потужність пульсацій є сумою потужностей, що зумовлені несиметрією та несинусоїдністю

$$n_p = s_p + (p_2 - P_2); \quad n_q = s_q + (q_2 - Q_2), \quad (24)$$

миттєві потужності гармонічних спотворень доцільно подати у такому вигляді:

$$s_p = (p - P) - (p_2 - P_2); \quad s_q = (q - Q) - (q_2 - Q_2). \quad (25)$$

Значення середньоквадратичних відхилень активної, реактивної та повної потужностей будуть:  $\sigma_p = 208,4$  кВт;  $\sigma_q = 337,9$  квар;  $\sigma_s = 397,0$  кВ·А. Розраховані за формулами (17), (24) значення потужностей пульсацій будуть:  $N_p = 208,5$  кВт;  $N_q = 338,0$  квар;  $N_s = \sqrt{N_p^2 + N_q^2} = 397,2$  кВ·А. З розрахованих за формулами (17), (25) значення потужностей гармонічних спотворень будуть:  $S_p = 189,9$  кВт;  $S_q = 319,5$  квар;  $S_s = \sqrt{S_p^2 + S_q^2} = 371,7$  кВ·А. Комплексні значення потужностей спотворень, що визначені за миттєвими значеннями на частоті шостої гармоніки напруги, будуть:  $\underline{S}_p = 189,9e^{-j14,5^\circ}$  кВ·А,  $\underline{S}_q = 319,5e^{-j175^\circ}$  кВ·А.

Розраховані за формулами (20) – (21) комплексні значення потужності спотворення відповідно будуть:  $\underline{S}_p = 196,0e^{-j14,5^\circ}$  кВ·А,  $\underline{S}_q = 329,8e^{-j175^\circ}$  кВ·А.

Відмінність між визначеними за миттєвими значеннями та розрахованими за комплексними значеннями потужності спотворення пояснюється неврахуванням усіх складників цієї потужності, зокрема, нехтуванням напругами та струмами зворотної послідовності.

У разі врахування комбінаційних складників, що зумовлені основною гармонікою прямої послідовності та п'ятою і сьомою гармоніками зворотної послідовності, а також основною гармонікою зворотної послідовності та п'ятою і сьомою гармоніками прямої послідовності, вирази (20), (21) набудуть вигляду

$$\begin{aligned} \underline{S}_{p,6} \approx & \frac{3}{\sqrt{2}} \operatorname{Im} \left[ \left( \dot{U}_{11} \dot{I}_{15}^* - \dot{U}_{15} \dot{I}_{11}^* \right) + \left( \dot{U}_{11} \dot{I}_{17}^* - \dot{U}_{17} \dot{I}_{11}^* \right) - \left( \dot{U}_{11} \dot{I}_{25}^* - \dot{U}_{25} \dot{I}_{11}^* \right) - \right. \\ & \left. - \left( \dot{U}_{11} \dot{I}_{27}^* - \dot{U}_{27} \dot{I}_{11}^* \right) - \left( \dot{U}_{21} \dot{I}_{15}^* - \dot{U}_{15} \dot{I}_{21}^* \right) - \left( \dot{U}_{21} \dot{I}_{17}^* - \dot{U}_{17} \dot{I}_{21}^* \right) \right] + \\ & + j \frac{3}{\sqrt{2}} \operatorname{Re} \left[ - \left( \dot{U}_{11} \dot{I}_{15} + \dot{U}_{15} \dot{I}_{11} \right) + \left( \dot{U}_{11} \dot{I}_{17} + \dot{U}_{17} \dot{I}_{11} \right) + \left( \dot{U}_{11} \dot{I}_{25} + \dot{U}_{25} \dot{I}_{11} \right) - \right. \\ & \left. - \left( \dot{U}_{11} \dot{I}_{27} + \dot{U}_{27} \dot{I}_{11} \right) + \left( \dot{U}_{21} \dot{I}_{15} + \dot{U}_{15} \dot{I}_{21} \right) - \left( \dot{U}_{21} \dot{I}_{17} + \dot{U}_{17} \dot{I}_{21} \right) \right]; \end{aligned} \quad (26)$$

$$\begin{aligned} \underline{S}_{q,6} \approx & - \frac{3}{\sqrt{2}} \operatorname{Re} \left[ \left( \dot{U}_{11} \dot{I}_{15} - \dot{U}_{15} \dot{I}_{11} \right) + \left( \dot{U}_{11} \dot{I}_{17} - \dot{U}_{17} \dot{I}_{11} \right) - \left( \dot{U}_{11} \dot{I}_{25} - \dot{U}_{25} \dot{I}_{11} \right) - \right. \\ & \left. - \left( \dot{U}_{11} \dot{I}_{27} - \dot{U}_{27} \dot{I}_{11} \right) - \left( \dot{U}_{21} \dot{I}_{15} - \dot{U}_{15} \dot{I}_{21} \right) - \left( \dot{U}_{21} \dot{I}_{17} - \dot{U}_{17} \dot{I}_{21} \right) \right] + \\ & + j \frac{3}{\sqrt{2}} \operatorname{Im} \left[ - \left( \dot{U}_{11} \dot{I}_{15} + \dot{U}_{15} \dot{I}_{11} \right) + \left( \dot{U}_{11} \dot{I}_{17} + \dot{U}_{17} \dot{I}_{11} \right) + \left( \dot{U}_{11} \dot{I}_{25} + \dot{U}_{25} \dot{I}_{11} \right) - \right. \\ & \left. - \left( \dot{U}_{11} \dot{I}_{27} + \dot{U}_{27} \dot{I}_{11} \right) + \left( \dot{U}_{21} \dot{I}_{15} + \dot{U}_{15} \dot{I}_{21} \right) - \left( \dot{U}_{21} \dot{I}_{17} + \dot{U}_{17} \dot{I}_{21} \right) \right], \end{aligned} \quad (27)$$

а комплексні значення потужності спотворення відповідно будуть:  $\underline{S}_p = 189,2e^{-j14,5^\circ}$  кВ·А,  
 $\underline{S}_q = 318,5e^{-j175^\circ}$  кВ·А.

Зазначимо, що за наявності несиметрії напруг і струмів  $\sqrt{Q_F^2 - Q^2} = 385,4$  квар не несе інформації ні про потужність пульсацій, ні про потужність спотворень.

**Висновки.** Проаналізовано можливості визначення потужності спотворення нелінійних навантажень трифазних електричних мереж із заземленою нейтраллю з використанням класичної та миттєвої теорії потужностей. Показано, що найкращим є застосування класичної теорії потужностей. Введено поняття ефективних на періоді напруги живлення активної, реактивної та повної потужностей, а також середньквадратичних відхилень на періоді напруги живлення активної, реактивної та повної потужностей. Показано, що для визначення потужності спотворення можна використовувати середньквадратичні відхилення на періоді напруги живлення активної, реактивної та повної потужностей. Показано, що потужність спотворення, зумовлену наявністю канонічних гармонік, достатньо вимірювати на частоті шостої гармоніки напруги живлення.

1. Нейман Л. Р. Теоретические основы электротехники: Учебник для вузов / Л. Р. Нейман, К. С. Демирчян. – Л.: Энергоиздат. Ленингр. отд-ние, 1981. – Т.1. – 536 с.
2. Тонкаль И. Е. Баланс энергий в электрических цепях / И. Е. Тонкаль, А. В. Новосельцев, С. П. Денисюк [и др.]; – К.: Наукова Думка, 1992. – 312 с.
3. Мощность переменного тока / А. Ф. Крогерис, К. К. Рашевец, Э. П. Трейманис, Я. К. Шинка. – Рига: Физ.-энерг. инст. Латв.АН, 1993. – 294 с.
4. Штамбергер Г. А. Измерения в цепях переменного тока (методы уравнивания) / Г. А. Штамбергер; Под ред. К. Б. Карандеева. – Новосибирск: Наука, 1975. – 164 с.
5. Akagi H. Instantaneous power theory and applications to power conditioning / H. Akagi, E. H. Watanabe, M. Aredes. – IEEE Press / Wiley-Interscience, 2007. – 379 p.
6. Бурбело М. Й. Вимірювання параметрів несиметричних швидкозмінних трифазних навантажень / М. Й. Бурбело, С. М. Мельничук, М. В. Никитенко // Технічна електродинаміка. – 2011. – № 2. – С. 54–56.
7. Бурбело М. Й. Визначення потужностей за несиметричних режимів трифазних мереж із заземленою нейтраллю / М. Й. Бурбело, С. М. Мельничук // Технічна електродинаміка. – 2015. – № 4. – С. 71–75.

УДК 519.9

Гінайло П.І., к.ф.-м.н.

Луцький національний технічний університет

**ЗАДАЧА ОПТИМАЛЬНОГО КЕРУВАННЯ ДЛЯ ДИФЕРЕНЦІАЛЬНИХ ВКЛЮЧЕНЬ З МНОГОЗНАЧНИМИ ВІДОБРАЖЕННЯМИ**

**Гінайло П.І. Задача оптимального керування для диференціальних включень з многозначними відображеннями.** Отримані необхідні умови екстремуму для задачі оптимального керування, які описуються диференціальними багатозначними включеннями. При цьому враховується наявність додаткових параметрів, за допомогою яких можна впливати на функціонал і фазові обмеження задачі.

Ключові слова: екстремум, диференціальні включення, необхідні умови екстремуму, оптимальне керування, багатозначні відображення.

**Гинайло П.И. Задача оптимального управления для дифференциальных включений с многозначными отображениями.** Получены необходимые условия экстремума для задачи оптимального управления, описываемые дифференциальными включениями. При этом учитываются наличие дополнительных параметров, с помощью которых можно влиять на функционал и фазовые ограничения задачи.

Ключевые слова: экстремум, дифференциальные включения, необходимые условия экстремума, оптимальное управление, многозначные отображения.

**Ginaylo P. The problem of optimal control for differential inclusions with multivalued maps.** Necessary extremum conditions for optimal control problems described by differential inclusions meaningful are received. This takes into account the availability of additional parameters, which can affect the functional restrictions and phase problems.

**Keywords:** extremum, differential inclusion necessary extremum conditions, optimal control, ambiguous map.

**Постановка наукової проблеми.** Різноманітні задачі оптимального керування в останні роки знаходять широке застосування в самих різних областях сучасної науки і техніки. Усі фізичні процеси, що мають місце в техніці, як правило, керовані, тобто можуть здійснюватися різними способами, в залежності від потреб людини. Тому і виникає питання про знаходження найкращого або оптимального в тому чи іншому розумінні керуванні процесом. Побудові найбільш загальних необхідних умов екстремуму присвячено дуже багато робіт різних вчених. Це умови екстремуму дають можливість передбачити структуру розв'язку.

**Аналіз досліджень.** Новий напрямок досліджень в теорії необхідних умов екстремуму сприяв розвитку нового підходу до багатьох задач оптимального керування, в якому центральне місце займає поняття багатозначного відображення [1-3]. Справа в тому, що задачу оптимального керування виявляється зручно трактувати як задачу оптимізації на множині траєкторій деякого диференціального включення. Поняття диференціального включення дозволяє охопити багато задач оптимального керування єдиним методом розв'язання. В основі теорії диференціальних включень лежить поняття багатозначного відображення.

**Метою роботи** є побудова необхідних умов екстремуму для диференціальних включень при наявності додаткових обмежень на координати траєкторії в динамічних задачах з багатозначними відображеннями.

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

Будемо розглядати скінченні простори  $X = R^n$ ,  $Y = R^m$ . Нехай  $a$  – багатозначне відображення,  $a(x) \subset R^n$ .

Означення 1. Будь-яку абсолютно неперервну вектор-функцію  $x(t) \in R^n$  що задовольняє майже скрізь включенню

$$\dot{x}(t) \in a(x(t)), \quad t \in [0,1]$$

назвемо траєкторією.

Зафіксуємо стандартний проміжок зміни одномірного аргументу  $t \in [0,1]$ .

Нехай для  $t \in [0,1]$  визначена абсолютно неперервна функція  $x(t) \in R^n$ , майже скрізь диференційована. При цьому може статися, що для майже всіх  $t$  виконується відношення

$$\frac{d}{dt} x(t) \in a(x(t)) \quad (1)$$

яке називається диференціальним включенням, тобто це диференціальне рівняння з багатозначною правою частиною.

Розв'язком диференціального включення називається абсолютно неперервна вектор-функція  $x(t)$ , яка майже скрізь задовольняє співвідношення (1) на заданому інтервалі змінної  $t$ .

Якщо множина  $a(x)$  складається лише з одного елемента, функції  $f(x)$ , тобто відображення  $a(x)$  – однозначне, то дане включення переходить в звичайне диференціальне рівняння

$$\frac{d}{dt}x(t) = f(x).$$

Видно, що диференціальне включення є ніщо інше, як узагальнення диференціальних рівнянь.

До диференціальних включень приводять, наприклад, задача про функції, що задовольняють диференціальному рівнянню з заданою точністю

$$\left[ \frac{dx(t)}{dt} - f(t, x(t)) \right] \leq \varepsilon$$

диференціальні нерівності

$$f\left(t, x, \frac{dx}{dt}\right) \geq 0.$$

Якщо вектор-функція  $f(x, u) \in R^n$  визначена для  $x \in R^n$  і  $u \in U$ , де  $U$  – деяка підмножина з  $R^s$ , то багатозначне відображення  $a(x)$  можна визначити так

$$a(x) = f(x, u) = \{y : y = f(x, u), u \in U\}.$$

Тоді диференціальне включення (1) переходить у співвідношення

$$\frac{d}{dt}x(t) \in f(x(t), u),$$

або

$$\frac{d}{dt}x(t) \in f(x(t), u(t)), \quad u(t) \in U. \quad (2)$$

Такі рівняння (2) зазвичай зустрічаються в задачах керування. Тут  $x(t)$  шукана вектор-функція, а  $u = u(t)$  – керування, тобто вектор-функція, яку можна вибрати довільно серед всіх допустимих керувань.

Із вище сказаного слідує, що диференціальне включення (1) може мати множину розв'язків. В зв'язку з цим в задачах оптимального керування знаходиться оптимальний розв'язок, який задовольняє задані умови. В подальших дослідженнях при розв'язку цього питання будемо використовувати теорему 1.

Теорема 1. Якщо  $a(x)$  – випуклозначне замкнуте обмежене відображення, напівнеперервне зверху по  $x$ , і  $dom a$  складається з кулі радіуса

$$r = (1 + \|x_0\| e^c - 1)$$

де  $c$  – константа така, що

$$\|a(x)\| \leq c(1 + \|x\|)$$

то існує розв'язок  $x(t)$  диференціального включення (1), яке задовольняє нерівність

$$\|x(t)\| \leq (1 + \|x_0\| e^{ct} - 1), \quad x(0) = x_0.$$

При цьому розв'язок  $x(t)$  визначений на всьому відрізку  $[0, 1]$  і задовольняє на цьому відрізку умову Ліпшиця.

Означення 2. Якщо  $x(t) \in R^n$ ,  $t \in [0,1]$  – довільна крива, то її  $\varepsilon$  - околom називається множина точок  $x$  таких, що

$$\|x - x(t)\| \leq \varepsilon$$

при деякому  $t \in [0,1]$ .

Оскільки диференціальне включення навіть при заданій початковій точці визначає не єдину траєкторію, що йому задовольняє, то має зміст ставити задачу знаходження серед усіх траєкторій диференціального включення таку, яка б володіла додатковими властивостями.

Питання про існування розв'язку оптимізаційної задачі задаватись не буде. Будемо припускати, що оптимальний розв'язок цієї задачі існує і будемо позначати його через  $\tilde{x}(t)$ . Тоді наша ціль заключається в тому, щоб знайти ті умови, яким цей розв'язок задовольняє, тобто отримати необхідні умови екстремуму.

Сформулюємо таку задачу.

Серед усіх траєкторій  $x(t)$ ,  $t \in [0,1]$ , які задовольняють майже скрізь диференціальному включенню

$$\dot{x}(t) \in a(x(t)), \quad t \in [0,1]$$

і умовам

$$x(0) \in N, \quad x(1) \in M$$

знайти таку, яка мінімізує вираз

$$I(x(\cdot)) = \int_0^1 g(x(t), t) dt + \varphi_0(x(1)).$$

Ця задача розв'язується при наступних умовах на дані що входять до неї. Крайні умови цієї задачі задаються множинами  $N$  і  $M$  за допомогою кінцевої системи нерівностей і рівнянь

$$N = \{x: \varphi_b(x) \leq 0\}, \\ M = \{x: \varphi_l(x) \leq 0\}$$

де

$$\varphi_b(x) = \max \left\{ \max_{i \in I^-} \varphi_b^i(x), \max_{i \in I^0} (\varphi_b^i(x)) \right\}.$$

Означимо також для  $\delta \geq 0$  для множин:

$$N_\delta = \{x: \varphi_b(x) \leq c\delta\}, \\ M_\delta = \{x: \varphi_l(x) \leq c\delta\}$$

де  $\varphi_b(x)$ ,  $\varphi_l(x)$  – функції, які задовольняють умови Лібшиця.

В якості функцій  $\varphi_b(x)$  і  $\varphi_l(x)$  можуть також бути взяті функції відстані до множини  $d(x/N)$  і  $d(x/M)$ .

Припущення А. Множини  $N$  і  $M$  замкнуті. Множини  $N_\delta$  і  $M_\delta$ ,  $\delta \geq 0$ , в кожній точці володіють локальними шатрами  $K_{N_\delta}(x)$  і  $K_{M_\delta}(x)$ , причому якщо послідовності

$$x_k \in N_{\delta_k}, \quad x_k^* \in K_{\delta_k}^*(x_k) \\ x'_k \in M_{\delta_k}, \quad x'_k{}^* \in K_{\delta_k}^*(x'_k)$$

збігаються до  $x_0, x_0^*$  і до  $x_\delta$  та  $x_\delta^*$  відповідно, а  $\delta_k \rightarrow 0$ , то

$$x_0^* \in K_N^*(x_0), \\ x_\delta^* \in K_M^*(x_\delta).$$

Тут в якості  $K_{N_\delta}(x)$  взятий конус

$$K_{N_\delta}(x) = \{ \bar{x} : \bar{x} = \gamma(x_1 - x), \gamma > 0, x_1 \in N_\delta \}.$$

Спряжений йому конус матиме вигляд

$$K_{N_\delta}^*(x) = \{ x^* : \langle \bar{x}, x^* \rangle \geq 0, \bar{x} \in K_{N_\delta}(x) \}.$$

Аналогічно

$$K_{M_\delta}(x) = \{ \bar{x} : \bar{x} = \gamma(x_1 - x), \gamma > 0, x_1 \in M_\delta \}$$

$$K_{M_\delta}^*(x) = \{ x^* : \langle \bar{x}, x^* \rangle \geq 0, \bar{x} \in K_{M_\delta}(x) \}.$$

Припущення В. Функції  $g(x, t)$  і  $\varphi_0(x)$  неперервні по  $x$  і  $t$  та задовольняють умову Ліпшиця

$$|g(x_1, t) - g(x_2, t)| < L \|x_1 - x_2\|$$

$$|\varphi_0(x_1) - \varphi_0(x_2)| < L \|x_1 - x_2\|$$

в будь-якій обмеженій області простору  $X$ . Константа  $L$  може залежати від цієї області, але не залежить від  $t \in [0, 1]$ . При цьому  $g(x, t)$  та  $\varphi_0(x)$  допускають в кожній точці  $(x, t)$  верхню випуклу апроксимацію, а їх субдиференціали по  $x$   $\partial(g(x, t))$  і  $\partial(\varphi_0(x))$  рівномірно обмежені в кожній обмеженій області і напівнеперервно зверху залежать від  $x$  і  $t$ .

Припущення С. Відображення  $a$  випуклозначне і замкнуте, задовольняє умову Ліпшиця в  $\xi$  – трубці траєкторії  $\tilde{x}(t)$ ,  $t \in [0, 1]$ , і множина  $a(\tilde{x}(0))$  обмежена. Конуси  $K_a(x, y)$  являються локальними шатрами і локально спряжене відображення  $a^*(y^*; (x, y))$  напівнеперервне зверху залежить від своїх аргументів рівномірно обмежене для всіх  $x$  з  $\xi$  – трубки траєкторії  $\tilde{x}(\cdot)$  і  $y \in a(x)$ .

Для сформульованої задачі при зроблених припущеннях отримані необхідні умови мінімуму.

Теорема 2. Нехай виконані припущення А, В, С і  $\tilde{x}(\cdot)$  мінімізує функціонал

$$I(x(\cdot)) = \int_0^1 g(x(t), t) dt + \varphi_0(x(1))$$

серед всіх траєкторій, які задовольняють включення

$$\dot{x}(t) \in a(x(t)), t \in [0, 1]$$

і крайні умови  $x(0) \in N$ ,  $x(1) \in M$ .

Тоді існує число  $\lambda_0 \geq 0$ , вектор  $x_a^*$  і функція  $x^*(t)$ ,  $t \in [0, 1]$  не рівні нулю одночасно, такі що:

$$1) \quad x^*(1) + x_a^* \in \lambda_0 \partial \varphi(x(1)) \quad x_a^* \in K_M^*(x(1)), \quad x^*(0) \in K_N^*(x(0))$$

функція  $x^*(t)$  задовольняє умову Ліпшиця

$$x^*(t) \in A^*(x^*(t); \tilde{x}(t) + \lambda \partial g(\tilde{x}(t), t))$$

майже скрізь на відріжку  $[0, 1]$ ;

$$2) \quad \langle \tilde{x}(t), x^*(t) \rangle = W_a(\tilde{x}(t), x^*) \text{ майже скрізь на } [0, 1].$$

**Висновки.** Отримані необхідні умови екстремуму для задачі оптимального керування з диференціальними багатозначними включеннями. При цьому враховано наявність додаткових параметрів, за допомогою яких можна впливати на функціонал і фазові обмеження задачі.

## Список використаних джерел

1. Пшеничный Б.Н. Выпуклый анализ и экстремальные задачи. – М.: Наука, 1980. – 320 с.
2. Пшеничный Б.Н. Необходимые условия экстремума. – М.: Наука, 1982. – 144 с.
3. Гінайло П.І. Необхідні умови екстремуму для задачі оптимального керування з континуумом обмежень. Наукові нотатки. Міжвузівський збірник (за напрямком „Інженерна механіка”), вип..11, част.2, Луцьк, 2002, С.15–17.
4. Гінайло П.І. Необхідні умови екстремуму для локально випуклої задачі. // Дванадцята Міжнародна наукова конференція ім. акад. М. Кравчука. Київ, 15–17 трав. 2008 р. Матеріали конф., т. I. – К.: НТУУ “КПІ”. – 2008. – 572с.

УДК 519.9

Гінайло П.І., к.ф.-м.н.

Луцький національний технічний університет

## ПРО ЕКВІВАЛЕНТНІСТЬ ДВОХ ЗАДАЧ ОПТИМАЛЬНОГО КЕРУВАННЯ З ДИФЕРЕНЦІАЛЬНИМИ ВКЛЮЧЕННЯМИ

**Гінайло П.І. Про еквівалентність двох задач оптимального керування з диференціальними включеннями.** Розглядається задача оптимального керування з диференціальними включеннями. Наводиться метод зведення її до еквівалентної задачі, яка розв'язується за допомогою вже відомих методів.

**Ключові слова:** екстремум, диференціальні включення, необхідні умови екстремуму, оптимальне керування, багатозначні відображення.

**Гинайло П.И. Об эквивалентности двух задач оптимального управления с дифференциальными включениями.** Рассматривается задача оптимального управления с дифференциальными включениями. Приводится путь сведения ее к эквивалентной задаче, решаемой с помощью уже известных методов.

**Ключевые слова:** экстремум, дифференциальные включения, необходимые условия экстремума, оптимальное управление, многозначные отображения.

**Ginaylo P. On the equivalence of two optimal control problems with differential inclusions.** The problem of optimal control of differential inclusions is received. An method of reducing it to an equivalent problem that is solved by the known methods.

**Keywords:** extremum, differential inclusion necessary extremum conditions, optimal control, ambiguous map.

**Постановка наукової проблеми.** В основі досліджень лежать результати теорії необхідних умов екстремуму і багатозначних відображень. Різні задачі оптимального керування в останні роки знаходять широке застосування в самих різних областях сучасної науки і техніки. Усі фізичні процеси, що мають місце в техніці, як правило, керовані, тобто можуть здійснюватися різними способами, в залежності від потреб людини. Тому і виникає питання про знаходження найкращого або оптимального в тому чи іншому розумінні керуванні процесом. Побудові найбільш загальних необхідних умов екстремуму присвячено дуже багато робіт різних вчених. Це умови екстремуму дають можливість передбачити структуру розв'язку.

**Аналіз досліджень.** Новий напрямок досліджень в теорії необхідних умов екстремуму сприяв розвитку нового підходу до багатьох задач оптимального керування, в якому центральне місце займає поняття багатозначного відображення [1-5]. Справа в тому, що задачу оптимального керування виявляється зручно трактувати як задачу оптимізації на множині траєкторій деякого диференціального включення. Поняття диференціального включення дозволяє охопити багато задач оптимального керування єдиним методом розв'язання. В основі теорії диференціальних включень лежить поняття багатозначного відображення.

**Метою роботи** є розв'язання задачі оптимального керування з диференціальними включеннями шляхом зведення її до еквівалентної задачі.

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

Нехай  $f(x, y)$ ,  $x \in R^n$ ,  $y \in R^m$  функція Ліпшиця своїх аргументів,  $a$  – багатозначне відображення, яке задовольняє умову Ліпшиця,  $a(x)$  – випукла компактна множина в  $R^n$ . Нехай в  $R^n$  задані такі множини  $N$  і  $M$ , а в  $R^n \times R^m$  – замкнута множина  $Q$ .

Сформулюємо задачу оптимального керування (задача I): серед всіх абсолютно неперервних на відрізку  $[0, 1]$  функцій  $x(t)$  і вимірних функцій  $y(t)$ , які задовольняють умовам

$$x(0) \in N, \quad x(T) \in M$$

$$\dot{x}(t) \in a(x(t)) \text{ майже скрізь на } [0, 1] \quad (1)$$

$$(x(t), y(t)) \in Q \quad (2)$$

знайти такі, які мінімізують функціонал

$$I_1(x(\cdot), y(\cdot)) = \int_0^1 f(x(t), y(t)) dt \quad (3)$$

Подальше дослідження пов'язане з виведенням необхідних умов екстремуму для задачі I.

Нехай тепер



$$Q(x) = \{y : (x, y) \in Q\}$$

$$W(x) = \min_y \{f(x, y) : y \in Q(x)\}$$

$$Q_f(x) = \{y \in Q(x) : W(x) = f(x, y)\}.$$

Сформулюємо другу задачу оптимального керування (задача II): серед всіх абсолютно неперервних функцій  $x(t)$ , які задовольняють умовам

$$x(0) \in N, \quad x(1) \in M$$

$$\dot{x}(t) \in a(x(t)) \text{ майже скрізь на } [0, 1]$$

знайти таку, яка мінімізує функціонал

$$I_2(x(\cdot)) = \int_0^1 w(x(t)) dt.$$

Очевидно, що

$$I_2(x(\cdot)) \leq I_1(x(\cdot), y(\cdot)),$$

якщо виконані умови (2).

Надалі дослідження буде заключатися в тому, що замість задачі I будемо розв'язувати задачу II. Для цього покажемо, що при деяких умовах задачі I і II еквівалентні. Це означає, що якщо  $x(\cdot), y(\cdot)$  – розв'язок задачі I, то  $x(\cdot)$  – розв'язок задачі II, і навпаки, якщо  $x(\cdot)$  – розв'язок задачі II, то існує така вимірна функція  $y(\cdot)$ , яка задовольняє умову (2), що пара  $x(\cdot), y(\cdot)$  є розв'язком задачі I.

Доведемо наступну теорему.

Теорема 1. Нехай функція  $f$  випукла по  $y$ , а багатозначне відображення  $Q(x)$  неперервне, випукле і компактнозначне. Тоді задачі I і II еквівалентні.

Доведення базується на лемі, яка має окремий інтерес.

Лема. В умовах теореми 1. функція  $w$  неперервна, а множина  $Q_f(x)$  напівнеперервна зверху.

Доведення. В силу неперервності відображення  $Q(x)$  для будь-якого  $\xi > 0$  знайдеться таке  $\delta(\xi) > 0$  що

$$Q(x') \subseteq Q(x) + \xi B$$

$$Q(x) \subseteq Q(x') + \xi B \quad (4)$$

для всіх  $\|x' - x\| \leq \delta(\xi)$ . Тут  $B$  – одинична куля в  $R^m$ . Тепер

$$w(x') = \min_{y^*} \{f(x', y + u) : y \in Q(x), u \in \xi B\} =$$

$$= \min_{y, u} \{f(x, y) + [f(x', y + u) - f(x, y)] : y \in Q(x), u \in \xi B\} \geq$$

$$\geq \min_y \{f(x, y) : y \in Q(x)\} + \min_{y, u} \{f(x', y + u) - f(x, y) : y \in Q(x), u \in \xi B\} =$$

$$= v(x) + v(x, x', \xi),$$

де

$$v(x, x', \xi) = \min_{y, u} \{f(x', y + u) - f(x, y) : y \in Q(x), u \in \xi B\}.$$

При доведенні ми скористалися відношеннями (4) і тим, що мінімум суми більше суми мінімумів. Оскільки функція задовольняє умову Ліпшиця, то

$$|f(x', y') - f(x, y)| \leq L_1 \|x' - x\| + L_2 \|y' - y\|$$

тому

$$|f(x', y + u) - f(x, y)| \leq L_1 \|x' - x\| + L_2 \xi$$

так як

$$\|u\| \leq \xi$$

тоді

$$v(x, x', \xi) \geq -L_1 \|x' - x\| + L_2 \xi$$

і так, остаточно

$$w(x') - w(x) \geq -L_1 \|x' - x\| + L_2 \xi$$

аналогічно отримуємо, змінюючи  $x'$  і  $x$  місцями, що

$$w(x) - w(x') \geq -L_1 \|x' - x\| + L_2 \xi.$$

Таким чином

$$|w(x') - w(x)| \leq L_1 \|x' - x\| + L_2 \xi.$$

Звідси слідує, що якщо  $\|x' - x\| \leq \min(\delta(\xi), \xi)$  то

$$\|w(x') - w(x)\| \leq (L_1 + L_2) \xi$$

чим і доведена неперервність функції  $w(x)$ .

Нехай тепер  $U$  – відкрита множина, містить  $Q_f(x)$ . Покажемо, що якщо  $x'$  достатньо близько до  $x$ , то

$$Q_f(x') \subseteq U$$

що еквівалентно напівнеперервності зверху відображення  $Q_f(x)$ .

Припустимо супротивне. Тоді знайдеться така послідовність  $x_k \rightarrow x$  і такі точки  $y_k \in Q_f(x_k)$ , що  $y_k \notin U$  для всіх достатньо великих  $k$ . Але за означенням відображень  $Q_f(x)$  і  $Q(x)$

$$\begin{aligned} w(x_k) &= f(x_k, y_k) \\ (x_k, y_k) &\in Q. \end{aligned} \tag{5}$$

Так як

$$Q_f(x) \subseteq Q(x_k) \subseteq Q(x) + \xi B$$

для достатньо великих  $k$  при  $\xi > 0$  в силу неперервності відображення  $Q(x)$ , то послідовність  $y_k$  обмежена, і можна вважати, що  $y_k \rightarrow y$ . В силу замкнутості множини  $Q$  і того, що  $(x_k, y_k) \rightarrow (x, y)$  отримуємо, що  $(x, y) \in Q$ .

З доведеної вище неперервності функції переходячи до границі в (5) отримуємо, що

$$\begin{aligned} w(x) &= f(x, y) \\ (x, y) &\in Q \end{aligned}$$

тобто  $y \in Q_f(x)$ . Значить послідовність  $y_k$ , яка збігається до  $y \in Q_f(x)$ , не належить відкритій множині  $U$ , яка містить  $Q_f(x)$ , а значить і  $y$ . Це суперечить означенню збіжності.

Лема доведена.

Тепер легко закінчити доведення теореми.

Нехай  $x(\cdot)$  – розв'язок задачі II. Тоді, так як при зроблених припущеннях множина  $Q_f(x)$  випукла і напівнеперервна залежить від  $x$ , то з результатів роботи слідує, що існує така вимірна функція  $y(t)$ , що

$$y(t) \in Q_f(x(t)),$$

тобто

$$(x(t), y(t)) \in Q$$

$$f(x(t), y(t)) = w(x(t)).$$

Значить

$$I_1(x(t), y(t)) = I_2(x(t)).$$

Так як  $I_1 \geq I_2$  завжди і  $I_2(x(\cdot)) = \min I_2$ , то пара  $x(\cdot), y(\cdot)$  є розв'язком задачі I, і значення мінімумів в задачах I і II співпадають.

Якщо тепер  $(x(\cdot), y(\cdot))$  є розв'язком задачі I, то

$$I_1(x(\cdot), y(\cdot)) = \min I_1 = \min I_2.$$

Покажемо, що  $f(x(t), y(t)) = w(x(t))$  майже скрізь.

Дійсно, якщо це не так, тобто на множині скінченої міри

$$f(x(t), y(t)) > w(x(t))$$

то

$$\min I_1 = \min I_2 = \int_0^1 f(x(t), y(t)) dt > \int_0^1 w(x(t)) dt = I_2(x(\cdot)).$$

Тобто  $I_2(x(\cdot)) < \min I_2$ , чого не може бути.

Теорему повністю доведено.

*Наслідок.* Якщо множина  $Q$  випукла і замкнута, і хоча б при одному  $x$  множина  $Q(x)$  обмежена, то припущення теореми виконується.

Це слідує з результатів роботи [1], які відносяться до випуклих відображень.

Потрібно отримати необхідні умови екстремуму для задачі I. Але тепер замість задачі I дослідимо задачу II. Необхідні умови екстремуму для задачі II дає теорема 2. Щоб отримати необхідні умови екстремуму для задачі II, потрібно, щоб вона задовольняла припущенням наведеної теореми.

Оскільки  $Q$  – випукле відображення, функція  $f(x, y)$  випукла, то можна застосувати теорему 5 [1]. В силу цієї теореми отримаємо, що функція  $f(x, y)$  випукла, а відповідний їй диференціал має вигляд:

$$\partial w(x) = \{x^* + Q^*(y^*, (x, y)) : (x^*, y^*) \in \partial f(x, y)\}.$$

Але як відомо, субдиференціал неперервної випуклої функції рівномірно обмежений і неперервний зверху по включенню.

В лемі ми довели, що функція  $w(x)$  неперервна. А неперервна випукла функція задовольняє умову Ліпшиця

$$|w(x) - w(x_0)| \leq L \|x - x_0\|, \quad L = \text{const}.$$

Це можна легко показати. Нехай  $x_0 = 0$  і  $\Omega$  куля радіуса  $r$  з центром в  $x_0$ . Виберемо  $y \in \Omega$

таке що задовольняє нерівність  $\|y\| \leq \frac{r}{2}$ . Покладемо

$$x = \frac{r}{2} \frac{y}{\|y\|}.$$

Тепер використовуючи нерівність

$$|w(Lx) - w(0)| \leq 2c_1 L \quad \text{отримаємо}$$

$$|w(y) - w(0)| = \left| w\left(\frac{2\|y\|}{r} x\right) - w(0) \right| \leq L \|y\|.$$

Послідовно виконуються всі умови припущень В [1].

Тепер до поставленої задачі II можна застосувати теорему 2. Тим самим ми отримаємо необхідні умови мінімуму для задачі I. Сформулюємо основний результат.

Теорема 2. якщо виконані умови  $A$  і  $C$ , а  $\tilde{x}(t)$ ,  $t \in [0,1]$  мінімізує

$$\int_0^1 f(x(t), y(t)) dt.$$

серед усіх траєкторій диференціального включення

$$\dot{x}(t) \in a(x(t)),$$

що задовольняють умовам  $x(0) \in N$ ,  $x(1) \in M$  і  $(x(t), y(t)) \in Q$ ,

тоді існує абсолютно неперервна функція  $p^*(t)$ ,  $t \in [0,1]$ , вимірні функції  $u^*(t)$  і  $v^*(t)$ , число  $\lambda \geq 0$ , не всі одночасно рівні нулю, такі що:

- 1)  $-p^*(1) \in K_M^*(\tilde{x}(1))$ ,  $p^*(0) \in K_N^*(\tilde{x}(0))$
- 2) функція  $p^*(t)$  задовольняє умові Ліпшиця і
 
$$-p^*(t) \in \lambda [u^*(t) + Q^*(v^*(t), x(t), y(t))] + A^*(p^*(t); \tilde{x}(t))$$

майже скрізь на  $[0,1]$

- 3)  $(u^*(t), v^*(t)) \in \partial f(x(t), y(t))$ ,  $\langle \dot{\tilde{x}}(t), p^*(t) \rangle = W_a(\tilde{x}(t), p^*(t))$

майже скрізь на  $[0,1]$ .

Тут  $A^*$  і  $Q^*$  спряжені відображення до  $a$  і  $Q$ .  $W_a$  опорна функція відображення  $a$  визначається за формулою V.3.1 [1].

Якщо функція  $f$  диференційована, тоді маємо:

Теорема 3. Нехай виконані припущення попередньої теореми. Тоді необхідні умови оптимальності траєкторії  $\tilde{x}(t)$ ,  $t \in [0,1]$  будуть такі умови:

- 1)  $p^*(0) \in K_N^*(\tilde{x}(0))$ ,  $-p^*(1) \in K_M^*(\tilde{x}(1))$
- 2) функція  $p^*(t)$  задовольняє умові Ліпшиця і
 
$$-p^*(t) \in \lambda [f'_x(\tilde{x}(t), \tilde{y}(t)) + Q^*(f'_x(\tilde{x}(t), \tilde{y}(t)); (x(t), y(t)))] + A^*(p^*(t); \tilde{x}(t))$$

майже скрізь на  $[0,1]$

- 3)  $\langle \dot{\tilde{x}}(t), p^*(t) \rangle = W_a(\tilde{x}(t), p^*(t))$

майже скрізь на  $[0,1]$ .

*Доведення.* Якщо функція  $f(x, y)$  диференційовна, то її субдиференціал  $\partial f(x, y)$  складається лише з одного вектора-градієнта функції  $f(x, y)$ , який складається з компонентів, відповідним частинним похідним:

$$f'_x(x, y) \text{ і } f'_y(x, y).$$

Підставивши їх в твердження 2 теореми 2. отримаємо необхідне.

**Висновки:** Розглянута задача оптимального керування з диференціальними включеннями. Наведено метод зведення цієї задачі до еквівалентної, яка розв'язується відомими методами.

1. Пшеничний Б.Н. Выпуклый анализ и экстремальные задачи. – М.: Наука, 1980. – 320 с.
2. Пшеничний Б.Н. Необходимые условия экстремума. – М.: Наука, 1982. – 144 с.
3. Гінайло П.І. Про еквівалентність двох задач оптимального керування. // Вісник Київського університету. Математика і механіка. 1986. Вип. 28. С.24–27.
4. Гінайло П.І. Необхідні умови екстремуму для задачі оптимального керування з континуумом обмежень. // Наукові нотатки. Міжвузівський збірник (за напрямком „Інженерна механіка”), вип.11, част.2, Луцьк, 2002, С.15–17.
5. Гінайло П.І. Необхідні умови екстремуму для локально випуклої задачі. // Дванадцята Міжнародна наукова конференція ім. акад. М. Кравчука. Київ, 15–17 трав. 2008 р. Матеріали конф., т. I. – К.: НТУУ “КПІ”. – 2008. – 572с.

УДК 539.3

Гуда О.В., Тимошук В.М., Крадінова Т.А.  
Луцький національний технічний університет

## МЕТОД ВИВЕДЕННЯ РІВНЯНЬ РІВНОВАГИ В НЕКЛАСИЧНІЙ ПОСТАНОВЦІ

Гуда О.В., Тимошук В. М., Крадінова Т. А. **Метод виведення рівнянь рівноваги в неklasичній постановці.** У даній роботі на основі варіаційного принципу Лагранжа для повної енергії пружної системи виведено рівняння рівноваги. Дані рівняння враховують деформації поперечного зсуву та обтиснення.

**Ключові слова:** ізотропні та трансропні пластини, поперечний зсув, поперечне обтиснення, згинальні моменти, поперечні сили, прогин, напруження.

**Форм. 12. Літ. 6.**

Гуда О.В., Тимошук В.М., Крадінова Т.А. **Метод вывода уравнений равновесия в неклассической постановке.** В данной работе с помощью вариационного принципа Лагранжа для полной энергии упругости получены уравнения равновесия, которые учитывают деформации поперечного сдвига и обжатия.

**Ключевые слова:** изотропные и трансропные пластины, поперечный сдвиг, поперечное обжатие, сгибальные моменты, поперечные силы, изгиб, напряжение.

Guda O.V., Tymoshchuk V.M., Kradinova T.A. **The method of withdrawal equilibrium equations in the nonclassical setting.** In this paper, which based on the Lagrange variational principle for the complete power of elastic systems, the equations of equilibrium were derived. These equations take into account the deformations of transverse shift and compression.

**Key words:** isotropic and transtropic plates, transverse shear, transverse compression, bending moments, transverse forces, deflection, stress,.

**Постановка проблеми.** Існують різні методи виводу диференціальних рівнянь рівноваги тонких пластин. Одним із основних і найпоширеніших методів виводу таких рівнянь є використання рівнянь рівноваги в моментах і зусиллях. Недоліком цього методу є невмотивованість запису граничних умов на краях пластини. Більш досконалими є варіаційні методи (Лагранжа, Рейсснера, змішаного методу), які дозволяють, разом із рівняннями рівноваги, виводити енергетично вмотивовані граничні умови. У багатьох працях вітчизняних та зарубіжних дослідників використовують розрахункові рівняння пластин та оболонок у неklasичній постановці. Більшість існуючих неklasичних теорій пластин і оболонок враховують деформацію поперечного зсуву, а деякі, частково, враховують ще і поперечне обтиснення. Проте, як показують дослідження, у розрахунках за дії контактних та локалізованих навантажень, слід урахувувати поперечне обтиснення якнайповніше, що дозволяє задовольняти більшості умов на поверхні

**Аналіз останніх досліджень та публікацій.** В основу неklasичних (уточнених) моделей напружено-деформованого стану пластин середньої товщини в більшості випадків їх авторами покладаються кінетичні гіпотези для складових вектора переміщень, де тангенціальні складові переміщень змінюються лінійно (klasична теорія тонких пластин, теорії типу Тимошенка, Е. Рейсснера) відносно поперечної координати або за законом кубічної параболи (теорії С.О. Амбарцумяна, В.З. Власова, Х.М. Муштарі, В.Г. Піскунова, О.О. Расказова, О.Ф. Рябова, Р. Крістенсена та ін.). Разом з тим, вплив поперечних деформацій у цих моделях (за виключенням В.Г. Піскунова) авторами враховувався частково. Пізніше вплив поперечного обтиснення почали враховувати в задачах про контактну взаємодію жорстких штампів із пластинками та оболонками [5]. Впливу поперечного обтиснення на вищі частоти коливань пластин і оболонок присвячено значно менше робіт. У багатьох випадках такі дослідження проводились у постановках просторової задачі теорії пружності.

**Метою дослідження** є виведення рівнянь руху трансропних пластин середньої товщини, які враховують як ефекти поперечного зсуву, так і деформацію поперечного обтиснення, поперечне нормальне напруження та інерцію обертання поперечних перерізів.

**Основні результати дослідження.** Для виведення рівнянь руху та граничних умов у круглій плиті використано варіаційний принцип Лагранжа для повної енергії пружної системи [2, 4]

$$\delta \Pi = \delta A, \quad (1)$$

де

$$\delta\Pi = \iiint_{V_p} (\sigma_r \cdot \delta\varepsilon_r + \sigma_\theta \cdot \delta\varepsilon_\theta + \sigma_z \cdot \delta\varepsilon_z + \tau_{r\theta} \cdot \delta\gamma_{r\theta} + \tau_{rz} \cdot \delta\gamma_{rz} + \tau_{\theta z} \cdot \delta\gamma_{\theta z}) dV_p \quad - \quad \text{варіація}$$

потенціальної енергії деформації;  $dV_p = r dr d\theta dz$  – елемент об'єму плити;

$$\delta A = \iiint_{V_p} (F_r \delta U + F_\theta \delta V + F_z \delta W) dV_p + \iint_S (q^- \delta W^- - q^+ \delta W^+) dS \quad - \quad \text{варіація роботи об'ємних та}$$

поверхневих сил;  $dS = r d\theta dr$  – елемент поверхні плити;  $W^\pm$  – компоненти вектора пружного переміщення на зовнішніх поверхнях  $z = \pm h$  плити;  $F_r = -\rho \frac{\partial^2 U}{\partial t^2}$ ;  $F_\theta = -\rho \frac{\partial^2 V}{\partial t^2}$ ;  $F_z = -\rho \frac{\partial^2 W}{\partial t^2}$  – проекції сил інерції на відповідні координатні осі, віднесені до одиниці об'єму, які формально виконують роль об'ємних сил;  $\rho$  – густина.

Використовуючи формули Коші для компонент деформації

$$\varepsilon_r = \frac{\partial U}{\partial r}, \quad \varepsilon_\theta = \frac{U}{r} + \frac{1}{r} \frac{\partial V}{\partial \theta}, \quad \varepsilon_z = \frac{\partial W}{\partial z},$$

$$\gamma_{r\theta} = \frac{1}{r} \frac{\partial U}{\partial \theta} + \frac{\partial V}{\partial r} - \frac{V}{r}, \quad \gamma_{rz} = \frac{\partial U}{\partial z} + \frac{\partial W}{\partial r}, \quad \gamma_{z\theta} = \frac{\partial V}{\partial z} + \frac{1}{r} \frac{\partial W}{\partial \theta},$$

та здійснюючи варіювання з урахуванням формул інтегрування частинами і співвідношень типу

$$\delta\left(\frac{\partial U}{\partial r}\right) = \frac{\partial}{\partial r}(\delta U), \quad \text{варіація потенціальної енергії буде}$$

$$\begin{aligned} \delta\Pi = & \iiint_{V_p} \left[ \frac{\partial}{\partial r}(\sigma_r \delta U) + \frac{\partial}{\partial \theta} \left( \frac{\sigma_\theta}{r} \delta V \right) + \frac{\partial}{\partial z}(\sigma_z \delta W) + \frac{\partial}{\partial \theta} \left( \frac{\tau_{r\theta}}{r} \delta U \right) + \frac{\partial}{\partial r}(\tau_{r\theta} \delta V) + \right. \\ & \left. + \frac{\partial}{\partial z}(\tau_{rz} \delta U) + \frac{\partial}{\partial r}(\tau_{rz} \delta W) + \frac{\partial}{\partial z}(\tau_{z\theta} \delta V) + \frac{\partial}{\partial \theta} \left( \frac{\tau_{\theta z}}{r} \delta W \right) \right] dV_p - \\ & - \iiint_{V_p} \left( \frac{\tau_{r\theta}}{r} \delta V - \frac{\sigma_\theta}{r} \delta U + \frac{\partial \sigma_r}{\partial r} \delta U + \frac{1}{r} \frac{\partial \sigma_\theta}{\partial \theta} \delta V + \frac{\partial \sigma_z}{\partial z} \delta W + \frac{1}{r} \frac{\partial \tau_{r\theta}}{\partial \theta} \delta U + \right. \\ & \left. + \frac{\partial \tau_{r\theta}}{\partial r} \delta V + \frac{\partial \tau_{rz}}{\partial z} \delta U + \frac{\partial \tau_{rz}}{\partial r} \delta W + \frac{\partial \tau_{\theta z}}{\partial z} \delta V + \frac{1}{r} \frac{\partial \tau_{\theta z}}{\partial \theta} \delta W \right) dV_p = \\ & = \iint_{S_r} r \sigma_r \delta U d\theta dz + \iint_{S_\theta} \sigma_\theta \delta V dr dz + \iint_{S_z} r \sigma_z \delta W d\theta dr + \\ & + \iint_{S_\theta} \tau_{r\theta} \delta U dr dz + \iint_{S_r} r \tau_{r\theta} \delta V d\theta dz + \iint_{S_z} r \tau_{rz} \delta U dr d\theta + \iint_{S_r} r \tau_{rz} \delta W d\theta dz + \iint_{S_z} r \tau_{z\theta} \delta V dr d\theta + \\ & + \iint_{S_\theta} \tau_{z\theta} \delta W dr dz - \iiint_V \left( \frac{\sigma_r}{r} \delta U + \frac{\tau_{r\theta}}{r} \delta V + \frac{\tau_{rz}}{r} \delta W + \frac{\tau_{r\theta}}{r} \delta V - \right. \\ & - \frac{\sigma_\theta}{r} \delta U + \frac{\partial \sigma_r}{\partial r} \delta U + \frac{1}{r} \frac{\partial \sigma_\theta}{\partial \theta} \delta V + \frac{\partial \sigma_z}{\partial z} \delta W + \frac{1}{r} \frac{\partial \tau_{r\theta}}{\partial \theta} \delta U + \frac{\partial \tau_{r\theta}}{\partial r} \delta V + \\ & \left. + \frac{\partial \tau_{rz}}{\partial z} \delta U + \frac{\partial \tau_{rz}}{\partial r} \delta W + \frac{\partial \tau_{\theta z}}{\partial z} \delta V + \frac{1}{r} \frac{\partial \tau_{\theta z}}{\partial \theta} \delta W \right) dV_p. \end{aligned} \quad (2)$$

Варіація потенціалу зовнішніх сил

$$\delta A = -\rho \iiint_{V_p} \left( \frac{\partial^2 U}{\partial t^2} \delta U + \frac{\partial^2 V}{\partial t^2} \delta V + \frac{\partial^2 W}{\partial t^2} \delta W \right) dV_p + \iint_S q_2 \delta \tilde{W} ds. \quad (3)$$

Якщо підставити рівності (2) і (3) у варіаційне рівняння (1) та прирівняти до нуля в потрібному інтегралі вирази біля незалежних варіацій  $\delta U$ ,  $\delta V$ ,  $\delta W$ , то отримаємо диференціальні рівняння руху елемента об'єму пластини в циліндричних координатах:

$$\begin{aligned} \frac{\partial \sigma_r}{\partial r} + \frac{1}{r} \frac{\partial \tau_{r\theta}}{\partial \theta} + \frac{\partial \tau_{rz}}{\partial z} + \frac{\sigma_r - \sigma_\theta}{r} &= \rho \frac{\partial^2 U}{\partial t^2}, \\ \frac{\partial \tau_{r\theta}}{\partial r} + \frac{1}{r} \frac{\partial \sigma_\theta}{\partial \theta} + \frac{\partial \tau_{\theta z}}{\partial z} + \frac{2\tau_{r\theta}}{r} &= \rho \frac{\partial^2 V}{\partial t^2}, \\ \frac{\partial \tau_{rz}}{\partial r} + \frac{1}{r} \frac{\partial \tau_{\theta z}}{\partial \theta} + \frac{\partial \sigma_z}{\partial z} + \frac{\tau_{rz}}{r} &= \rho \frac{\partial^2 W}{\partial t^2}. \end{aligned} \quad (4)$$

Щоб отримати рівняння рівноваги через зусилля і моменти, а також граничні умови на краях пластинки, використано представлення напружень через внутрішні зусилля та моменти:

$$\begin{aligned} \sigma_r &= \frac{N_r}{2h} + \frac{M_r z}{I} + \frac{(1-\alpha)\tilde{E}}{G'} f_0(z) \left( \frac{\partial Q_r}{\partial r} + \frac{v}{r} \frac{\partial Q_\theta}{\partial \theta} + \frac{v}{r} Q_r \right) + \\ &+ \frac{(1-\alpha)\tilde{E}h^2}{2E'} f_0(z) \left( \frac{\partial^2 q_2}{\partial r^2} + \frac{v}{r^2} \frac{\partial^2 q_2}{\partial \theta^2} + \frac{v}{r} \frac{\partial q_2}{\partial r} \right) + \\ &+ A'(q_1 + q_2(f_0(z) - 1)) + zA'\rho \frac{\partial^2 W}{\partial t^2}; \\ \sigma_\theta &= \frac{N_\theta}{2h} + \frac{M_\theta z}{I} + \frac{(1-\alpha)\tilde{E}}{G'} f_0(z) \left( v \frac{\partial Q_r}{\partial r} + \frac{1}{r} \frac{\partial Q_\theta}{\partial \theta} + \frac{1}{r} Q_r \right) + \\ &+ \frac{(1-\alpha)\tilde{E}h^2}{2E'} f_0(z) \left( v \frac{\partial^2 q_2}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2 q_2}{\partial \theta^2} + \frac{1}{r} \frac{\partial q_2}{\partial r} \right) + \\ &+ A'(q_1 + q_2(f_0(z) - 1)) + zA'\rho \frac{\partial^2 W}{\partial t^2}; \\ \tau_{r\theta} &= \frac{N_{r\theta}}{2h} + \frac{H_{r\theta} z}{I} + \frac{(1-\alpha)\tilde{E}}{G'} f_0(z) \left( \frac{\partial Q_\theta}{\partial r} + \frac{1}{r} \frac{\partial Q_r}{\partial \theta} + \frac{1}{r} Q_\theta \right) + \\ &+ \frac{(1-\alpha)\tilde{E}h^2}{2E'} f_0(z) \frac{1}{r} \frac{\partial}{\partial r} \left( \frac{\partial q_2}{\partial \theta} + \frac{\partial q_2}{\partial r} \right); \\ \tau_{rz} &= \frac{G'}{K'} \left( 1 - \frac{z^2}{h^2} \right) Q_r; \quad \tau_{\theta z} = \frac{G'}{K'} \left( 1 - \frac{z^2}{h^2} \right) Q_\theta; \end{aligned} \quad (5)$$

де  $f_0(z) = \frac{z}{4h^3} (0,6h^2 - z^2)$ .

Запишемо вирази для компонент пружних переміщень, у наступному вигляді [6]:

$$\begin{aligned} U(r, \theta, z) &= u + \gamma_r z + \psi_r \left( \frac{z}{5} - \frac{z^3}{3h^2} \right); \quad V(r, \theta, z) = v + \gamma_\theta z + \psi_\theta \left( \frac{z}{5} - \frac{z^3}{3h^2} \right); \\ W(r, \theta, z) &= w + \frac{2\alpha_0}{E'} q_1 z + \frac{3\alpha_0}{4hE'} \tilde{q}_2 z^2 \left( 1 - \frac{z^2}{6h^2} \right) - A' \left( z\theta_0 + \frac{z^2\theta_1}{2} - \frac{z^4\theta_3}{4h^2} \right) = \\ &= w + \frac{2\alpha_0}{E'} q_1 z + \frac{3\alpha_0}{4hE'} \tilde{q}_2 z^2 \left( 1 - \frac{z^2}{6h^2} \right) - A' z\theta_0 + \frac{1}{2} A' z^2 \theta_3 \left( 3h^2 + \frac{z^2}{2h^2} \right) + \frac{1}{2} A' z^2 \Delta w, \end{aligned} \quad (6)$$

де  $\gamma_r, \gamma_\theta$  – узагальнені кути повороту;  $\psi_r, \psi_\theta$  – функції поперечного зсуву.

Використовуючи формули (5), формули для деформацій та вирази для компонент пружних переміщень (6), знайдено вираз для варіації потенціальної енергії:

$$\begin{aligned} \delta\Pi &= \iint_S \int_{-h}^h (\sigma_r \cdot \delta\varepsilon_r + \sigma_\theta \cdot \delta\varepsilon_\theta + \sigma_z \cdot \delta\varepsilon_z + \tau_{r\theta} \cdot \delta\gamma_{r\theta} + \tau_{rz} \cdot \delta\gamma_{rz} + \tau_{\theta z} \cdot \delta\gamma_{\theta z}) dz dS = \\ &= \iint_S \left( N_r \delta \left( \frac{\partial u}{\partial r} \right) + M_r \delta \left( \frac{\partial \gamma_r}{\partial r} \right) + \frac{N_\theta}{r} \delta u + \frac{M_\theta}{r} \delta \gamma_r + N_\theta \delta \left( \frac{\partial v}{r \partial \theta} \right) + M_\theta \delta \left( \frac{\partial \gamma_\theta}{r \partial \theta} \right) + \right. \\ &+ N_{r\theta} \delta \left( \frac{\partial u}{r \partial \theta} \right) + H_{r\theta} \delta \left( \frac{\partial \gamma_r}{r \partial \theta} \right) + N_{r\theta} \delta \left( \frac{\partial v}{\partial r} \right) + H_{r\theta} \delta \left( \frac{\partial \gamma_\theta}{\partial r} \right) - \frac{N_{r\theta}}{r} \delta v - \frac{H_{r\theta}}{r} \delta \gamma_\theta + \\ &\left. + Q_r \delta \gamma_r + Q_r \delta \left( \frac{\partial \tilde{w}}{\partial r} \right) + Q_\theta \delta \gamma_\theta + Q_\theta \delta \left( \frac{\partial \tilde{w}}{r \partial \theta} \right) \right) dS. \end{aligned}$$

Використовуючи формули інтегрування частинами та варіювання за незалежними змінними  $u, v, \tilde{w}, \gamma_r, \gamma_\theta$ , отримуємо для  $\delta\Pi$  та  $\delta A$ :

$$\begin{aligned} \delta\Pi &= - \iint_S \left[ \left( \frac{\partial N_r}{\partial r} + \frac{1}{r} \frac{\partial N_{r\theta}}{\partial \theta} + \frac{N_r - N_\theta}{r} \right) \delta u + \left( \frac{\partial N_{r\theta}}{\partial r} + \frac{1}{r} \frac{\partial N_\theta}{\partial \theta} + \frac{2N_{r\theta}}{r} \right) \delta v + \right. \\ &+ \left( \frac{\partial Q_r}{\partial r} + \frac{1}{r} \frac{\partial Q_\theta}{\partial \theta} + \frac{Q_r}{r} + q_2 \right) \delta \tilde{w} + \left( \frac{\partial M_r}{\partial r} + \frac{1}{r} \frac{\partial H_{r\theta}}{\partial \theta} + \frac{M_r - M_\theta}{r} - Q_r \right) \delta \gamma_r + \\ &\left. + \left( \frac{\partial H_{r\theta}}{\partial r} + \frac{1}{r} \frac{\partial M_\theta}{\partial \theta} + \frac{2}{r} H_{r\theta} - Q_\theta \right) \delta \gamma_\theta \right] dS + \int_L \left[ (N_r l + N_{r\theta} m) \delta u + \right. \\ &\left. + (N_{r\theta} l + N_\theta m) \delta v + (Q_r l + Q_\theta m) \delta \tilde{w} + (M_r l + H_{r\theta} m) \delta \gamma_r + (H_{r\theta} l + M_\theta m) \delta \gamma_\theta \right] dL. \end{aligned} \quad (7)$$

$$\delta A = -2\rho h \iint_S \left( \frac{\partial^2 u}{\partial t^2} \delta u + \frac{\partial^2 v}{\partial t^2} \delta v + \frac{h^2}{3} \frac{\partial^2 \gamma_r}{\partial t^2} \delta \gamma_r + \frac{h^2}{3} \frac{\partial^2 \gamma_\theta}{\partial t^2} \delta \gamma_\theta \right) dS + \iint_S \tilde{q}_2 \delta \tilde{w} dS.$$

Тут  $L$  – межа контуру області  $S$ ,  $l$  та  $m$  – напрямні косинуси нормалі до контуру пластини.

Прирівнюючи вирази біля незалежних варіацій в (7), отримано систему рівнянь руху через внутрішні сили та моменти:

$$\begin{aligned} \frac{\partial N_r}{\partial r} + \frac{1}{r} \frac{\partial N_{r\theta}}{\partial \theta} + \frac{N_r - N_\theta}{r} &= 2\rho h \frac{\partial^2 u}{\partial t^2}, \\ \frac{\partial N_{r\theta}}{\partial r} + \frac{1}{r} \frac{\partial N_\theta}{\partial \theta} + \frac{2N_{r\theta}}{r} &= 2\rho h \frac{\partial^2 v}{\partial t^2}, \\ \frac{\partial M_r}{\partial r} + \frac{1}{r} \frac{\partial H_{r\theta}}{\partial \theta} + \frac{M_r - M_\theta}{r} - Q_r &= \frac{2}{3} \rho h^3 \frac{\partial^2 \gamma_r}{\partial t^2}, \\ \frac{\partial H_{r\theta}}{\partial r} + \frac{1}{r} \frac{\partial M_\theta}{\partial \theta} + \frac{2}{r} H_{r\theta} - Q_\theta &= \frac{2}{3} \rho h^3 \frac{\partial^2 \gamma_\theta}{\partial t^2}, \\ \frac{\partial Q_r}{\partial r} + \frac{1}{r} \frac{\partial Q_\theta}{\partial \theta} + \frac{Q_r}{r} &= -q_2 + 2\rho h \frac{\partial^2 \tilde{w}}{\partial t^2}, \end{aligned} \quad (8)$$

де  $\tilde{w} = w + \frac{1}{6} A' \Delta w h^2 + \frac{9\alpha_0 h A_2 q_2}{40E'}$ ;  $\{N_r, N_\theta, N_{r\theta}, Q_r, Q_\theta\} = \int_{-h}^h \{\sigma_r, \sigma_\theta, \tau_{r\theta}, \tau_{rz}, \tau_{\theta z}\} dz$ ;

$\{M_r, M_\theta, H_{r\theta}\} = \int_{-h}^h \{\sigma_r, \sigma_\theta, \tau_{r\theta}\} z dz$ ;  $Q_r = K' \cdot \psi_r$ ;  $Q_\theta = K' \cdot \psi_\theta$ ;  $q^+, q^-$  – навантаження на зовнішніх

поверхнях пластини ( $z = \pm h$ ), що направлені вниз, у напрямку осі  $Oz$ ;  $G'$  – модуль поперечного зсуву матеріалу пластинки;  $\psi_r, \psi_\theta$  – деформації поперечного зсуву серединної поверхні пластинки.

Рівняння (8) можна отримати з системи (4), якщо, наслідуючи С.О.Амбарцумяна [1], помножити всі рівняння системи (4) на  $dz$ , а перших два ще і на  $z dz$  та проінтегрувати їх в межах



від  $-h$  до  $h$ . Разом із тим, така методика не дозволяє отримати енергетично коректні граничні умови на краю пластинки.

Граничні умови отримуємо з контурного інтегралу, що входить у рівняння (7):

$$\begin{aligned} (N_r l + N_{r\theta} m) \delta u = 0; \quad (N_{r\theta} l + N_\theta m) \delta v = 0; \quad (Q_r l + Q_\theta m) \delta w = 0; \\ (M_r l + H_{r\theta} m) \delta \gamma_r = 0; \quad (H_{r\theta} l + M_\theta m) \delta \gamma_\theta = 0. \end{aligned} \quad (9)$$

З системи рівнянь (9) можна отримати статичні або геометричні граничні умови, в залежності від того який множник прирівняти до нуля.

Підставивши в рівняння (4) замість внутрішніх сил та моментів їх вирази, з врахуванням попередніх зауважень, отримаємо рівняння руху через переміщення  $u, v, w_\tau$  та кути повороту  $\gamma_r, \gamma_\theta$ :

$$\begin{aligned} \Delta u + \frac{1+v}{r} \frac{\partial}{\partial \theta} \left( \frac{\partial v}{\partial r} - \frac{1}{r} \frac{\partial u}{\partial \theta} + \frac{v}{r} \right) - \frac{1}{r^2} \left( u + 2 \frac{\partial v}{\partial \theta} \right) &= -\frac{v''(1+v)}{E} \frac{\partial q_1}{\partial r} + 2\rho h \frac{\partial^2 u}{\partial t^2}; \\ \Delta v + \frac{1+v}{1-v} \frac{1}{r} \frac{\partial}{\partial \theta} \left( \frac{\partial u}{\partial r} + \frac{1}{r} \frac{\partial v}{\partial \theta} + \frac{u}{r} \right) - \frac{1}{r^2} \left( v - 2 \frac{\partial u}{\partial \theta} \right) &= -\frac{2v''(1+v)}{E(1-v)} \frac{1}{r} \frac{\partial q_1}{\partial \theta} + 2\rho h \frac{\partial^2 v}{\partial t^2}; \\ \Delta \gamma_r + \frac{1+v}{2r} \frac{\partial}{\partial \theta} \left( \frac{\partial \gamma_\theta}{\partial r} - \frac{1}{r} \frac{\partial \gamma_r}{\partial \theta} + \frac{\gamma_\theta}{r} \right) - \frac{1}{r^2} \left( \gamma_r + 2 \frac{\partial \gamma_\theta}{\partial \theta} \right) &= \frac{4\psi_r}{5\varepsilon_\tau} - \frac{3v''(1+v)}{5hE} \frac{\partial q_2}{\partial r} + \frac{\rho}{\tilde{E}} \frac{\partial^2 \gamma_r}{\partial t^2} - \\ - A' \frac{\rho}{\tilde{E}} \frac{\partial^3 \tilde{W}}{\partial r \partial t^2}; \\ \Delta \gamma_\theta + \frac{1+v}{1-v} \frac{1}{r} \frac{\partial}{\partial \theta} \left( \frac{\partial \gamma_r}{\partial r} + \frac{1}{r} \frac{\partial \gamma_\theta}{\partial \theta} + \frac{\gamma_r}{r} \right) - \frac{1}{r^2} \left( \gamma_\theta - 2 \frac{\partial \gamma_r}{\partial \theta} \right) &= \frac{8\psi_\theta}{5\varepsilon_\tau(1-v)} - \frac{6v''(1+v)}{5(1-v)hE'} \frac{1}{r} \frac{\partial q_2}{\partial r} + \\ + \frac{\rho}{\tilde{E}} \frac{\partial^2 \gamma_\theta}{\partial t^2} - A' \frac{\rho}{r\tilde{E}} \frac{\partial^3 \tilde{W}}{\partial \theta \partial t^2}; \\ K' \Delta w_\tau &= -q_2 + 2\rho h \frac{\partial^2 \tilde{w}}{\partial t^2}. \end{aligned} \quad (10)$$

$$\begin{aligned} \text{Тут } \psi_r &= \frac{\partial w_\tau}{\partial r} - \frac{1}{r} \frac{\partial \Omega}{\partial \theta}; \quad \psi_\theta = \frac{1}{r} \frac{\partial w_\tau}{\partial \theta} + \frac{\partial \Omega}{\partial r}; \quad \gamma_r = -\frac{\partial \bar{w}}{\partial r} - \frac{4}{5} \frac{1}{r} \frac{\partial \Omega}{\partial \theta}; \quad \gamma_\theta = -\frac{1}{r} \frac{\partial \bar{w}}{\partial \theta} + \frac{4}{5} \frac{\partial \Omega}{\partial r}; \\ \bar{w} &= w - \frac{2.4 + \chi_0}{3 + \chi_0} w_\tau + \frac{q_2 h}{E_0}; \quad E_0 = \frac{40}{9} (3 + \chi_0) E'; \quad \chi_0 = \frac{3v''}{2G/G' - v''}; \quad \Delta = \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2}; \\ \hat{W} &= w + 0.3A'h^2 \Delta w + 0.43\alpha_0 A_2 q_2 \frac{h}{E'}. \end{aligned}$$

Таким чином ми одержали систему п'яти рівнянь руху в шуканих функціях  $u, v, w_\tau, \gamma_r, \gamma_\theta, \Omega$ . До цих рівнянь мають бути приєднані граничні умови (9) та початкові умови при  $t = 0$ :

$$\begin{aligned} w &= w_0(r, \theta), \quad \frac{\partial w}{\partial t} = w_1(r, \theta), \\ u &= u_0(r, \theta), \quad \frac{\partial u}{\partial t} = u_1(r, \theta), \\ v &= v_0(r, \theta), \quad \frac{\partial v}{\partial t} = v_1(r, \theta), \end{aligned} \quad (11)$$

де  $w_0, v_0, u_0, u_1, v_1, w_1$  – задані компоненти початкового переміщення і початкової швидкості від точки  $(r, \theta)$ .

Систему рівнянь рівноваги (10) у тій частині, де вони описують згин пластинки, можна звести до більш звичного вигляду, якщо замість величин  $\gamma_r, \gamma_\theta, \psi_r, \psi_\theta$  підставити їх вирази через функції  $w, w_\tau, \Omega$ :

$$D\Delta^2 w_q = \left( 1 - \varepsilon_1 \Delta + \frac{\varepsilon' \rho h^4}{4G} \frac{\partial^2}{\partial t^2} \right) q_2 - m(1 - \varepsilon_1 h^2 \Delta) \frac{\partial^2 \tilde{w}(r, t)}{\partial t^2} - m \varepsilon' \frac{\rho h^2}{4G} \frac{\partial^4 \tilde{w}}{\partial t^4};$$

$$K' \Delta w_\tau = -q_2 + m \frac{\partial^2 \tilde{w}}{\partial t^2}; \quad \Delta \Omega - k^2 \Omega = \frac{\rho}{G} \frac{\partial^2 \Omega}{\partial t^2}, \quad (12)$$

де  $\varepsilon_1 = \frac{h^2}{10(1-\nu)} \left( 8 \frac{G}{G'} - 3\nu'' \right)$ ;  $D = \frac{2}{3} \frac{Eh^3}{1-\nu^2}$ ;  $m = 2\rho h$  – маса одиниці поверхні пластинки;

$\varepsilon' = 0.1 \left( 8 \frac{G}{G'} + \nu'' \right)$ ;  $w_q = w + \varepsilon_2 q_2 / D$ ;  $k^2 = \frac{5}{2} \frac{G'}{G} h^{-2}$ ;  $\varepsilon_2 = \frac{h^4}{20(1-\nu^2)} (1-\alpha) \left( \frac{E}{E'} + A' \frac{E}{G'} \right)$ ;

$\alpha = \frac{\nu'' \cdot G'}{2G}$ .

Отримані рівняння (12) враховують додатково інерцію обертання поперечних перерізів пластини та вплив нормального напруження  $\sigma_z$ . Якщо покласти нулю параметри  $\varepsilon'$  та  $A'$ , а також  $\Delta' \equiv 1 - \varepsilon \Delta$ , то ці фактори в рівняннях (12) враховуватись не будуть. Неврахування інерції веде до втрати правої частини в рівнянні Гельмгольца.

**Висновки.** Для виведення рівнянь рівноваги та граничних умов у круглій плиті, використано варіаційний принцип Лагранжа для повної енергії пружної системи. Отримані рівняння цілком співпадають за формою з відповідними умовами та рівняннями для пластин класичної теорії. Відмінність вносять лише коефіцієнти, що враховують поперечний зсув та обтиснення. Дані рівняння враховують додатково інерцію обертання поперечних перерізів пластини та вплив нормального напруження.

1. Амбарцумян С.А. Теория анизотропных пластин. Прочность, устойчивость и колебания / С.А. Амбарцумян. – 2-е изд., перераб. и доп. – М.: Наука, 1987. – 360 с.
2. Васильев В.В. Механика конструкций из композитных материалов / В.В. Васильев. – М.: Машиностроение, 1988. – 272 с.
3. Гринченко В.Т. Равновесие и установившиеся колебания упругих тел конечных размеров / В.Т. Гринченко. К.: Наукова думка, 1978. – 264 с.
4. Лурье А.И. Пространственные задачи теории упругости / А.И. Лурье. – М.: Гостехиздат, 1955. – 491 с.
5. Пелех Б.Л. Об одном обобщении теории упругих трансверсально-изотропных плит применительно к некоторым контактными задачам / Б.Л. Пелех, В.И. Швабюк // Сопrotивление материалов и теория сооружений: республиканский межведомственный научно-технический сборник. – 1975. – Вып. 26. – С. 40–48.
6. Швабюк В.И. Учет эффекта сжимаемости нормали в контактных задачах для трансверсально-изотропных плит / В.И. Швабюк // Прикладная механика. – 1980. – Т. 16. – №9. – С. 71–77.

УДК 004.413 (045)

Жигаревич О.К.

Луцький національний технічний університет

## ДОСЛІДЖЕННЯ ТА ПРЕДСТАВЛЕННЯ НАВЧАЛЬНОЇ СИСТЕМИ PROTUS 2.0

**Жигаревич О.К. Дослідження та представлення навчальної системи PROTUS 2.0.** Розробка навчального програмного забезпечення у нашій державі є глобальним явищем, яке включає в себе соціально – технічні системи. Сучасне програмне забезпечення характеризується масивністю, складністю, швидкою інтеграцією у соціальні відносини, застосуванням нових принципів при формулюванні задачі, проектуванні програмного забезпечення, розробці, супроводі, обміні та використанні. Саме взаємодія різних сфер впливу на розробку програмного забезпечення спонукала до застосування нових принципів при визначенні життєвого циклу та середовища розробки в цілому. Дослідження створеної навчальної системи **PROTUS 2.0.** дозволить вносити та корегувати нові навчальні матеріали, перейти на новий етап у представленні та оцінюванні знань.

**Ключові слова:** навчальна система, програмне забезпечення, дослідження, літературні джерела.

**Жигаревич О.К. Исследования и представления учебной системы PROTUS 2.0.** Разработка учебного программного обеспечения в нашей стране является глобальным явлением, которое включает в себя социально - технические системы. Современное программное обеспечение характеризуется массивностью, сложностью, быстрой интеграцией в социальные отношения, применением новых принципов при формулировке задачи, проектировании программного обеспечения, разработке, сопровождении, обмене и использовании. Именно взаимодействие различных сфер влияния на разработку программного обеспечения побудило к применению новых принципов при определении жизненного цикла и среды разработки в целом. Исследование созданной обучающей системы PROTUS 2.0. позволит вносить и корректировать новые учебные материалы, перейти на новый этап в представлении и оценке знаний.

**Ключевые слова:** учебная система, программное обеспечение, исследования, литературные источники.

**Zhyharevych O.K. Research and present educational system PROTUS 2.0.** The development of educational software in our country is a global phenomenon that includes social - technical systems. Modern software is characterized by massiveness, complexity, rapid integration into social relations, the use of new principles in the formulation of the problem, software design, development, maintenance, sharing and use. It is the interaction of different spheres of influence on software development led to the application of new principles in determining the life cycle and development environment as a whole. The study established educational system PROTUS 2.0. will adjust and make new training materials, move to a new stage in the presentation and evaluation of knowledge.

**Keywords:** educational system, software, research literature.

### Вступ

Архітектура навчальної системи, забезпечена кількома онтологіями та представлена, як спосіб вирішення проблем технічного обслуговування та повторного використання компонентів системи навчання. Впроваджена архітектура розширює можливості використання онтології, де представлення кожного компонента розроблене відповідно до конкретної онтології, яка дозволяє:

- сприяти чіткому розподілу завдань складових системи навчання;
- зробити явний зв'язок між компонентами;
- вказати інтерфейс для доступного оновлення компонентів;
- підкреслити переваги використання веб-семантики (Semantic Web) в розвитку навчальних систем.

Система Protus 2.0 пропонує два види персоналізації для кожного окремого студента: персоналізації на основі напряму навчання студентів та персоналізації на основі індивідуального сприйняття інформації.

Коли студент починає свою роботу у Protus 2.0, на першому кроці є вибір напряму навчання. Список напрямів навчання відкриваються для кожного студента за алгоритмом ArgioiAll, який заснований на вибраних опціях в процесі навчання (деталі про виконаних алгоритму в Protus представлені в роботі [1]). Перелік рекомендованого матеріалу, який повинен бути представлений студенту, створюється у відповідності з формою даних «Онтологія навчальної моделі програмного забезпечення» і рейтингу частих послідовностей, що надаються системою Protus 2.0. Всі рекомендовані дії виконуються в режимі реального часу під час навчальних сесій. Надана рекомендація, яка очікується, стимулює значну точність, щодо відповідності вимог (потреб) студентів до вивчення матеріалу і, таким чином, забезпечить більш високий рівень студентського сприйняття.

Запити до онтології даних були здійснені за допомогою SPARQL – мовою запитів для RDF [2], а мова семантики Web-правил – (SWRL) була використана для міркувань на основі правил [3]. Детальна інформація про реалізацію правил адаптації в Protus 2.0 представлено в [4], [5].

### Навчальна взаємодія студента з Protus 2.0

Новий (користувач) студент заповнює реєстраційну форму для того, щоб створити власний профіль і оновити «Онтологію навчальної моделі програмного забезпечення». Кожен профіль зберігає особисті дані, надані безпосередньо студентом, а саме: прізвище, ім'я користувача, попередні результати перевірки знань, переваги і т.д. (відомі як статична інформація), та інформацію про напрям навчання, поточні досягнення і поведінку (відомі як динамічна інформація). Коли студенти (користувачі) вже зареєстровані в системі, їх стилі навчання повинні бути перевірені. Студент повинен заповнити анкету, яка використовується для визначення його/її кращого стилю навчання [6]. Стиль навчання студента виявляє акцент на перевагу деяких методів представлення над іншими. Ці результати зберігаються в «Онтології навчальної моделі» (рис. 1), яку буде використано для первинної адаптації в Protus 2.0.

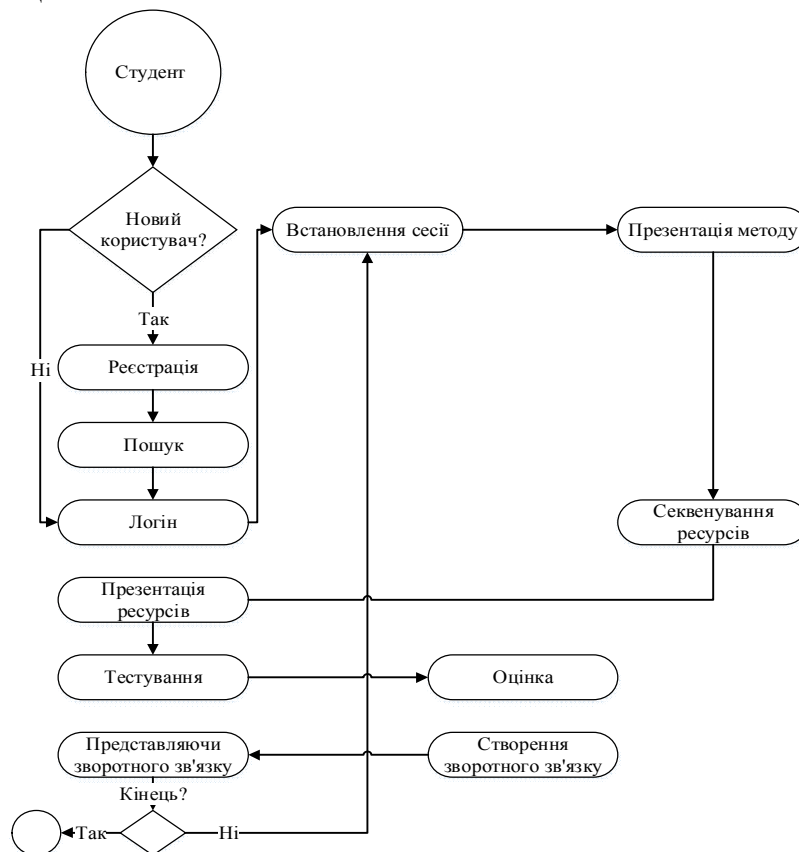


Рис. 1. Онтологія навчальної моделі

Коли студент (не залежно від того, чи він/вона є новим користувачем, або вже таким який був зареєстрованим) зареєструвався, ініціюється сесія, яка базується на основі індивідуального стилю навчання студента, що вже сформовано та збережено у «Онтології навчальної моделі програмного забезпечення», і рекомендуються йому/їй у вигляді послідовності занять (рис 1). Студент може вільно змінювати порядок занять, які він/вона проходить. Після вибору заняття з множини всіх занять, наявних в Protus 2.0, система вибирає метод презентації для заняття, який також базується на основі визначеного найкращого стилю навчання студента. Для решти занять студенти можуть вільно переключатись між презентаціями з використанням шкали середнього балу. Даний варіант був представлений, для того щоб початкова оцінка навчання користувача може ввести в оману в загальному процесі персоналізації. Коли студент виконає перелік навчальних завдань (тестів), система оцінює його рівень знань. Тест містить питання з кількома варіантами вибору відповідей і завдання коду завершення. Після того, як студент закінчив тест, Protus 2.0 забезпечує зворотний зв'язок студента на його/її відповіді і демонструє результат.

Рекомендації не можуть бути зроблені для всієї студенської групи разом, тому що навіть для студентів за однаковим напрямком навчання, їх здатність вирішувати завдання може бути абсолютно

різною через рівень їх знань. При такому підході відтворюємо метод об'єднання даних в якості першого кроку, до здійснення об'єднання студентів на основі їх вибраного напрямку навчання. Створені об'єднання використовуються для ідентифікації когерентних відборів, для зручності навчальної роботи. Після чого список рекомендацій та побажань може бути створений у відповідності до рейтингів, що надаються системою Protus 2.0.

### Стилі навчання

Відомо, що всі студенти мають різні дані до навчання, бажання і підходи. Психологи називають це індивідуальними відмінностями – стилю навчання. Тому, дуже важливо визначити стилі використання для різних типів студентів у відповідності до навчального середовища, яким вони надають перевагу, і виділити найбільш ефективні. Стилі навчання можуть бути визначені як унікальні способи, в яких студенти орієнтуються на сам процес, сприйняття та засвоєння нової інформації [7].

Є більш ніж сімдесят ідентифікованих підходів для дослідження та/або опису переважаючих стилів навчання. Під час дослідження використаний один такий інструмент збору даних, що називається індекс стилів навчання (ICN) [8]. ICN включає в себе 44 вільно доступних питань. Це інструмент багатоваріантного вибору стилю навчання, який оцінює зміни в їх індивідуальних перевагах за чотирма вимірами або напрямками (доменами) [9], такими як: обробка інформації, сприймання інформації та розуміння інформації. У кожній з чотирьох областей ICN є дві категорії:

- обробка інформації: активні студенти;
- сприйняття інформації: здатні до сприйняття та з розвинутою інтуїцією;
- засвоєння інформації: здатні візуально, наглядно сприймати;
- розуміння інформації: послідовні і глобальні.

Кращий стиль навчання може бути досліджений через надання студентові вибору дисциплін для навчання, які він/вона бажає [9].

На початку проведення тестувань програма Protus 2.0 видає запит про інформацію яка визначає статус курсу в онтології навчальної моделі конкретного студента. (рис. 2). Дані включають інформацію про поточні заняття і категорії стилю навчання студента в одному з чотирьох напрямків студента ICN (рис. 1). Запит відповідних ресурсів, які будуть представлені студентові базуються на основі інформації та відправляються в модуль програми. Крім того, контролюються та перевіряється вся робота студента, а також всі запити, які він/вона відправляє системі.

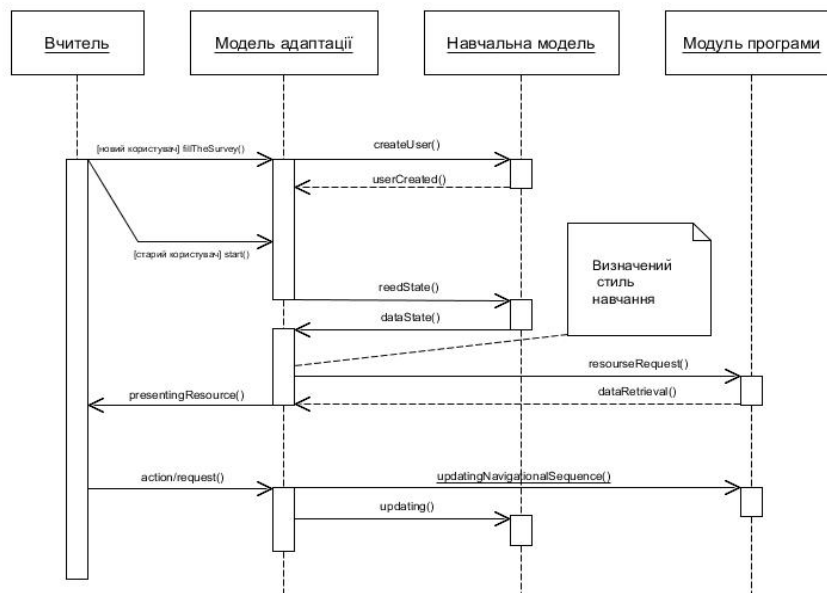


Рис. 2. Онтологія навчальної моделі студента

Якщо студент успішно вивчив, сприйняв потрібну інформацію, яка супроводжується і підтримується певними ресурсами, то система повинна додати той ресурс та деталі інформації про виконання до успішного навігаційного шаблону. Крім того, якщо студент взаємодіє із ресурсом та

протягом певного часу він/вона виконав тести і отримав бали, тоді система повинна запам'ятати результати роботи студента і відмітити, що студент вже переходить до наступного рівня.

Якщо студент не досяг необхідного рівня знань за результатами тестування, заснованого на методі презентації, що використовувалися для його/її певної категорії стилю навчання, то його/її поточні категорії стилю навчання буде змінено. У таких випадках, система змінює поточний стиль навчання студента на його альтернативний, з того ж напрямку (домена).

Стили навчання згруповані попарно (активні і відображувальні, змістовні і інтуїтивні, візуальні і вербальні, послідовні і глобальні), тому кожен стиль навчання має тільки один варіант в напрямку (домені).

Наприклад, якщо студент з вербального (словесного) стилю навчання взаємодіє з системою і протягом цієї взаємодії він/вона зверталися до відповідної концепції, але не здобували достатньої оцінки (необхідний рівень оцінки зберігається в глобальній величині, *required* (необхідно), тоді стиль навчання цього студента повинен бути змінений на його альтернативний з напрямку (домена) сприйняття інформації. Візуальний стиль навчання передбачає, що під час наступного тестування буде представлено ресурси, які визначені для підтримки конкретної категорії стилю навчання.

Висновок.

Проаналізовано наукові статті в яких досліджено стилі навчання, зроблено аналіз літературних джерел по темі. Наведено чітке визначення, щодо система Protus 2.0 програмного забезпечення. Побудована онтологія навчальної моделі.

Розглянуто та досліджено поняття стилів навчання. Запити до онтології даних здійснені за допомогою SPARQL – мовою запитів для RDF, а мова семантики Web-правил – (SWRL) була використана для міркувань на основі правил.

Досліджено систему Protus 2.0, яка забезпечує представляє взаємозв'язок викладачів програмного забезпечення, користувачів програмного забезпечення, та навчальну установу ВУЗ. Потрібно розуміти, що ключову роль у цій взаємодії відіграє людина, бо вона розробляє, користується, та навчається. Можна говорити про дану навчальну систему, як одну із галузей науки, яку потрібно, ще довго досліджувати.

1. Klačnja-Milićević, Vesin B., Ivanović M., Budimac, Z.: Integration of Recommendations into Java Tutoring System. In Proceedings of the 4th International Conference on Information Technology ICIT 2013 Jordan, Paper no 154. (2013)
2. Recommender System based on Intelligent Data Mining Techniques. Computer Science and Information Systems, Vol. 9, No. 2, 713-740. (2012)
3. Recommender System based on Intelligent Data Mining Techniques. Computer Science and Information Systems, Vol. 12, No. 2, 713-740. (2015)
4. SCORM 2012 3rd Edition - Overview, 16 November 2013, Advanced Distributed Learning. <http://www.adlnet.gov>.
5. Sparql – query language for RDF, [www.w3.org/TR/rdf-sparql-query/](http://www.w3.org/TR/rdf-sparql-query/) 34. Swartout, W., Tate, A.: Ontologies. Intelligent Systems and their Applications, IEEE, Vol. 14, No. 1, 18-19. (2015)
6. Henze, N., Dolog, P. Hejdl, W.: Reasoning and Ontologies for Personalized E- Learning in the Semantic Web. Educational Technology & Society, Vol. 7, No. 4, 82-97. (2014)
7. Brusilovsky, P., Vassileva, J.: Course sequencing techniques for large-scale webbased Education. International Journal of Continuing Engineering Education and Lifelong Learning 2013 - Vol. 13, No.1/2
8. Gascueña, J. M., Fernández-Caballero, A., González, P.: Domain Ontology for Personalized E-Learning in Educational Systems. In Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies, 456-458. (2015)

УДК 62-1/-9:004.5:004.453

Лактіонов О. І., аспірант

Полтавський національний технічний університет імені Юрія Кондратюка (ПНТУ)

**ВЕРСТАТ З ЧИСЛОВИМ ПРОГРАМНИМ КЕРУВАННЯМ ЯК ОБ'ЄКТ І СУБ'ЄКТ  
УПРАВЛІННЯ**

**Лактіонов О. І. Верстат з числовим програмним керуванням як об'єкт і суб'єкт управління.** Використовуючи системний аналіз і системний підхід, в статті розглядається верстат з числовим програмним керуванням як система, яка має чотири підсистеми з відповідними елементами, які між собою взаємодіють і об'єднані спільною кінцевою метою і результатом. Система розглядається як цілісність, яка має в своєму складі ряд взаємопов'язаних частин, кожна з яких впливає на загальні характеристики в цілому. Єдиний механізм функціонування системи забезпечується зв'язками між її складовими, які мають індивідуальні риси, властивості і особливе значення.

Розглядається верстат з числовим програмним управлінням як об'єкт і суб'єкт управління. Система розглядається як цілісність, яка має ряд взаємопов'язаних частин. Виділяється чотири основні підсистеми: неорганічна, органічна, система магнітного поля і соціальна підсистема. Відзначається, що об'єктом системи є верстат з ЧПК, який відповідає за точність виготовлення деталі; суб'єктом виступає оператор, що забезпечує якість всіх операцій. Наводиться список індикаторів професійної компетентності оператора верстата з числовим програмним управлінням.

Вперше ЧПК розглядається як система, де рівноправні і обов'язкові чотири підсистеми. Складові підсистеми: неорганічна, органічна, система магнітного поля і соціальна підсистема. Всі підсистеми мають зворотну дію, котра є основною складовою управління.

**Ключові слова.** Верстат з числовим програмним керуванням, оператор, суб'єкт та об'єкт управління, неорганічна підсистема, органічна підсистема, система магнітного поля, соціальна підсистема.

**Лактіонов А. И. Станок с числовым программным управлением как объект и субъект управления.** Используя системный анализ и системный подход, в статье рассматривается станок с числовым программным управлением как система, которая имеет четыре подсистемы с соответствующими элементами, которые между собой взаимодействуют и объединены общей конечной целью и результатом. Система рассматривается как целостность, которая имеет в своем составе ряд взаимосвязанных частей, каждая из которых влияет на общие характеристики в целом. Единственный механизм функционирования системы обеспечивается связями между ее составляющими, которые имеют индивидуальные черты, свойства и особое значение.

Рассматривается станок с числовым программным управлением как объект и субъект управления. Система рассматривается как целостность, которая имеет ряд взаимосвязанных частей. Выделяется четыре основные подсистемы: неорганическая, органическая, система магнитного поля и социальная подсистема. Отмечается, что объектом системы является станок с ЧПУ, который отвечает за точность изготовления детали; субъектом выступает оператор, обеспечивающий качество всех операций. Приводится список индикаторов профессиональной компетентности оператора станка с числовым программным управлением.

Впервые ЧПУ рассматривается как система, где равноправны и обязательны четыре подсистемы. Составляющие подсистемы: неорганическая, органическая, система магнитного поля и социальная подсистема. Все подсистемы имеют обратную связь, которая является основной составляющей управления.

**Ключевые слова.** Станок с числовым программным управлением, оператор, субъект и объект управления, неорганическая подсистема, органическая подсистема, система магнитного поля, социальная подсистема.

**Laktionov O.I. Numerical control machine tool as the object and the subject of control.** Interaction control components together is difficult due to the difference of each individual element approach, which affects the overall performance as a whole. The only mechanism of functioning of the system provides communication between its components, which have individual traits, characteristics and special value.

Explore the machine with numerical control as the object and the subject of management. Consider a system of integrity, having in its composition a number of interrelated parts. Separate the four main subsystems: inorganic, organic, magnetic field system and the social subsystem. Determine the subject and the object of the system.

The first CNC viewed as a system where equal, be sure to four subsystems. The components of the subsystem: inorganic, organic, magnetic field system and the social subsystem. All subsystems have a feedback, which is the main control component.

**Keywords.** The machine with numerical control, the operator, the subject and object of management, subsystem, inorganic, organic subsystem, the magnetic field system, social subsystem.

**Вступ та постановка проблеми.** Застосування верстатів з числовим програмним керування (ЧПК) є прогресивним напрямом автоматизації виробництва. Сукупність впливів на механізми верстату, що забезпечують виконання технологічного циклу обробки, відноситься до управління верстатом. Підсистемою управління є пристрій або сукупність пристроїв, що реалізують ці впливи. Комплексна реалізація механізму управління сприяє підвищенню якості кінцевої продукції.

Взаємодія елементів системи керування у сукупності є складним завданням в силу різницевого підходу кожного окремого елементу, яких впливає на загальні характеристики в цілому. Єдиний механізм функціонування системи забезпечується зв'язками між її складовими, які мають індивідуальні риси, властивості й особливе значення.

**Мета роботи.** Дослідити верстат з числовим програмним керуванням як об'єкт і суб'єкт управління. Розглянути систему як цілісність, що становить у своєму складі низку взаємозалежних частин. Відокремити чотири основні підсистеми: неорганічна, органічна, система магнітного поля та соціальна підсистема. Визначити суб'єкт та об'єкт системи.

**Аналіз останніх досліджень і публікацій.** На сьогодні, серед наукових праць присвячених висвітленню питання системного підходу варто відзначити роботи Р. Акоффа, Ф. О. Емері [1], К.А. Маца [2], М. В. Аніщенко [3], М. В. Плахотний, В. О. Дашенко [4].

Р. Акоффа, Ф. О. Емері наводять базові поняття системи цілеспрямованих дій, та відокремлюють поняття система, ґрунтуючись на чисельних наукових дослідженнях того часу.

Принципи ізоморфізму, ієрархічності, вакантної ніші, цілісності, самодостатності, автономності, внутрішнього і зовнішнього сполучення, амбівалентності описуються на сторінках монографії К. А. Маца.

Проте, незважаючи на масштабні наукові дослідження у зазначеній сфері питання системного аналізу та системного підходу у рамках теорії управління складними системами залишається відкритим та потребує детального опрацювання.

**Вклад основного матеріалу дослідження.** Використовуючи системний аналіз та системний підхід розглянемо верстат із числовим програмним керуванням як систему, яка має чотири підсистеми з відповідними елементами, які між собою взаємодіють, об'єднані загальною кінцевою метою і результатом.

Система розглядається як цілісність, що становить у своєму складі низку взаємозалежних частин, кожна з яких впливає на загальні характеристики в цілому. Єдиний механізм функціонування системи забезпечується зв'язками між її складовими, які мають індивідуальні риси, властивості й особливе значення.

На сьогодні, теорія управління складними системами відокремлює три типи зв'язків:

- зв'язки, щодо забезпечення посилення загального ефекту керування – синергетичні зв'язки спільної дії;
- зв'язки, що сприяють формуванню соціально-економічних відносин – функціонально необхідні зв'язки;
- зв'язки, що відрізняються особливістю застосування для певної системи – надлишкові зв'язки.

З погляду загально-філософського підходу, система – це сукупність визначених елементів, між якими є закономірний зв'язок чи взаємодія. Характеристики кожного окремого елементу, становлять загальну структуру системи в цілому.

З позиції верстата із ЧПК як системи керування, варто відокремити складові підсистеми: неорганічна, органічна, система магнітного поля та соціальна підсистема.

На математичному рівні формула має вигляд:

$$\text{ЧПК} = 4 \text{ підсистеми} + \text{функція}$$

Взаємодія чотирьох підсистем є якістю.

З точки зору сензитивно-інтегрованої якості і емерджентності до неорганічної підсистеми варто віднести деталі верстата, елементи кріплення, функціональні вузли (рис.1).



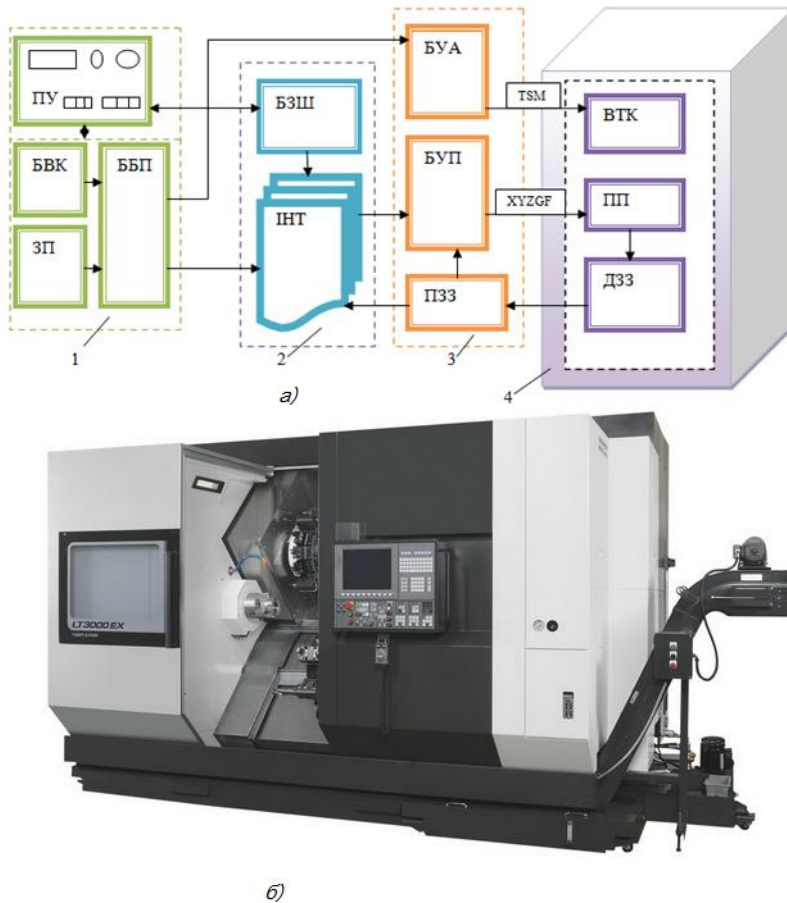


Рисунок 1 – Верстат з ЧПК а) Структурна схема системи ЧПК верстата: 1 – блоки введення; 2 – блоки відпрацювання; 3 – блоки виводу; 4 – верстат з виконавчими пристроями.

ПУ – панель управління; ЗП – зчитувач перфострічок; БВК – блок введення корекції; ББП – блок буферної пам'яті; БЗШ – блок завдання швидкості; ІНТ - інтерполятор; БУП - блок управління приводом; БУА – блок управління електроавтоматикою; ПП – привід подач; ВТК – влаштування технологічних команд; ДЗЗ – датчики зворотного зв'язку; ПЗЗ – пристрій зворотного зв'язку; X, Y, Z, I, J, K, G, F – команди переміщень; T, M, S, F – технологічні команди

б) Фото верстату з ЧПУ TWIN STAR LT 3000 EX [5]

До органічної підсистеми: мастила, фарба, електричні апарати.

Елементами соціальної підсистеми є люди. Без виконавця верстат не працює.

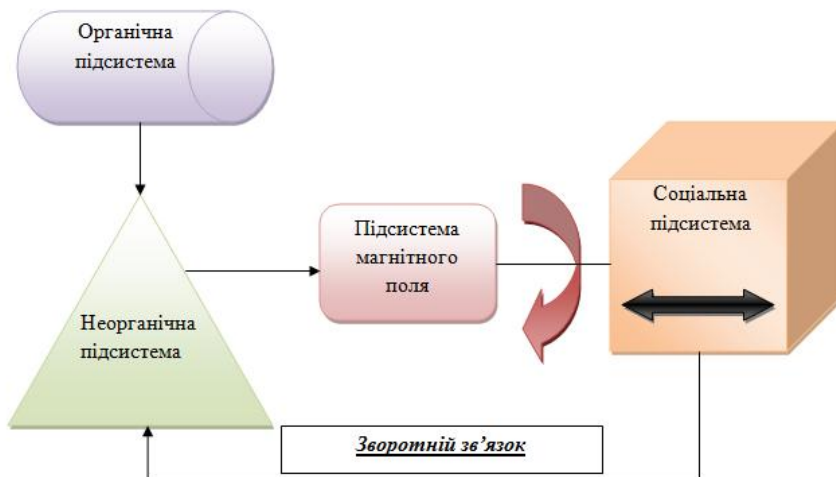


Рисунок 2 – Обернений зворотній зв'язок

Усі підсистеми мають обернений зворотній зв'язок, що є основною складовою управління (рис.2).

Управлінський цикл являє собою сукупність послідовно здійснюваних управлінських операцій, в ході яких суб'єкт управління досягає бажаних результатів. У процесі здійснення кожного окремого циклу здійснюється ряд певних операцій, які послідовно змінюють один одного. Ця послідовність операцій, яка здійснюється суб'єктом, пов'язана з тим, що будь-який конкретний процес є циклічним і розчленовується на певні етапи, стадії, з необхідністю слідує один за одним.

Управлінський цикл складається з чотирьох основних принципів: постановка завдання, планування, організація, інструктаж (рис.3).



Рисунок 3 – Управлінський цикл

Блок контролю функціонує за оцінкою оператора відповідно до стандартів.

Так як верстат з ЧПК є об'єктом системи, який відповідає за точність виготовлення деталі.

Суб'єкт – забезпечує якість всіх операцій. Головним суб'єктом верстата з ЧПК є оператор.

Рівень професійної компетентності оператора верстата з числовим програмним керуванням [6], оцінюється наступними індикаторами:

1) Знання, уміння і навички:

– будова верстата з числовим програмним керуванням;

– правила управління та обслуговування верстата з числовим програмним керуванням;

– основи гідравліки, механіки та електротехніки;

– системи програмного управління верстатами;

– кінематичні схеми верстатів з числовим програмним керуванням;

– оформлення технічної документації;

– виконання управління верстатом з пульту;

– обслуговування багатоцільових верстатів з числовим програмним управлінням.

2) Професійно-важливі якості:

– професійні навички програмування верстату;

– швидкість сенсорно-моторних реакцій;

– просторове представлення;

- пам'ять на числа, форми і просторове розташування.
- 3) Мотиваційна складова:
  - вільний вибір активності при здійсненні професійної діяльності;
  - оцінка ефективності виконання професійної діяльності;
  - вибірковість уваги до аспектів ситуації;
  - валентність об'єкта;
  - виявлення зони цілей, релевантних мотиву;
  - підвищена сприйнятливості до об'єкта актуальної мотивації;

***Висновки з даного дослідження і перспективи подальших розробок у даному напрямку.***

Вперше ЧПК розглядається як система, де рівноправними, обов'язковими є чотири підсистеми. Складові підсистеми: неорганічна, органічна, система магнітного поля та соціальна підсистема. Усі підсистеми мають обернений зворотній зв'язок, що є основною складовою управління.

Соціальна система, що керує верстатом, у вигляді оператора, включає наступні ознаки об'єкту порівняння: знання, уміння і навички; професійно важливі якості; мотивацію. Відповідність всім індикаторам професійної компетенції сприяє підвищенню якості продукції на виході виробничого циклу.

1. Р. Акофф. О целеустремленных системах / Р. Акофф, Ф. Эмери – М.: Книга по Требованию, 2012. – 270 с.
2. Маца К.А. Системы неорганические, органические, социальные: свойства и принципы организации: монография / К.А.Маца. – К.: Обрії, 2008. – 196 с.
3. Аніщенко М. В. Системи числового програмного керування : навч. посібник / М. В. Аніщенко. – Х. : Підручник НТУ "ХП", 2012. – 312 с.
4. Плахотний М. В. Алгоритмічне та програмне забезпечення системи керування станком з ЧПУ / М. В. Плахотний, В. О. Дашенко // Інформаційні технології та комп'ютерна інженерія. Тези доповідей Міжнародної науково-практичної конференції. м. Вінниця, 19–21 травня 2010 року. – Вінниця: ВНТУ, 2010. – С.56-57.
5. Андреев Г. И. Работа на станках с ЧПК. Система ЧПК FANUC / Г. И. Андреев, Д. Ю. Кряжев. Санкт-Петербург: Взлет, 2007. – 84 с.
6. Минин М.Г., Соловьев А.Н. Компетентностный подход в оценке учебных достижений студентов технического вуза// Известия Томского политехнического университета. 2007. – Т. 310. – №2. – С. 258-260.

УДК 539.3

Максимович В.М.<sup>1</sup>, Соляр Т.Я.<sup>2</sup>, Приходько О.С.<sup>1</sup>

<sup>1</sup> Луцьк. держ. техн. ун-т, Луцьк,

<sup>2</sup> Ін-т прикл. проблем механіки і математики ім. Я. С. Підстригача НАН України, Львів

### ЧИСЛОВИЙ АЛГОРИТМ ВИЗНАЧЕННЯ НАПРУЖЕНЬ БІЛЯ ПІДКРІПЛЕНИХ ОТВОРІВ У ПЛАСТИНКАХ.

**Максимович В.М., Соляр Т.Я., Приходько О.С. Числовий алгоритм визначення напружень біля підкріплених отворів у пластинках.** Розроблено алгоритм визначення напружень у пластинках із пружними включеннями, які послаблені отворами. Алгоритм базується на модифікованих інтегральних рівняннях, за яких умови на межі отвору задовольняються тотожно. Інтегральні рівняння розв'язано чисельно за допомогою методу механічних квадратур. Проведено дослідження напружень біля включень із отворами різної форми. Встановлено характерні особливості у розподілі напружень залежно від форми включень та отворів.

**Ключові слова:** пластинки, включення, отвори, напруження, модифіковані інтегральні рівняння

**Максимович В.М., Соляр Т.Я., Приходько О.С. Численный алгоритм определения напряжений возле подкрепленных отверстий в пластинках.** Разработан алгоритм определения напряжений в пластинках с упругими включениями, которые ослаблены отверстиями. Алгоритм базируется на модифицированных интегральных уравнениях, при которых условия на границе отверстия удовлетворяются тождественно. Интегральные уравнения решены численно с помощью метода механических квадратур. Проведено исследование напряжений возле включений с отверстиями различных форм. Установлены характерные особенности в распределении напряжений в зависимости от формы включений и отверстий.

**Ключевые слова:** пластинки, включения, отверстия, напряжения, модифицированные интегральные уравнения

**V. M. Maksymovych, T. Ya. Solyar, O. S. Prykhod'ko. Numerical algorithm for determining stresses near reinforcements openings in the plates.** An algorithm to determine the stresses in the plates with elastic inclusions, weakened by the openings, is developed. The algorithm is based on the modified integral equations, for which the conditions on the opening boundary are satisfied identically. The integral equations are solved numerically by the method of mechanical quadratures. Investigation of the stresses near inclusions with openings of different forms is carried out. The singularities in distribution of stresses, depending on the form of inclusions and openings, are determined.

**Keywords:** plates, inclusions, openings, stresses, modified integral equations

**Постановка проблеми.** Для зменшення концентрації напружень біля отворів їх підкріплюють пружними кільцями. Для проведення розрахунків на міцність пластинок із підкріпленими отворами необхідно дослідити концентрацію напружень. Така проблема зводиться до визначення напружень у пластинках із пружними включеннями, які додатково послаблені отворами.

#### **Аналіз останніх досліджень і публікацій.**

Дослідженню напруженого стану пластинок із включеннями присвячено значно менше праць, ніж однорідним пластинкам [2-8]. Підхід до дослідження таких задач на основі методу граничних інтегральних рівнянь (МГІР) запропоновано у [8-10].

**Невирішені частини проблеми.** Застосування МГІР стосовно до пластинок нескінченних розмірів (зокрема, для півплощин й смуг) та для випадків, коли включення розміщені близько до отвору або отвори мають складну форму, ускладнюється. Тому в літературі для окремих класів таких пластинок із отворами і тріщинами побудовані модифіковані інтегральні рівняння, за яких умови на вибраній межі пластинки задовольняються тотожно [1, 9].

**Метою дослідження** є розробка підходу до розрахунку напружено-деформованого стану (НДС) пластинок із включеннями, що послаблені отворами, який базується на модифікованих інтегральних рівняннях. Для реалізації підходу побудовано розв'язок типу Гріна, який входить в інтегральне рівняння.

**Основні результати дослідження.** Розглядається задача про дослідження НДС ізотропної пластинки з пружними включеннями з іншого матеріалу, які послаблені отворами. Прийmemo: пластинка займає область  $D$ , обмежену контуром  $L_D$ ; пластинка перебуває під дією прикладених на нескінченності зусиль; в точках  $c_j$ , які належать включенню та матриці, діють зосереджені сили  $(X_j, Y_j)$ ,  $(j = 1, \dots, J)$ ; пластинка перебуває в умовах плоского напруженого стану, межа отвору у включенні вільна від навантажень. Пружні сталі для включень позначатимемо через  $G_j, \chi_j$  ( $j = 1, \dots, J$ ), а матриці – через  $G_0, \chi_0$ . Тут  $G_j$  – модулі зсуву,  $\chi_j = (3 - \nu_j) / (1 + \nu_j)$ ,  $\nu_j$  –

коефіцієнти Пуассона. Вважатимемо також, що на межі поділу матеріалів має місце ідеальний механічний контакт.

**Основні співвідношення.** Виразимо загальний розв'язок поставленої задачі через комплексні потенціали Колосова-Мухелішвілі  $\Phi(z)$ ,  $\Psi(z)$  [6]. Використаємо співвідношення для визначення вектора напружень  $q_\Gamma = N_\Gamma + iT_\Gamma$  у довільній точці  $z$  кривої  $\Gamma$  на дотичній до неї площинці через комплексні потенціали [6, 9]:

$$q_\Gamma(z) = \Phi(z) + \overline{\Phi(z)} + d\bar{z} / dz [z\overline{\Phi'(z)} + \overline{\Psi(z)}], \quad (1)$$

де  $dz$  – диференціал змінної  $z$  на кривій,  $N_\Gamma$ ,  $T_\Gamma$  – нормальні і дотичні проекції вектора на кривій  $\Gamma$ .

Похідні від переміщень пластинки з пружними характеристиками  $G, \chi$  на кривій  $\Gamma$  визначатимемо за формулами

$$2G(u' + iv') = \chi\Phi(z) - \overline{\Phi(z)} - d\bar{z} / dz [z\overline{\Phi'(z)} + \overline{\Psi(z)}], \quad (2)$$

де  $u' + iv' = d(u + iv) / dz$ .

**Модифіковані інтегральні рівняння.** Розглянемо пластинку, що займає область  $D$ , межа цієї області ненавантажена, пластинка перебуває під дією зусиль на нескінченності (для пластинок нескінченних розмірів) та зосереджених сил.

Введемо у розгляд потенціали Мухелішвілі  $\Phi_j^D$ ,  $\Psi_j^D$ ,  $j=1,2$ , які в довільній точці  $t = c \in D$  мають полюси

$$\Phi_j^D \sim \frac{C_j}{c-z}, \quad \Psi_j^D \sim \frac{\bar{C}_j}{c-z} - \frac{c\bar{C}_j}{(c-z)^2}. \text{ при } C_1 = 1, C_2 = i.$$

Тоді інтегральне зображення загального розв'язку задачі теорії пружності для області  $D$  запишемо у вигляді [4]

$$\begin{aligned} \Phi(z) &= \int_L [R(t)\Phi_1^D(z,t) + I(t)\Phi_2^D(z,t)] ds + \Phi_D(z), \\ \Psi(z) &= \int_L [R(t)\Psi_1^D(z,t) + I(t)\Psi_2^D(z,t)] ds + \Psi_D(z), \end{aligned} \quad (3)$$

де  $L = L_1 + L_2 + \dots + L_J$ ,  $\Phi_D(z)$ ,  $\Psi_D(z)$  – потенціали для суцільної області  $D$ , які відповідають прикладеному до пластинки зовнішньому навантаженню.

За побудовою потенціали (3) автоматично задовольняють умову відсутності напружень на межі області  $D$  та вектор напружень  $q$ , що відповідає цим потенціалам, є непервними при переході через контур  $L$ .

Інтегральні рівняння матимуть вигляд

$$\beta_k 2\pi \frac{ds}{dt} (iR - I) + S_k(\Phi, \Psi) = f, \quad z \in L_k, k = 1, \dots, J, \quad (4)$$

де  $f = -S_k(\Phi_D, \Psi_D)$ ,

$$S_k(\Phi, \Psi) = \alpha_k \Phi(z) - \overline{\Phi(z)} - d\bar{z} / dz [z\overline{\Phi'(z)} + \overline{\Psi(z)}], \quad z \in L. \quad (5)$$

$$\alpha_k = \frac{H_k \chi_0 - \chi_k}{H_k - 1}, \quad \beta_k = \frac{1}{2} \frac{H_k (\chi_0 + 1) + (\chi_k + 1)}{H_k - 1}, \quad H_k = G_k / G_0.$$

Тут функції  $\Phi, \Psi$  – визначаються за формулами (3) при  $z \in L$ , у яких сингулярні інтеграли розглядаються в сенсі головного значення за Коші.

**Числовий алгоритм розв'язування інтегральних рівнянь.** Для розв'язування рівняння (5) використаємо метод механічних квадратур. При цьому рівняння контуру інтегрування запишемо параметрично у вигляді  $\xi = \alpha(\theta)$ ,  $\eta = \beta(\theta)$ ,  $0 \leq \theta \leq 2\pi$ . Розглянемо випадок одного

включення. Введемо вузлові точки на контурі  $z_v = \tilde{x}_v + i\tilde{y}_v$ ,  $t_k = x_k + iy_k$ , де  $\tilde{x}_k = \alpha(\tau_k)$ ,  $\tilde{y}_k = \beta(\tau_k)$ ,  $x_k = \alpha(\theta_k)$ ,  $y_k = \beta(\theta_k)$ ,  $\theta_k = Hk$ ,  $\tau_n = \theta_n - H/2$ ,  $H = 2\pi/N$ ,  $N$  – кількість вибраних вузлових точок.

Поклавши у співвідношеннях (4)  $x = \tilde{x}_v$ ,  $y = \tilde{y}_v$ ,  $v = 1, \dots, N$  і замінивши інтеграли квадратурною формулою прямокутників, яка для періодичних підінтегральних функцій, що можуть містити ядро Коші є формулою підвищеної точності, отримаємо систему рівнянь вигляду

$$\beta_k 2\pi \left( \frac{ds}{dt} \right)_v (iR_v - I_v) + H \sum_{n=1}^N (S_{vn}^{(R)} R_n + S_{vn}^{(I)} I_n) = f_v, \quad v = 1, \dots, N, \quad (6)$$

де

$$S_{vn}^{(R)} = S_k(\Phi_1^D(z_v, t_n), \Psi_1^D(z_v, t_n)), \quad S_{vn}^{(I)} = S_k(\Phi_2^D(z_v, t_n), \Psi_2^D(z_v, t_n)), \\ f_v = -S_k(\Phi_D(z_v), \Psi_D(z_v)), \quad R_n = R(x_n, y_n) s'_n, \quad I_n = I(x_n, y_n) s'_n, \quad s'_n = (ds/d\theta)|_{\theta=\theta_n}.$$

Тут враховано, що функції  $S_k(\Phi, \Psi)$  містять інтеграли типу Коші.

Доповнимо цю систему рівняннями, які отримані заміною параметрів  $t$  і  $\tau$  на  $t + H/2$  і  $\tau + H/2$ , відповідно.

$$\beta_k 2\pi \left( \frac{ds}{dt} \right)_n (iR_n - I_n) + H \sum_{v=1}^N (S_{nv}^{(R)} R_v + S_{nv}^{(I)} I_v) = f_n, \quad n = 1, \dots, N. \quad (7)$$

Системи рівнянь (6) і (7) є замкненими. Визначивши величин  $R, I$  у вузлових точках потенціали (а далі й напруження) у довільних точках пластинки знайдемо за квадратурними формулами. Під час знаходження напружень на межі поділу необхідно додатково врахувати складові, що виникають при граничному переході у формулі Сохоцького.

**Включення, послаблене отвором.** Розглянемо нескінченну пластинку з включенням, яке послаблене отвором, межа якого  $\Gamma$  вільна від навантаження. За область  $D$  виберемо нескінченну площину з отвором, обмеженим контуром  $\Gamma$ . Задача визначення допоміжних потенціалів  $\Phi_j^D(z, c)$ ,  $\Psi_j^D(z, c)$  зводиться до знаходження потенціалів  $\Phi_0, \Psi_0$ , які мають полюси

$$\Phi_0 \sim \frac{C}{c-z}, \quad \Psi_0 \sim \frac{\bar{C}}{c-z} - \frac{\bar{c}C}{(c-z)^2}, \quad (8)$$

за умови, що відповідні до цих потенціалів напруження на межі області  $D$  відсутні. Тут  $C$  – довільна комплексна стала.

Розглянемо детальніше випадок, коли отвір має форму еліпса. Для побудови потенціалів  $\Phi_0, \Psi_0$  необхідно розглянути задачу теорії пружності для пластинки із еліптичним ненавантаженим отвором за умови, що у внутрішніх точках комплексні потенціали  $\varphi_0(z)$  та  $\psi_0(z)$  мають такі особливості [1]:

$$\varphi_0 \sim -C \ln(z-c), \quad \psi_0 \sim \bar{C} \ln(z-c) + \bar{c}C \frac{1}{z-c}, \quad (9)$$

де  $C, c$  – довільні комплексні сталі

Розв'язок знаходимо з використанням методу Мусхелішвілі [6]. Для цього використовуємо функцію  $z = \omega(\zeta)$ , що конформно відображає область  $D$  на зовнішність одиничного круга, де  $\omega(\zeta) = R(\zeta + m/\zeta)$ ,  $R = (a+b)/2$ ,  $m = (a-b)/(a+b)$ ,  $a$  і  $b$  – півосі еліптичного отвору,  $|\zeta| > 1$ .

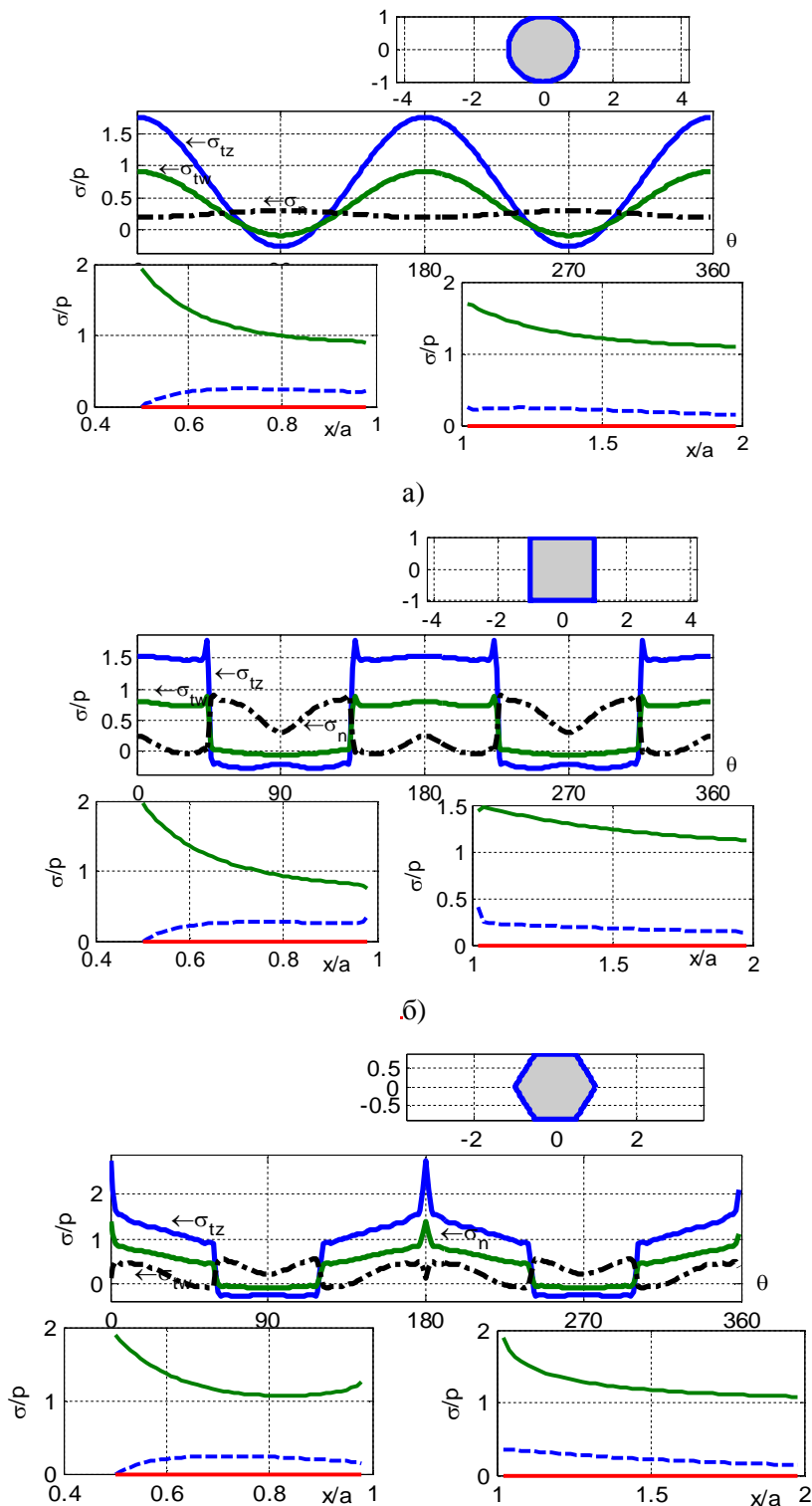
Комплексні потенціали  $\varphi(\zeta) = \varphi_0(\omega(\zeta))$ ,  $\psi(\zeta) = \psi_0(\omega(\zeta))$  в площині  $\zeta$  матимуть вигляд [1]

$$\varphi(\zeta) = U(\zeta) + \overline{\gamma C} \frac{\zeta_*^2}{(\zeta - \zeta_*)}, \quad \psi(\zeta) = V(\zeta) + \gamma C \frac{1}{(\zeta - \zeta_0)} - \frac{\zeta(1+m\zeta^2)}{\zeta^2 - m} \varphi'(\zeta), \quad (10)$$

де  $c = \omega(\zeta_0)$ ,  $\zeta_* = 1/\bar{\zeta}_0$ ,  $\gamma = \frac{\overline{\omega(\zeta_0)} - \overline{\omega(1/\zeta_0)}}{\omega'(\zeta_0)}$ ,

$$U(\zeta) = -C \left[ \ln(\zeta - \zeta_0) + \ln\left(\frac{\zeta - \zeta_*}{\zeta}\right) \right], \quad V(\zeta) = \bar{C} \left[ \ln(\zeta - \zeta_0) + \ln\left(\frac{\zeta - \zeta_*}{\zeta}\right) \right].$$

**Результати розрахунків.** Розглянемо пластинку із включенням, яке додатково послаблене круговим отвором радіуса  $R$ . Результати розрахунків для випадку, коли включення кругове радіуса  $a$  при  $R=a/2$ ,  $E_0 = 0,5E$  наведено на рис. 1.а.



в)

Рис. 1. Кругове (а), квадратне (б) та шестикутне (в) включення із круговим отвором

Аналогічні результати для квадратного і шестикутного включень наведено на рис.1.б,в.

На основі наведених графіків випливає, що: при м'якшому включенні концентрація напружень біля отвору зменшується. Для включень усіх форм виявилось, що коефіцієнт концентрації напружень (ККН)  $\approx 2$ . Тобто, форма включень мало вплинула на ККН. Значно більше впливає форма включень на напруження на межі поділу матеріалів.

Аналогічні результати для жорсткішого включення при  $E_0 = 2E$  наведено на рис. 2, 3

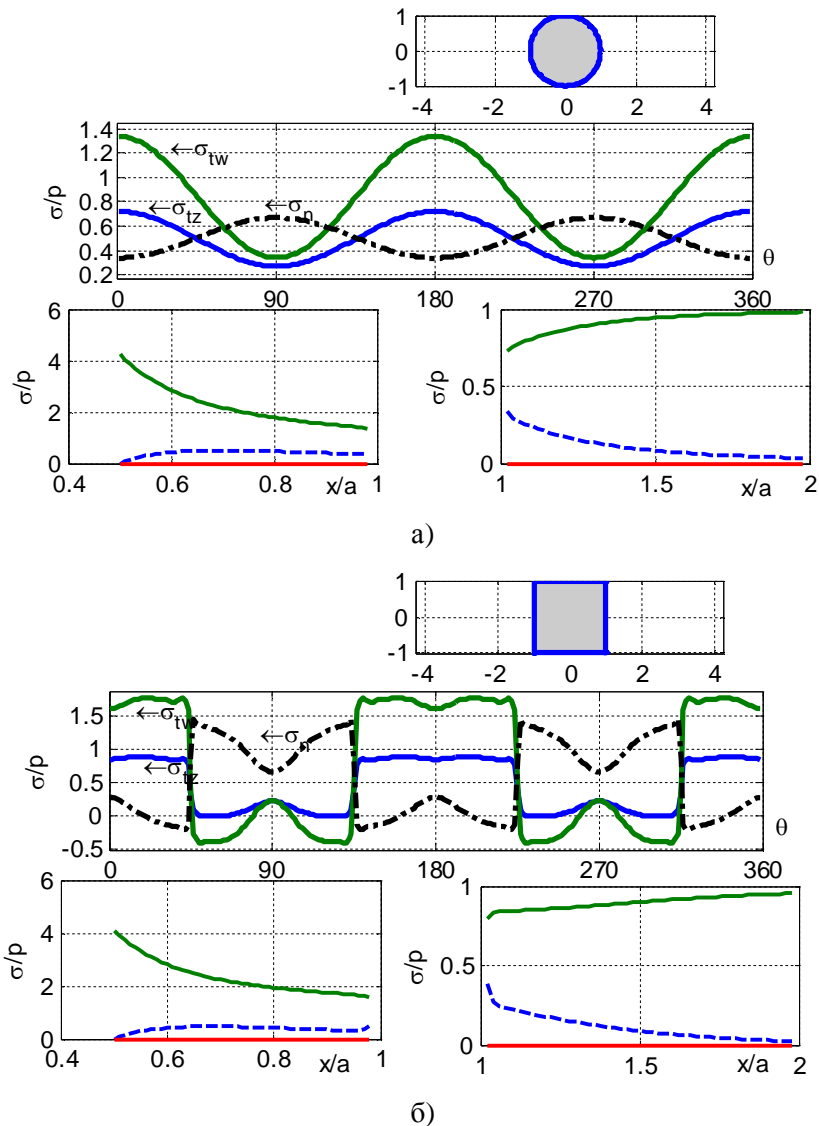


Рис. 2. Кругове (а) та квадратне (б) включення із круговим отвором



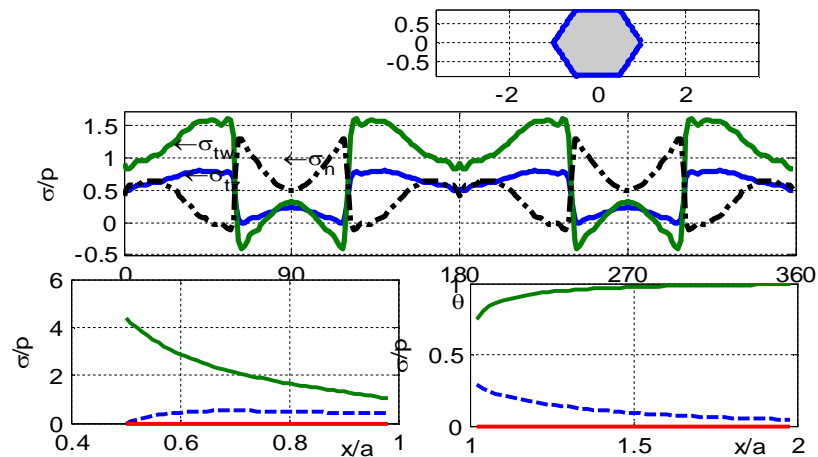


Рис. 3. Шестикутне включення із круговим отвором

На основі аналізу отриманих результатів розрахунків впливає, що: при жорсткішому включенні концентрація напружень біля отвору зростає. Для включень усіх форм виявилось, що коефіцієнт концентрації напружень (ККН)  $\approx 4$ . Тобто, форма включень мало вплинула на ККН. Значно більше впливає форма включень на напруження на межі поділу матеріалів, причому напруження найбільші в околі кутових точок.

**Висновки.** Побудовано модифіковані інтегральні рівняння для пластинок із пружними включеннями, які додатково послаблені отворами. Основна перевага таких рівнянь полягає в тому, що граничні умови на деяких межах задовольняються тотожно, що дозволяє підвищити точність розрахунків та зменшити об'єм обчислень. Розв'язування інтегральних рівнянь виконано чисельно з використанням методу механічних квадратур. Проведене тестування вказує на високу точність та ефективність розробленого підходу. З використанням розробленого алгоритму досліджено напруження у включеннях та в їх околі залежно від форми включень. Встановлено характерні особливості у розподілі напружень залежно від форми включень та їх пружних характеристик.

1. Божидарнік В.В. Визначення напруженого стану біля крайових тріщин у пластині з отвором складної форми / В.В. Божидарнік, О.В. Максимович // Фіз.-хім. механіка матеріалів. – 2010. – № 1. – С. 19-26.
2. Довідниковий посібник. / під заг.ред. В.В. Панасюка, Т12: Ін'єкційні технології відновлення робоздатності пошкоджених споруд тривалої експлуатації / В.І. Маруха, В.В. Панасюк, В.П. Силованюк Під ред. акад. НАН України В.В. Панасюка – Львів: Вид-во "Сполом", 2009. – 262 с.
3. Космодамианский А.С. Анизотропные многосвязные среды / А.С. Космодамианский // Учебн. Пособие. – Донецк: Изд-во Донецк. ун-та, 1970. – С. 233.
4. Максимович В.М. Визначення напружень біля пружних включень у пластинках складної форми / В.М. Максимович, О.О. Приходько, Т.Я. Соляр // Мат. методи і фіз.-мех. поля. – 2014. – **57**, №3. – С. 109-118.  
*Te same: Maksymovych V. M., Prykhod'ko O. S., Solyar T. Ya. Determination of stresses near elastic inclusions in plates of complex shape // J. Math. Sci. – 2016. – **217**, No. 3. – P. 271-282.*
5. Мартинович Т.Л. Крайові умови в інтегральній формі задачі про напружений стан в анізотропній пластинці з несиметрично підкріпленим краєм / Т.Л. Мартинович, В.К. Божидарнік // Вісник Львів. у-ту ім. Франка. Сер. мех.-мат. – 1972. – Вип. 7. – С. 112-118.
6. Мусхелишвили Н.И. Некоторые основные задачи математической теории упругости / Н.И. Мусхелишвили // М.: Наука, 1966. – С. 708.
7. Савин Г.Н. Распределение напряжений около отверстий / Г.Н. Савин // К.: Наук. думка, 1968. – С. 888.
8. Савин Г.М. Довідник з концентрації напружень / Г.М. Савін, В.І. Тульчій // К.: Вища школа, 1976. – С. 412.
9. Саврук М.П. Численный анализ в плоских задачах теории трещин / М.П. Саврук, П.Н. Осив, И.В. Прокопчук // К.: Наук. думка, 1989. – С. 248.
10. Саврук М. П. Сингулярные интегральные уравнения плоских задач теплопроводности и термоупругости для кусочно-однородной плоскости с трещинами / М.П. Саврук, В.М. Зеленьак // Фіз.-хім. механіка матеріалів. – 1986. – **22**, № 3. – С. 82-88.

УДК 539.3

Максимович О.В., Лавренчук С.В.

Луцький національний технічний університет

## ЧИСЛОВИЙ АНАЛІЗ НАПРУЖЕНЬ БІЛЯ ШТАМПІВ І ТРІЩИН У АНІЗОТРОПНІЙ ПІВПЛОЩИНІ ЗА ВРАХУВАННЯ ТЕРТЯ

**Максимович О.В., Лавренчук С.В. Числовий аналіз напружень біля штампів і тріщин у анізотропній півплощині за врахування тертя.** Запропоновано алгоритм розрахунку напружень біля штампа і тріщин в анізотропній півплощині, в якому враховано тертя під штампом та контакт берегів тріщин. Розв'язування контактної задачі для штампа параболічної форми за врахування тертя проведено на основі підходів Галіна та Мусхелішвілі. Дослідження напружень біля тріщин виконано методом інтегральних рівнянь, причому задачу визначення контактних напружень на берегах тріщин зведено до задачі квадратичного програмування. Виконано аналіз напружень біля тріщин для ізотропного і анізотропних матеріалів залежно від розміщення тріщин.

**Ключові слова:** плоска контактна задача, анізотропна півплощина, кулонівське тертя, контакт берегів тріщин, інтегральне рівняння, задача квадратичного програмування, потенціали Лехніцького.

**Максимович О.В., Лавренчук С.В. Численный анализ напряжений возле штампов и трещин в анизотропной полуплоскости с учетом трения.** Предложен алгоритм расчета напряжений у штампа и трещин в анизотропной полуплоскости, в котором учтены трения под штампом и контакт берегов трещин. Решение контактной задачи для штампа параболической формы с учетом трения проведено на основе подходов Галина и Мусхелишвили. Исследование напряжений у трещин выполнено методом интегральных уравнений, причем задачу определения контактных напряжений на берегах трещин сведено к задаче квадратичного программирования. Выполнен анализ напряжений у трещин для изотропного и анизотропных материалов в зависимости от размещения трещин.

**Ключевые слова:** плоская контактная задача, анизотропная полуплоскость, кулоновское трение, контакт берегов трещин, интегральное уравнение, задача квадратичного программирования, потенциалы Лехницкого.

**Maksymovych O.V., Lavrenchuk S.V. Numerical analysis of stresses at the stamps and cracks in anisotropic half-plane taking into account friction.** The algorithm for calculating of stresses at the stamp and cracks in the anisotropic half-plane, which takes into account friction under the stamp and contact crack is proposed. The solution of the contact problem for the parabolic shape stamp on account of friction has been performed based on the approaches of Galina and Muskhelishvili. The research of stresses at the cracks has been executed by the method of integral equations and the problem of determining of contact stresses on crack has been kept to a quadratic programming problem. The analysis of stresses at the cracks for isotropic and anisotropic materials depending on the placement cracks has been executed. Analysis of stresses at the cracks for isotropic and anisotropic materials depending on the placement cracks is executed.

**Keywords:** flat contact task, anisotropic half-plane, the Coulomb friction, contact crack, integral equations, quadratic programming problem, potentials of Lehnitskyu.

**Актуальність та постановка проблеми.** Відомо, що в області контакту штампів із пружними тілами можуть виникнути високі напруження. А це, в свою чергу, може призвести до зародження і розвитку крайових тріщин під різними кутами до межі тіла. Очевидно, що для практики дослідження такої контактної поведінки є на часі. Наприклад, при розгляді підшипників, в яких ролики контактують із біговою доріжкою, має місце багаторазове навантаження відносно тріщин, які вже з'явилися раніше, тобто тут будуть мати місце повторні навантаження. Тому тріщини, що зародились на початкових стадіях, будуть розвиватися за втомним механізмом. У відповідних розрахунках необхідно також врахувати контакт берегів тріщин та тертя під штампом, яке може суттєво впливати на шукані напруження.

**Метою даної роботи** є дослідження напружень біля тріщин у анізотропній півплощині за врахування тертя під штампом та контакту берегів тріщин на основі розподілу напружень, які виникають під штампом за врахування тертя.

**Аналіз досліджень та публікацій.** Проблема визначення напружень у пружних тілах із тріщинами, які контактують із штампами достатньо повно вивчена для ізотропних тіл. При розгляді таких задач, як правило, тіло моделюють півплощиною, штамп приймають параболічним, враховують контакт берегів тріщин [4, 8]. Для спрощення приймають, що розподіл напружень під штампом описується за відомою формулою Герца, яка справедлива для випадку гладкого штампа. При врахуванні сил тертя наближено приймають, що дотичні напруження під штампом пропорційні вказаним вище нормальним напруженням. Аналогічний підхід розглядався для анізотропних тіл у роботі [10]. В той же час у літературі побудовано точний розв'язок контактної задачі теорії пружності [2,6], в якому враховано тертя під штампом і який не описується розподілом Герца.

**Визначення контактних напружень в суцільній півплощині.** Розглянемо анізотропну півплощину  $y < 0$ , яка навантажена в області  $L_u$  ( $c < x < d$ ) штампом з основою  $y = f(x)$ . Прийmemo, що напруження під штампом задовольняють умовам рівноваги

$$\int_c^d \sigma_y(x) dx = P, \quad \int_c^d x \sigma_y(x) dx = M,$$

та закону Кулона, згідно до якого дотичні напруження пов'язанні з відповідними нормальними за допомогою співвідношення  $\tau_{xy} = k\sigma_y$ ; де  $P$  і  $M$  – вертикальна сила та момент, що прикладені до штампа, а  $k$  – коефіцієнт кулонівського тертя. Також при цьому приймемо, що границя поза областю контакту (позначимо цю ділянку областю  $L_\sigma$ ) вільна від навантаження, тобто, що на ній відсутніми є як нормальні, так і дотичні напруження під штампом.

Тоді граничні умови для знаходження комплексних потенціалів, що визначені при  $y < 0$  запишуться у вигляді

$$\operatorname{Re}[s_1\Phi(x) + s_2\Psi(x)] = 0, \quad \text{при } -\infty < x < \infty, \quad (1)$$

$$\operatorname{Re}[\Phi(x) + \Psi(x)] = 0, \quad \text{при } x \in L_\sigma, \quad (2)$$

$$2\operatorname{Re}[q_1\Phi(x) + q_2\Psi(x)] = f'(x), \quad \text{при } x \in L_u. \quad (3)$$

де  $s_1$  та  $s_2$  – корені характеристичного рівняння [1],  $\alpha_{12}$ ,  $\alpha_{22}$  та  $\alpha_{26}$  – пружні сталі відповідного анізотропного матеріалу, а  $q_j = \alpha_{12}s_j + \frac{\alpha_{22}}{s_j} - \alpha_{26}$ ,  $j = \overline{1, 2}$ .

**Розв'язок контактної задачі.** На основі [2] потенціали можуть бути зображені у вигляді

$$\Phi(z) = \frac{S_2}{S_2 - S_1} F(z), \quad \Psi(z) = -\frac{S_1}{S_2 - S_1} F(z), \quad (4)$$

де  $S_j = k + s_j$ ,  $F(z)$  – аналітична при  $y < 0$  функція.

Умова  $\tau_{xy} = k\sigma_y$  виконується автоматично всюди на межі півплощини. Підставивши (4) у третю граничну умову при  $x \in L_u$ , отримаємо

$$2\operatorname{Re}[\lambda F(x)] = f'(x), \quad (5)$$

де  $\lambda = \frac{S_2q_1 - S_1q_2}{S_2 - S_1} = -k \left( \alpha_{12} - \frac{\alpha_{22}}{s_1s_2} \right) + \alpha_{22} \left( \frac{1}{s_1} + \frac{1}{s_2} \right) - \alpha_{26}$ .

Вимагаючи, щоби при  $x \in L_\sigma$   $\sigma_y = 0$ , отримаємо умову [2]

$$\operatorname{Re} F(x) = 0. \quad (6)$$

Легко перевірити, що при цьому при  $x \in L_\sigma$  також виконується умова  $\tau_{xy} = 0$ .

Знайдемо функцію  $F(z)$  методом лінійного спряження [6]. Для цього продовжимо її, поклавши при  $y > 0$   $F(z) = -\overline{F(z)}$ . Тоді умови (5) і (6) переписуться у вигляді

$$F^+ - F^- = 0, \quad \text{при } x \in L_\sigma;$$

$$F^+ - gF^- = f_1(x), \quad \text{при } x \in L_u,$$

де  $g = \lambda/\lambda$ ,  $f_1(x) = -f'(x)/\lambda$ .

На основі [6] звідси знаходимо

$$F(z) = \frac{1}{2\pi i X(z)} \int_c^d \frac{f_1(t) X^+(t) dt}{t - z} + \frac{C}{X(z)}, \quad (7)$$

де  $X(z) = (z - c)^\rho (z - d)^{1-\rho}$ , причому  $X^- = gX^+$ ,  $\rho = \frac{\ln g}{2\pi i}$ ,  $C$  – довільна дійсна стала.

Для визначення сталої  $C$ , врахуємо те, що у випадку, коли до межі півплощини прикладено навантаження з рівнодійною  $(P_X, P_Y)$ , то потенціали Лехніцького при  $z \rightarrow \infty$  будуть мати вигляд [2]

$$\Phi(z_1) \sim -\frac{1}{2\pi i} \frac{P_X + s_2 P_Y}{s_1 - s_2} \frac{1}{z_1} = -\frac{1}{2\pi i} \frac{S_2}{s_1 - s_2} \frac{P_Y}{z_1}, \quad \Psi(z_2) \sim \frac{1}{2\pi i} \frac{P_X + s_1 P_Y}{s_1 - s_2} \frac{1}{z_2} = \frac{1}{2\pi i} \frac{S_1}{s_1 - s_2} \frac{P_Y}{z_2}.$$

Звідси випливає, що для випадку розглядуваної нами контактної задачі при  $z \rightarrow \infty$  отримаємо, що

$$F(z) \sim \frac{P_Y}{2\pi i} \frac{1}{z}.$$

А це в свою чергу зі співвідношення (7) дасть нам рівність  $C = \frac{P}{2\pi i}$ .

Таким чином маємо

$$F(z) = -\frac{1}{2\pi i \lambda X(z)} \int_c^d \frac{f'(t) X^+(t) dt}{t-z} + \frac{P}{2\pi i X(z)} = \frac{1}{2\pi i |\lambda| X(z)} \int_c^d \frac{f'(t) |X^+(t)| dt}{t-z} + \frac{P}{2\pi i X(z)}, \quad (8)$$

Тут враховано  $X^+(t) = |X^+(t)| (e^{\pi i})^{1-\rho} = -|X^+(t)| e^{-\pi i \frac{\ln g}{2\pi i}} = -|X^+(t)| \sqrt{\frac{\lambda}{\lambda}}$ .

Контактні напруження будемо шукати за формулою  $\sigma_y = 2 \operatorname{Re} F^-$ .

Знайдемо момент  $M$  сил, які виникають під штампом при отриманому розв'язкові (7) відносно початку координат. Він визначатиметься за формулою

$$M = 2 \operatorname{Re} F_1,$$

де  $F_1 = \int_c^d t F^-(t) dt$ .

Для знаходження величини  $F_1$  врахуємо

$$\int_{\Gamma} z F(z) dz = \int_c^d t (F^+ - F^-) dt = (g-1) \int_c^d t F^- dt + \int_c^d t f_1 dt,$$

де  $\Gamma$  – замкнений контур, який вміщує відрізок  $(c, d)$ , інтегрування вздовж якого проходить за годинниковою стрілкою.

Звідси

$$F_1 = \frac{1}{g-1} \int_{\Gamma} z F(z) dz - \frac{1}{g-1} \int_c^d t f_1 dt.$$

Далі будуть виникати інтеграли вигляду

$$I = \int_c^d s(t) X^+(t) dt, \quad I_1 = \int_c^d \frac{s(t) dt}{X^+(t)},$$

які також аналогічно [6] можна звести до знаходження контурних інтегралів

$$I = \frac{1}{1-g} \int_{\Gamma} s(t) X(t) dt, \quad I_1 = \frac{g}{g-1} \int_{\Gamma} \frac{s(t) dt}{X(t)}. \quad (9)$$

де  $s(z)$  – аналітична функція.

Ці контурні інтеграли можуть бути обчислені за методом лишків, якщо врахувати розклади в ряди Лорана при  $z \rightarrow \infty$

$$X(z) = z + l D_1 + \frac{l^2 D_2}{z} + \frac{l^3 D_3}{z^2} + \dots, \quad \frac{1}{X(z)} = \frac{1}{z} + \frac{l C_1}{z^2} + \dots, \quad (10)$$

де  $D_1 = d_1 - x_c/l$ ,  $D_2 = d_2$ ,  $D_3 = d_3 + d_2 x_c/l$ ,  $C_1 = -D_1$ ,  $d_1 = -(1-2\rho)$ ,  $d_2 = 2\rho(\rho-1)$ ,  $d_3 = 2\rho(\rho-1)(2\rho-1)/3$ ,  $x_c = (d+c)/2$ ,  $l = (d-c)/2$ .

Для знаходження контурного інтегралу врахуємо, що при  $z \rightarrow \infty$

$$F(z) = \frac{1}{2\pi i} \left[ \frac{P}{z} + \frac{1}{z^2} \left( l C_1 P - \int_c^d f_1(t) X^+(t) dt \right) + \dots \right].$$

Тоді використовуючи теорему про лишки знаходимо

$$F_1 = -\frac{lC_1P + \frac{1}{\lambda} \int_c^d f'(t) X^+(t) dt}{g-1} + \frac{1}{\lambda(g-1)} \int_c^d t f'(t) dt.$$

Звідси отримуємо

$$M = 2 \operatorname{Re} F_1 = lC_1P - \frac{1}{\lambda_2} \operatorname{Im} \left[ \int_c^d f'(t) X^+(t) dt \right] = lC_1P + \frac{1}{|\lambda|} \int_c^d f'(t) |X^+(t)| dt. \quad (11)$$

**Приклади.**

а) Розглянемо випадок плоского штампа, що повертається на кут  $\varepsilon$ , до якого прикладені сила  $P$ . Тоді покладаючи в (8)  $f' = \varepsilon$ , отримуємо

$$F_\varepsilon(z) = -\frac{\varepsilon}{2\pi i \lambda (1-g) X(z)} \int_\Gamma \frac{X(t) dt}{t-z} + \frac{P}{2\pi i X(z)} = -\frac{\varepsilon}{\lambda(1-g)} \left( 1 - \frac{z + lD_1}{X(z)} \right) + \frac{P}{2\pi i X(z)}. \quad (12)$$

Момент знаходимо за формулою (9)

$$M = lC_1P - \frac{\varepsilon}{2\lambda_2^2} \operatorname{Re} \left[ \bar{\lambda} \int_\Gamma X(z) dz \right] = lC_1P + \frac{\pi\varepsilon}{\lambda_2} l^2 D_2. \quad (13)$$

Тут враховано, що має місце рівність  $\int_\Gamma P_x dz = -2\pi i l^2 D_2$ .

З цієї формули може бути визначений кут повороту у випадку, коли заданий момент  $M$ , який прикладено до штампа

$$\varepsilon = -\frac{\lambda_2 (PlC_1 - M)}{\pi l^2 D_2}. \quad (14)$$

Звідси, зокрема, для випадку ізотропного матеріалу маємо

$$\alpha_{22} = 1/E, \quad \alpha_{12} = -\nu/E, \quad \alpha_{26} = 0, \quad s_1 \rightarrow i, \quad s_2 \rightarrow i.$$

При цьому мають місце наступні спрощення

$$\lambda = -k \left( \alpha_{12} - \frac{\alpha_{22}}{s_1 s_2} \right) + \alpha_{22} \left( \frac{1}{s_1} + \frac{1}{s_2} \right) = \frac{-k(1-\nu) - 2i}{E}, \quad \lambda_2 = -\frac{2}{E};$$

і при  $c = -l$ ,  $d = l$  одержимо, що  $C_1 = (1 - 2\rho)$ ,  $D_2 = 2\rho(\rho - 1)$ .

З формули (14) отримаємо

$$\varepsilon = \frac{Pl(1 - 2\rho) - M}{E\pi\rho(\rho - 1)l^2}.$$

Цей розв'язок збігається з наведеними в [6] співвідношеннями для ізотропної задачі.

б) Розглянемо випадок параболічного штампа, коли  $f' = x/R + \varepsilon$ , до якого прикладена сила  $P$ , де  $\varepsilon$  – кут повороту штампа,  $R$  – радіус закруглення. Тоді

$$F_\varepsilon(z) = -\frac{1}{2\pi i \lambda R (1-g) X(z)} \int_\Gamma \frac{(t + \varepsilon R) X(t) dt}{t-z} + \frac{P}{2\pi i X(z)} = \frac{1}{2i\lambda_2 R} \left( z + \varepsilon R - \frac{z^2 + (lD_1 + \varepsilon R)z + lD_1\varepsilon R + l^2 D_2 - P\lambda_2 R / \pi}{X(z)} \right).$$

Момент знаходимо за формулою (11)

$$M = lC_1P - \frac{1}{2R\lambda_2^2} \operatorname{Re} \left[ \bar{\lambda} \int_\Gamma (z + \varepsilon R) X(z) dz \right] = lC_1P + \frac{\pi l^2}{R\lambda_2} (D_2\varepsilon R + lD_3).$$

Тут враховано, що  $\int_\Gamma (z + \varepsilon R) X dz = -2\pi i l^2 (\varepsilon R D_2 + lD_3)$ .

З цієї формули може бути визначений кут повороту у випадку, коли заданий момент, який прикладено до штампа

$$D_2 R \varepsilon = (M - l C_1 P) \frac{R \lambda_2}{\pi l^2} - l D_3. \quad (15)$$

Розглянемо випадок, коли область контакту невідома. Використаємо заміну  $z = \zeta + x_c$ . Тоді з (12) отримаємо

$$F_\varepsilon = \frac{1}{2i\lambda_2 R} \left( \zeta + \varepsilon' - \frac{\zeta^2 + (ld_1 + \varepsilon')\zeta + ld_1 \varepsilon' + l^2 d_2 - P\lambda_2 R / \pi}{X_0(\zeta)} \right), \quad (16)$$

де  $\varepsilon' = x_c + \varepsilon R$ ,  $X_0(\zeta) = (\zeta + l)^\rho (\zeta - l)^{1-\rho}$ .

Область контакту знаходимо з умов обмеженості напружень при  $\zeta = \pm l$ , які набувають вигляду

$$ld_1 + \varepsilon' = 0, \quad l^2 + ld_1 \varepsilon' + l^2 d_2 - P\lambda_2 R / \pi = 0.$$

До цих рівнянь долучаємо взаємозв'язок між моментом і поворотом штампа

$$d_2 \varepsilon' = (M - l C_1 P) R \lambda_2 / \pi l^2 - l d_3.$$

Звідси отримаємо

$$\varepsilon' = -ld_1, \quad l = \sqrt{\frac{P\lambda_2 R}{2\rho(1-\rho)\pi}}, \quad x_c = \frac{M}{P} + \frac{d_1 l}{3}, \quad \varepsilon R = -\frac{M}{P} - \frac{4d_1 l}{3}. \quad (17)$$

У нових змінних одержимо

$$F_\varepsilon(z) = \frac{1}{2i\lambda_2 R} \left( \zeta + \varepsilon' - \frac{\zeta^2 - l^2}{X_0(\zeta)} \right) = -\frac{P}{4\pi i \rho(1-\rho)l^2} \left( \zeta - ld_1 - \frac{\zeta^2 - l^2}{X_0(\zeta)} \right).$$

Звідси для нормальних напружень під штампом у нових змінних отримаємо наступне співвідношення

$$\sigma_y = -\frac{1}{R|\lambda|} |l + \xi|^{1-\rho} |l - \xi|^\rho = P \frac{\sin \pi \rho}{2\pi \rho(1-\rho)l^2} |l + \xi|^{1-\rho} |l - \xi|^\rho, \quad (18)$$

де  $\xi = x - x_c$ .

**Визначення напружень біля тріщин.** Для визначення напружень в півплощині з похилою крайовою тріщиною будемо використовувати розроблений в роботах [1, 5, 9] алгоритм, який базується на методі граничних інтегральних рівнянь. При цьому знаходження контактних напружень на берегах тріщин проводилось на основі підходу [5], в якому ця задача зводиться до задачі квадратичного програмування. Для його реалізації використовувалося отриманий вище розв'язок контактної задачі для суцільної півплощини, для якого в області контакту діють нормальні напруження (18) та відповідні їм дотичні напруження (пов'язані з нормальними за законом Кулона).

**Результати розрахунків.** Розглянемо випадок дії параболічного штампа, коли його центр розміщено в точці  $x_0$ , тріщина довжиною  $L$  виходить на межу пластинки в точці  $(0,0)$ , кут між тріщиною і віссю  $Ox$  рівний  $\beta$  (рис. 1) та між штампом і основою має місце тертя, що описується законом Кулона.

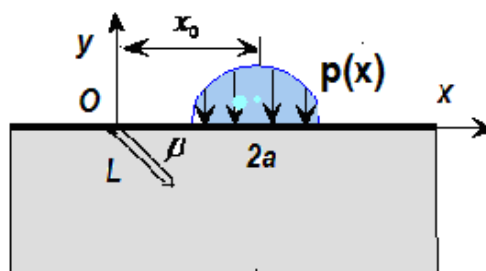


Рис. 1 – Схема задачі

Розподіл відносних коефіцієнтів інтенсивності напружень (КІН)  $F_{II} = K_{II} / P_0 \sqrt{\pi L}$  для ізотропної пластинки при  $L/a = 1$ ,  $\beta = 90^\circ$  та при значеннях коефіцієнта тертя  $k$  відповідно: 0; 0.25; 0.5 і 0.75 наведено на рис. 2,а (значення  $k$  нанесено біля кривих) залежно від параметра  $\lambda = x_0/L$ , де  $a$  – півширина ділянки контакту,  $P_0 = 2|P|/\pi a$ . Штриховими лініями на даному рисунку наведено відносні КІН  $F_I = K_I / P_0 \sqrt{\pi L}$ .

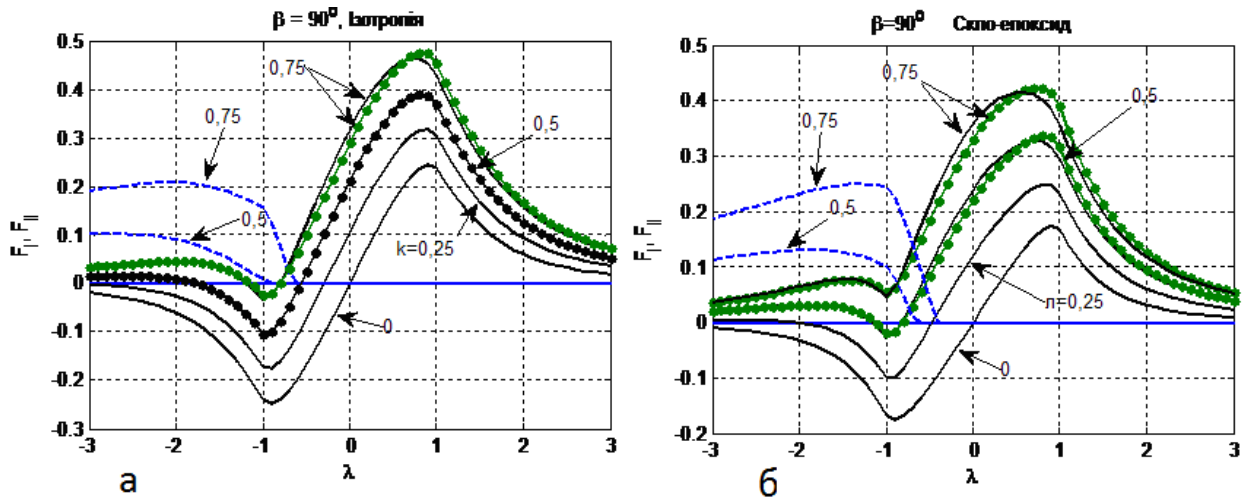


Рис. 2 – Відносні КІН для тріщин, нахилених під кутом  $\beta = 90^\circ$

Аналогічні результати розрахунків для матеріалу скло-епоксид, для якого відношення максимального модуля пружності  $E_x$  до мінімального  $E_y$  дорівнює 3, наведено на рис. 2,б.

На рис. 2 крапками при значеннях  $k$  0.5 та 0.75 наведено результати розрахунків для випадку, коли контактні напруження під штампом описувались розподілом Гауса [4, 10]. При менших значеннях кулонівського коефіцієнта тертя розраховані КІН для двох розглянутих розподілів напружень штампом практично не відрізняються.

Аналогічні результати розрахунків для кутів нахилу тріщин  $\beta = 150^\circ$  та  $\beta = 30^\circ$  наведено відповідно на рис. 3 та на рис. 4.

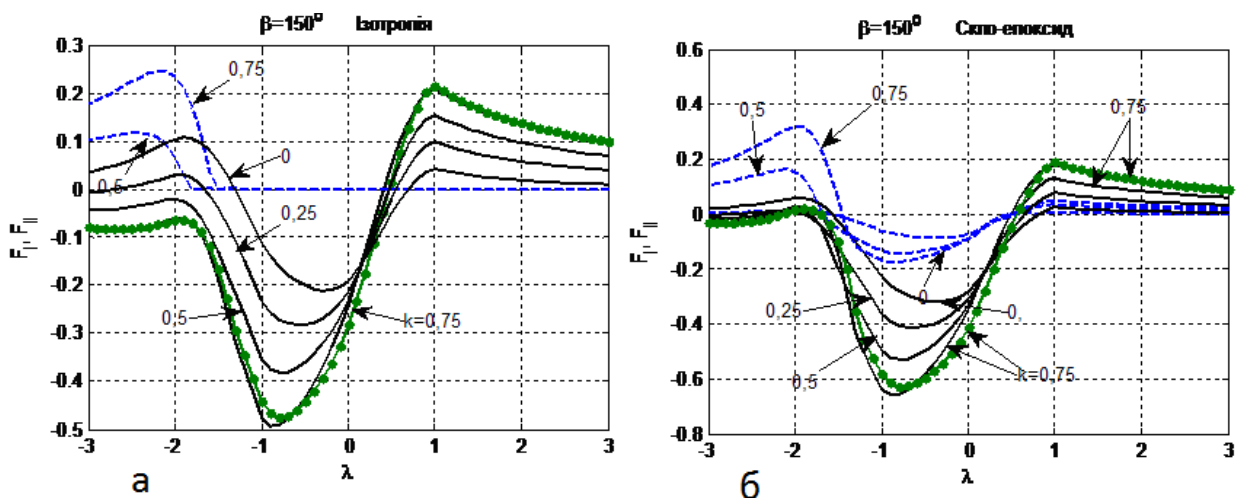


Рис. 3 – Відносні КІН для тріщин, нахилених під кутом  $\beta = 150^\circ$



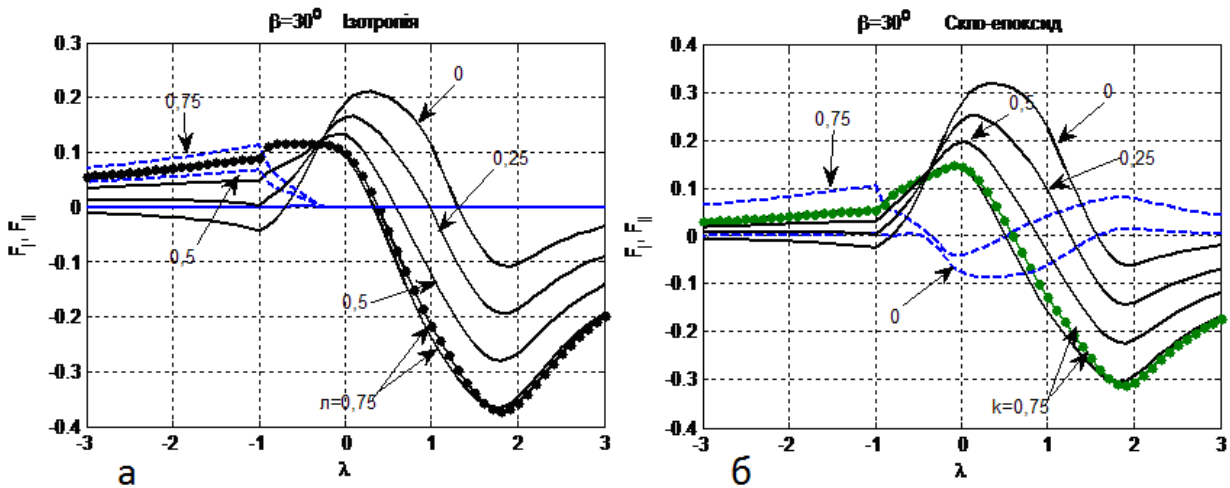


Рис. 4 – Відносні КІН для тріщин, нахилених під кутом  $\beta = 30^\circ$

Наведені на рис. 3-4 значення КІН для ізотропного матеріалу практично збігаються з даними, які отримані в роботах [4, 7] іншим методом.

Розраховані відносні КІН для істотно анізотропного матеріалу *CFI*, для якого відношення модулів пружності  $E_x/E_y = 400/8.62$ , наведено на рис. 5,а. А на рис 5,б наведено результати розрахунків для цього ж матеріалу, коли модуль пружності є максимальним у вертикальному напрямку.

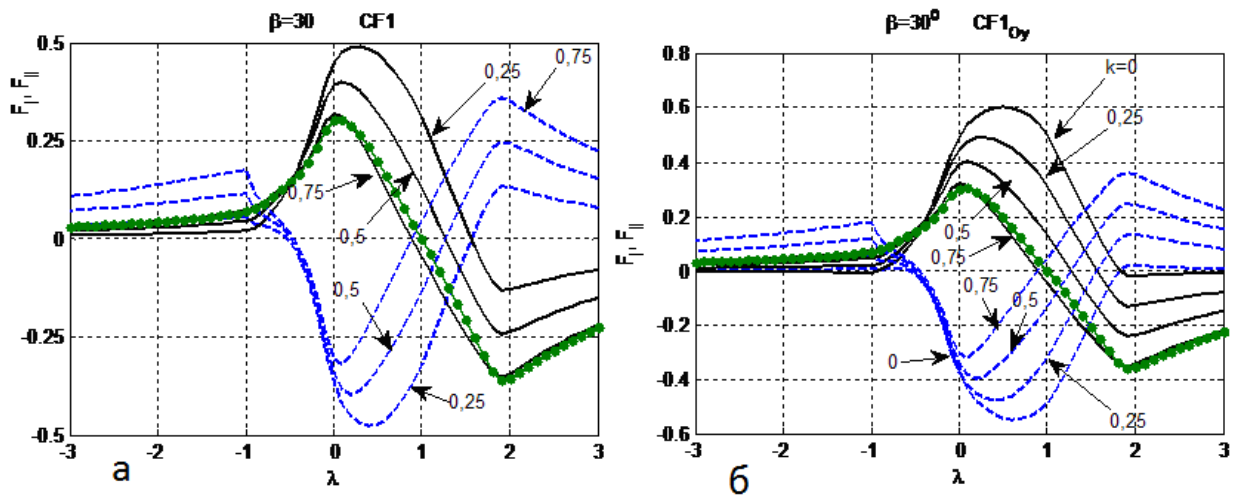


Рис. 5 – Відносні КІН для тріщин, нахилених під кутом  $\beta = 30^\circ$  (матеріал *CFI*)

На основі проведених розрахунків можна зробити наступні висновки: для ізотропних матеріалів КІН  $K_I$  за врахування контакту берегів тріщин не дорівнює нулю тільки тоді, коли штамп розміщений лівіше тріщини, причому він є меншим за величиною, ніж КІН  $K_{II}$ ; КІН  $K_I$  відмінні від нуля для всіх анізотропних матеріалів за врахування контакту берегів тріщин, причому вони є сумірними із КІН  $K_{II}$  для істотно анізотропних матеріалів; КІН є більшими за величиною у випадку, коли жорсткість матеріалу є більшою у вертикальному напрямку; при значеннях коефіцієнта тертя  $k < 0.5$  КІН з достатньою для практики точністю можуть бути розраховані на основі опису тиску під штампом за формулою Герца.

На основі порівняння наведених графіків можна зробити висновок про значний вплив анізотропії на значення КІН, причому зміна напрямку армування істотно змінює як характер, так величину КІН. В більшості із розглянутих випадків мав місце контакт берегів тріщин біля вершини.

**Висновки.** Розроблено методику розрахунку напружень біля штампів і тріщин у анізотропній

півплощині, в якому враховано тертя під штампом та контакт берегів тріщин. Дослідження напружень біля тріщин виконано методом інтегральних рівнянь, причому задачу визначення контактних напружень на берегах тріщин зведено до задачі квадратичного програмування. Виконано аналіз напружень біля тріщин для ізотропного і анізотропних матеріалів залежно від розміщення тріщин відносно штампа. Встановлено вплив анізотропії та напрямків армування матеріалів на величини та характер КІН.

1. Божидарнік В.В. Пружна та гранична рівновага анізотропних пластинок з отворами і тріщинами / В.В. Божидарнік, О.В. Максимович // Луцьк: ЛДТУ, 2003. – 226 с.
2. Галин Л.А. Контактные задачи теории упругости. – М. : Наука, 1980. – 304 с.
3. Дацишин О.П. Напружений стан півплощини з крайовою пологою тріщиною під герцівським навантаженням / О.П. Дацишин, Г.П. Марченко // Фіз.-хім. мех. матеріалів. – 2008. – № 1. – С. 23-34.
4. Дацишин О.П. Особливості контактування берегів крайової тріщини за рухомого герцівського навантаження / О.П. Дацишин, А.Ю. Глазов, А.Б. Левус // Фіз.-хім. мех. матеріалів. – 2013. – № 5. – С. 31-42.
5. Максимович О.В. Розрахунок напруженого стану анізотропних пластинок з отворами і криволінійними тріщинами при врахуванні контакту їхніх берегів // Вісник Тернопільського державного технічного університету. – 2009. – № 3. – С. 36-42.
6. Мухелишвили Н.И. Некоторые основные задачи математической теории упругости. – М. : Наука, 1966. – 708 с.
7. Панасюк В.В. Контактна задача про дію штампа на границю півплощини, послабленої системою криволінійних тріщин / В.В. Панасюк, А.П. Дацишин, Г.П. Марченко // Фіз.-хім. механіка матеріалів. – 1995. – № 6. – С. 7-16.
8. Саврук М.П. Тиск з тертям абсолютно жорсткого штампа на пружний півпростір з тріщинами / М.П. Саврук, А.І. Томчик // Фіз.-хім. механіка матеріалів. – 2010. – № 3. – С. 5-16.
9. Maksymovych O. Doubly periodic cracks in the anisotropic medium with the account of contact of their faces / O. Maksymovych, Ia. Pasternak, H. Sulym, S. Kutsyk // Acta mechanica et automatica. – 2014. – Vol. 8, № 3. – P. 160–164.
10. Maksimovich O.V. Consideration of anisotropy and contact of cracks edge at stress calculations of rolling bearings / O.V. Maksimovich, A.V. P'yushin, A.D. Ivashhuk // Vestnik OrelGAU – December 2014. – Vol. 6, № 51. – P. 80-86.

УДК 519.688:539.3

Пастернак В.В., к.т.н.; Пастернак Я.М., д.ф.-м.н.  
Луцький національний технічний університет

## ОБЪКТНО-ОРИЕНТОВАНА РЕАЛИЗАЦИЯ МЕТОДУ ГРАНИЧНЫХ ЭЛЕМЕНТІВ ТРИВИМІРНОЇ ТЕРМОМАГНІТОЕЛЕКТРОПРУЖНОСТІ

**Пастернак В.В., Пастернак Я.М. Об'єктно-орієнтована реалізація методу граничних елементів тривимірної термомагнітоелектропружності.** Розглянуто принципи побудови об'єктно-орієнтованого сирцевого коду розробленого раніше методу граничних елементів тривимірної термомагнітоелектропружності. Вказано на переваги такої реалізації порівняно із процедурною. Запропоновано під'єднувати розроблені програми як модулі інтерпретованих мов високого рівня для створення потужних комплексів інженерного розрахунку.

**Ключові слова:** метод граничних елементів; сирцевий код; системи інженерних розрахунків

**Пастернак В.В., Пастернак Я.М. Объектно-ориентированная реализация метода граничных элементов трехмерной термомагнитоэлектродупругости.** Рассмотрены принципы построения объектно-ориентированного исходного кода разработанного ранее метода граничных элементов трехмерной термомагнитоэлектродупругости. Указано на преимущества такой реализации по сравнению с процедурной. Предложено подсоединять разработанные программы как модули интерпретированных языков высокого уровня для создания мощных комплексов инженерных расчетов.

**Ключевые слова:** метод граничных элементов; исходный код; системы инженерных расчетов

**Pasternak V.V., Pasternak Ia.M. Object-oriented programming of boundary element method for 3D thermomagnetoelastocity.** The paper considers principles for development of object-oriented source code for 3D boundary element approach obtained earlier for 3D thermomagnetoelastocity. It shows the advantages of the proposed approach comparing to the procedural one. It proposes to link the developed programs as modules for high-level interpreted programming languages for development of powerful computer aided engineering systems.

**Keywords:** boundary element method; source code; computer aided engineering

**Вступ.** На цей час у розробці програмного забезпечення, зокрема й комплексів інженерних розрахунків, паралельно використовується декілька парадигм програмування: процедурне, об'єктно-орієнтоване, шаблонне тощо. Це стосується і програмної реалізації методу граничних елементів (МГЕ). Традиційно тут використовується процедурне програмування мовою FORTRAN, адже розробка МГЕ розпочалася ще у 60-х роках ХХ століття [1], коли такі концепції були практично єдиними. З тих пір опубліковано велику кількість монографій та підручників (напр., [2–4]), де подано сирцевий код низки програм аналізу двовимірних та тривимірних задач теплопровідності, пружності та термодупругості тіл із отворами, глобулярними включеннями та тріщинами.

Проте в два останні десятиліття при реалізації нових концепцій МГЕ все частіше використовується об'єктно-орієнтований підхід [5–7], що дає можливість інкапсулювати дані, а завдяки механізмам успадковування та поліморфізму є потужним засобом повторного використання коду та створення бібліотек для інтерпретованих мов високого рівня. Однак подані на даний час концепції програмної реалізації МГЕ є радше прикладами використання об'єктного підходу без орієнтації на його прикладне застосування у системах інженерних розрахунків та проектування (CAE – Computer Aided Engineering).

Тому у цій роботі здійснено опис основних етапів програмної реалізації раніше розробленого методу граничних елементів тривимірної термомагнітоелектропружності структурно-неоднорідних тіл [8], що дає можливість гнучкої розширюваності коду та його поєднання із готовими модулями для створення системи інженерних розрахунків.

**Основні співвідношення та обчислювальна схема МГЕ [8].** Узагальнені у [8] визначальні співвідношення лінійних стаціонарних термомагнітоелектропружності та теплопровідності мають такий вигляд:

$$\tilde{\sigma}_{ij,j} + \tilde{f}_i = 0, \quad h_{i,i} - f_h = 0; \quad (1)$$

$$\tilde{\sigma}_{ij} = \tilde{C}_{ijkm} \tilde{u}_{k,m} - \tilde{\beta}_{ij} \theta, \quad h_i = -k_{ij} \theta_{,j}, \quad (2)$$

де

$$\begin{aligned}
\tilde{u}_i &= u_i, \quad \tilde{u}_4 = \phi, \quad \tilde{u}_5 = \psi; \quad \tilde{f}_i = f_i, \quad \tilde{f}_4 = -q, \quad \tilde{f}_5 = b_m; \\
\tilde{\sigma}_{ij} &= \sigma_{ij}, \quad \tilde{\sigma}_{4j} = D_j, \quad \tilde{\sigma}_{5j} = B_j; \\
\tilde{C}_{ijkm} &= C_{ijkm}, \quad \tilde{C}_{ij4m} = e_{mij}, \quad \tilde{C}_{4jkm} = e_{jkm}, \quad \tilde{C}_{4j4m} = -\kappa_{jm}, \\
\tilde{C}_{ij5m} &= h_{mij}, \quad \tilde{C}_{5jkm} = h_{jkm}, \quad \tilde{C}_{5j5m} = -\mu_{jm}, \\
\tilde{C}_{4j5m} &= -\gamma_{jm}, \quad \tilde{C}_{5j4m} = -\gamma_{jm}; \\
\tilde{\beta}_{ij} &= \beta_{ij}, \quad \tilde{\beta}_{4j} = -\chi_j, \quad \tilde{\beta}_{5j} = \nu_j \quad (i, j, k, m = 1, 2, 3).
\end{aligned} \tag{3}$$

Тут  $\sigma_{ij}$  – компоненти тензора напружень;  $h_i$  – компоненти вектора густини теплового потоку;  $D_i$  – електричне зміщення;  $B_i$  – індукція магнітного поля;  $f_i$  – об'ємні сили;  $q$  – густина вільних зарядів;  $f_h$  – густина розподілених джерел (стоків) тепла;  $b_m$  – об'ємна густина постійного електричного струму, яка для діелектрика є нульовою;  $u_i$  – переміщення точок тіла;  $\phi$  – електричний потенціал;  $\psi$  – потенціал стаціонарного магнітного поля;  $\theta$  – зміна температури порівняно з відліковою;  $C_{ijkm}$  – пружні сталі;  $k_{ij}$  – коефіцієнти теплопровідності;  $\beta_{ij}$  – модулі теплового розширення (коефіцієнти теплових напружень);  $e_{ijk}$  – п'єзоелектричні сталі;  $\kappa_{ij}$  – діелектричні сталі матеріалу;  $h_{ijk}$  – п'єземагнітні сталі;  $\mu_{ij}$ ,  $\gamma_{ij}$  – магнітна та електромагнітна проникності матеріалу;  $\chi_i$  – піроелектричні коефіцієнти;  $\nu_i$  – піромагнітні коефіцієнти. Тензори з компонентами  $C_{ijkm}$ ,  $k_{ij}$ ,  $\kappa_{ij}$ ,  $\mu_{ij}$ ,  $\gamma_{ij}$  та  $\beta_{ij}$  вважаються симетричними. У формулах прийняте правило Айнштейна підсумовування за індексом, що повторюється. Кома в індексах означає операцію диференціювання за координатою, індекс якої стоїть після коми, тобто,  $u_{i,j} \equiv \partial u_i / \partial x_j$ . Великі індекси змінюються від 1 до 5, а маленькі від 1 до 3, тобто  $I = 1, 2, \dots, 5$ ,  $i = 1, 2, 3$ .

Відповідні їм крайові інтегральні рівняння теплопровідності та термомагнітоелектропружності отримані у формі

$$\begin{aligned}
\frac{1}{2}\theta(\mathbf{x}_0) &= \iint_{\partial\mathbf{B}} \Theta^*(\mathbf{x}, \mathbf{x}_0) h_n(\mathbf{x}) dS(\mathbf{x}) - \text{CPV} \iint_{\partial\mathbf{B}} H^*(\mathbf{x}, \mathbf{x}_0) \theta(\mathbf{x}) dS(\mathbf{x}) \\
&\quad - \iiint_{\mathbf{B}} \Theta^*(\mathbf{x}, \mathbf{x}_0) f_h(\mathbf{x}) dV(\mathbf{x}),
\end{aligned} \tag{4}$$

$$\begin{aligned}
\frac{1}{2}\tilde{u}_I(\mathbf{x}_0) &= \iint_{\partial\mathbf{B}} U_{II}(\mathbf{x}, \mathbf{x}_0) \tilde{t}_J(\mathbf{x}) dS(\mathbf{x}) - \text{CPV} \iint_{\partial\mathbf{B}} T_{II}(\mathbf{x}, \mathbf{x}_0) \tilde{u}_J(\mathbf{x}) dS(\mathbf{x}) \\
&\quad + \iint_{\partial\mathbf{B}} [R_I(\mathbf{x}, \mathbf{x}_0) \theta(\mathbf{x}) + V_I(\mathbf{x}, \mathbf{x}_0) h_n(\mathbf{x})] dS(\mathbf{x}) \\
&\quad + \iiint_{\mathbf{B}} U_{IJ}(\mathbf{x}, \mathbf{x}_0) \tilde{f}_J(\mathbf{x}) dV(\mathbf{x}) - \iiint_{\mathbf{B}} V_I(\mathbf{x}, \mathbf{x}_0) f_h(\mathbf{x}) dV(\mathbf{x}),
\end{aligned} \tag{5}$$

де  $\mathbf{x}_0 \in \partial\mathbf{B}$ , а межа  $\partial\mathbf{B}$  тіла  $\mathbf{B}$  є вважається гладкою у точці колокації  $\mathbf{x}_0$ .

Для розв'язування інтегральних рівнянь (4), (5) у [8] запропоновано такий алгоритм МГЕ:

1. межа тіла розбивається на розривні квадратичні чотиристоронні елементи;
2. для кожного елемента задаються функції форми для інтерполяції крайових функцій за вузловими значеннями, зокрема, для крайових елементів тріщин чи клинів останні враховують степеневу особливість розв'язку;
3. здійснюється числове інтегрування отриманих після перших двох кроків означених поверхневих інтегралів із застосуванням нелінійних відображень та спеціальних квадратур для побудови системи лінійних алгебричних рівнянь стосовно невідомих вузлових значень крайових функцій;

4. розв'язується отримана система рівнянь та визначається термомагнітоелектронапружений стан у бажаних точках тіла, а також коефіцієнти інтенсивності фізико-механічних полів у точках особливої геометрії.

Перші два кроки пов'язані з геометрією задачі та крайовими умовами, тому їх доцільно інкапсулювати в один абстрактний тип даних, що описуватиме граничні елементи. Інші два кроки пов'язані безпосередньо із розв'язуванням задачі, тому в об'єктній моделі вони віднесені до іншого класу.

**Об'єктно-орієнтована реалізація МГЕ термомагнітоелектропружності.** Для чотиристоронніх граничних елементів створено клас, декларативну частину якого подано нижче.

```
class BoundaryElement
{
private:
//Вузли топології елемента звичайні, нерозривні
    double NodesX[3][3],
           NodesY[3][3],
           NodesZ[3][3];

    double Lagr1DTopol(int p, double t);
    double DiffLagr1DTopol(int p, double t);

public:
//Вузли задання крайових умов розривні
    double tP[5][3][3],
           tM[5][3][3],
           uP[5][3][3],
           uM[5][3][3],
           thetaP[3][3],
           thetaM[3][3],
           hnP[3][3],
           hnM[3][3];

    BEType ElementType;
    HTType ElementHTType;

    BoundaryElement(void);
    BoundaryElement(double X[3][3], double Y[3][3], double Z[3][3]);
    void SetDefaultShapeFunctions(void);
    void SetShapeFunctions(int DispShFn, int TracShFn,
                           int TempShFn, int HeatShFn);
    void SetElement(double X[3][3], double Y[3][3], double Z[3][3]);
    void Position(double xi, double eta, double x[3]);
    void DiffR(double xi, double eta, double dr_dxi[3], double dr_deta[3]);
    void Normal(double xi, double eta, double n[3], double &J);
    void RotateElement(double xC[3], double phi_x,
                      double phi_y, double phi_z);
    void TranslateElement(double x, double y, double z);

    double ShapeFunctionDisp(const int n, const double xi, const double eta);
    double ShapeFunctionTrac(const int n, const double xi, const double eta);
    double ShapeFunctionTemp(const int n, const double xi, const double eta);
    double ShapeFunctionHeat(const int n, const double xi, const double eta);
};
```

Як видно з лістингу, приватна частина класу містить три 3x3-масиви, що містять абсциси, ординати та аплікати 9 вузлів граничного елемента. Також клас має приватні методи `Lagr1DTopol` та `DiffLagr1DTopol` що задають одновимірні інтерполяційні поліноми Лагранжа та їх похідні, які використовуються для апроксимації елемента поверхнею другого порядку

$$\mathbf{x}(\xi, \eta) = \sum \mathbf{x}_{ij} L_i(\xi) L_j(\eta). \quad (6)$$

Відкрита частина класу містить масиви для зберігання крайових умов у вузлах інтерполяції ( $t_P$ ,  $t_M$ ,  $u_P$ ,  $u_M$ ,  $\theta_P$ ,  $\theta_M$ ,  $h_P$ ,  $h_M$ ), а також змінні, що відображають які з умов є заданими (`ElementType`, `ElementHTType`). Віднесення цих змінних до відкритої частини класу хоч і порушує «хороший тон» об'єктного програмування, проте є зручним та зменшує код програми.

Окрім створених методів задання вузлів елемента (`SetElement`) та функцій форми (`SetShapeFunctions`) зручними у користуванні є методи `RotateElement`, `TranslateElement`, що дають можливість повертання та паралельного перенесення граничного елемента. Зокрема, для створення зображеної на рис. 1 сітки дископодібної тріщини достатньо задати вузли 1 внутрішнього та 1 зовнішнього елемента, а потім шляхом копіювання і повертання створити решту елементів.

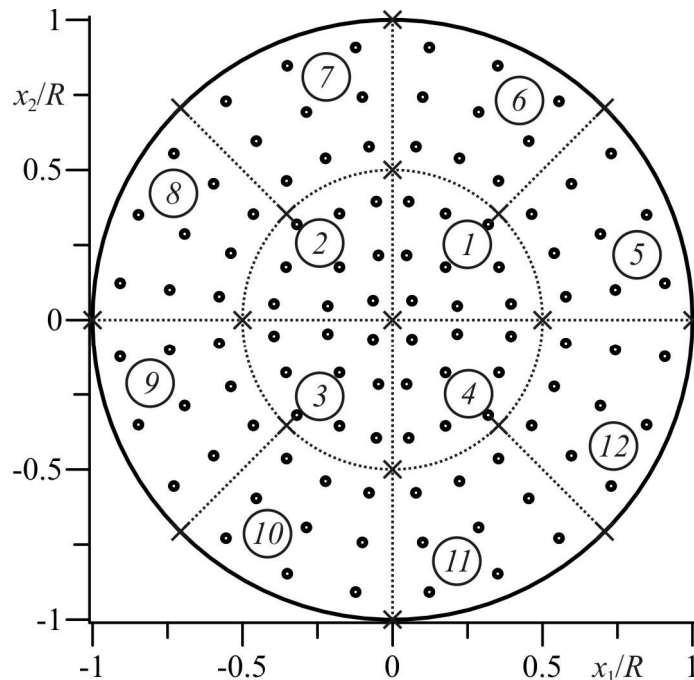


Рис. 1. Сітка розбиття для дископодібної тріщини

Граничноелементні методи розв'язування інтегральних рівнянь (4), (5) інкапсульовані в класі `ThermoSolid3D`, лістинг якого подано нижче (для компактності викладу не подано захищених методів обчислення ядер рівнянь та їхнього інтегрування).

```
class ThermoSolid3D
{
protected:
    double CC[5][3][5][3],
           beta[5][3],
           kk[3][3];

    double SS[5][3][5][3],
           alpha[5][3];

    BoundaryElement *Elements;
    int NElements;

    WinklerInclusion3D *WinklerInclusions;
    int NWinklerInclusions;

    int NForces;
    ConcFactor *Forces;

    int NHeatSources;
```

```
HeatSource *HeatSources;

double ThetaInf;
double SigmaInf[5][3], UI_j_Inf[5][3];

public:
    ThermoSolid3D(double C1[5][3][5][3],
                 double beta1[5][3],
                 double k1[3][3]);
    ~ThermoSolid3D();
    void AllocateElements(const int Number);
    void SetBEGeometry(const int Elem, double X[3][3],
                     double Y[3][3], double Z[3][3]);
    void RotateBE(const int Elem, double xC[3],
                 double phi_x, double phi_y, double phi_z);
    void TranslateBE(const int Elem, double x, double y, double z);
    void SetBEType(const int Elem, const int type=1, const int HTtype=1);
    void SetBEConditions(const int Elem,
                        double HT_BCp[3][3], double HT_Bcm[3][3],
                        double MEE_BCp[5][3][3],
                        double MEE_Bcm[5][3][3]);

    void BEMSolver(void);
    void HTTemperature(double x0[3], double &theta);
    void HTHeatFlux(double x0[3], double h[3]);
    void Displacement(double x0[3], double u[5]);
    void Stress(double x0[3], double sigma[5][3]);
    void FIF(int Element, int flag, double x,
            double x0[3], double KK[5], double Kt[2]);
    void SetShapeFunctions(int Element, int DispShFn,
                          int TracShFn, int TempShFn, int HeatShFn);
    void SetSigmaInf(double sigma[5][3]);
    void SetThetaInf(const double theta);
    void AddCForce(ConcFactor Force);
    void DeleteCForces(void);
    void AddHeatSource(HeatSource Source);
    void DeleteHeatSources(void);
};
```

Закрита частина класу містить дані про фізичні властивості матеріалу тіла, вказівник на масив граничних елементів, пам'ять під який виділяється динамічно методом `AllocateElements`. Також захищена частина класу містить динамічні масиви, що задають розташування та інтенсивність зосереджених джерел тепла та розширених сил, навантаження на безмежності тощо.

Методи `SetBEGeometry`, `RotateBE`, `TranslateBE`, `SetBEType`, `SetBEConditions`, `SetShapeFunctions` класу у свою чергу викликають відповідні методи класу граничних елементів для маніпуляцій з останніми.

Метод `BEMSolver` є реалізацією запропонованої у [8] схеми граничноелементного розв'язування інтегральних рівнянь (4), (5). Після цього методи `HTTemperature`, `HTHeatFlux`, `Displacement`, `Stress` дають можливість обчислити температуру, тепловий потік, розширені переміщення та напруження у довільній точці тіла. Крім цього, метод `FIF` дає можливість визначати коефіцієнти інтенсивності фізико-механічних полів на фронті тріщин та тонких податних слабо проникних включень.

За допомогою програми `SWIG` [9] отриманий об'єктний код інтегровано до середовища інтерпретованої мови високого рівня `Python`. Завдяки цьому створено комплекс інженерних розрахунків на основі побудованої бібліотеки та додатків `NumPy`, `SciPy`, `matplotlib`.

**Висновки.** Завдяки об'єктно-орієнтованій структурі створений сирцевий код методу граничних елементів тривимірної термомагнітоелектропружності структурно неоднорідних тіл дає можливість його поєднання з розвиненими багатомодульними програмними комплексами інтерпретованих мов високого рівня, завдяки чому можна створювати потужні комплекси інженерних розрахунків. Окрім цього, із

використанням механізму успадковування розроблений код можна легко поширювати і на інші класи задач, а саме для півпростору чи біматеріалу, що забезпечує ефективне повторне використання коду та гнучкість моделі.

1. Cheng A.H.D. Heritage and early history of the boundary element method / A.H.D. Cheng, D.T. Cheng // *Engineering Analysis with Boundary Elements*. – 2005. – **29**, 3. – P. 268–302.
2. Бенерджи П. Методы граничных элементов в прикладных науках / П. Бенерджи, Р. Батгерфилд. – М.: Мир, 1984. – 494 с.
3. Dominguez J. Boundary elements in dynamics / J. Dominguez. – Boston: Computational Mechanics Publications, 1993. – 700 p.
4. Beer G. The boundary element method with programming / G. Beer, I. Smith, C. Duenser. – New York: Springer, 2008. – 496 p.
5. Lage C. The application of object-oriented methods to boundary elements / C. Lage // *Comput. Methods Appl. Mech. Engrg.* – 1998. – **157**. – P. 205–213.
6. Qiao H. Object-oriented programming for the boundary element method in two-5al heat transfer analysis / H. Qiao // *Advances in Engineering Software*. – 2006. – **37**. – P. 248–259.
7. Wang W. Object-oriented programming in boundary element methods using C++ / W. Wang, X. Ji, Y. Wang // *Advances in Engineering Software*. – 1999. – **30**. – P. 127–132.
8. Pasternak Ia. A comprehensive study on Green's functions and boundary integral equations for 3D anisotropic thermomagnetoelctroelasticity / Ia. Pasternak, R. Pasternak, H. Sulym // *Engineering Analysis with Boundary Elements*. – 2016. – **64**. – P. 222–229.
9. <http://www.swig.org/>



УДК 629.113

Решетило О.М., Павлік П.В., Токарчук В.В.  
Луцький національний технічний університет

## ПІДВИЩЕННЯ ПОТУЖНОСТІ ТА ЗМЕНШЕННЯ ШКІДЛИВИХ ВИКИДІВ ДИЗЕЛЬНОГО ДВИГУНА ВНУТРІШНЬОГО ЗГОРАННЯ АВТОМОБІЛЯ

**Решетило О.М., Павлік П.В., Токарчук В.В. Підвищення потужності та зменшення шкідливих викидів дизельного двигуна внутрішнього згорання автомобіля.** На базі плати Arduino Uno розроблена автоматизована система керування якістю повітря, що подається в паливну систему дизельного двигуна внутрішнього згорання автомобіля, що дозволить збільшити потужність двигуна, зменшить викиди вуглекислого газу та розхід палива. Це досягається за рахунок подачі повітря в двигун за допомогою турбіни з електричним приводом, його іонізації та розпилення води.

**Ключові слова:** двигун внутрішнього згорання, електродвигун, турбіна, іонізатор повітря, насос, Arduino.

**Reshetylo O.M., Pavlik P.V., Tokarchuk V.V. Increased capacity and reduce emissions of diesel internal combustion engine car.** Based on Arduino Uno board developed the automated control system of quality of air supplied to the fuel system of a diesel internal combustion engine car which will increase engine power, reduce emissions of carbon dioxide and consumption of fuel. This is achieved by supplying air to the engine using a turbine with electric drive, its ionization and spraying water.

**Keywords:** internal combustion engine, electric drive, turbine, ionizer air, pump, Arduino.

**Решетило А.Н., Павлік П.В., Токарчук В.В. Повышение мощности и снижение вредных выбросов дизельного двигателя внутреннего сгорания автомобиля.** На базе платы Arduino Uno разработана автоматизированная система управления качеством воздуха, который подается в топливную систему дизельного двигателя внутреннего сгорания автомобиля, что обеспечит увеличение мощности двигателя, уменьшит выбросы углекислого газа та расход топлива. Это достигается за счет подачи воздуха в двигатель при помощи турбины с электрическим приводом, его ионизации и распыления воды.

**Ключевые слова:** двигатель внутреннего сгорания, электродвигатель, турбина, ионизатор воздуха, насос, Arduino.

**Вступ.** На сьогоднішній день в Україні кількість автомобілів зростає з року в рік. Переважна більшість це автомобілі з Європи з двигунами внутрішнього згорання, що не відповідає сучасному стандарту «Євро 5». Парк автомобілів з дизельними двигунами, які не обладнані навіть турбінами, займає значну його частину. Вони чинять значний вплив на забруднення навколишнього середовища, оскільки мають значну витрату палива, яке повністю не згорає під час роботи двигуна. При цьому вони споживають значну кількість палива, оскільки мають менший к.к.д., навіть в порівнянні з дизельними двигунами, що обладнані турбіною.

**Постановка проблеми.** Для підвищення екологічності старого автомобільного парку з дизельними двигунами, по-перше необхідно зменшити кількість шкідливих викидів вуглекислого газу, висококонцентрованих токсичних компонентів та тяжких металів, що є основними забруднювачами атмосфери. По-друге, із зростанням цін на паливо та державною програмою із зменшення витрати енергоносіїв, доцільно вдосконалити паливну систему, яка б зменшила витрату палива за рахунок підвищення потужності дизельного двигуна внутрішнього згорання.

**Метою даної роботи** є розробка автоматизованої системи керування якістю та кількістю подачі повітря в дизельний двигун внутрішнього згорання автомобіля.

### **Основні результати дослідження.**

Як відомо, чим більше повітря надходить у циліндри двигуна внутрішнього згорання, тим краще згорає паливо і підвищується його потужність, тому двигун з турбіною значно економічний і потужніший аналогічного атмосферного двигуна.

Турбіни, що встановлюються на автомобілі, бувають двох типів:

- турбокомпресор з ротором, що приводиться в рух за допомогою відпрацьованих газів. В цій системі подача масла в турбіну здійснюється із загального контуру системи змащення двигуна, тому турбіну для дизельного двигуна встановити досить складно.

- турбокомпресор з ротором, що приводиться в рух за допомогою пасового приводу. В цій системі турбіна в рух приводиться за допомогою пасу від колінчастого валу і система примусового її змащення відсутня, але принцип нагнітання повітря в камеру згорання залишається аналогічним. За рахунок приводного пасу, турбіна починає обертатися навіть на малих обертах двигуна, створюючи тиск у камері згорання, поступово розкручуючись зі збільшенням обертів.

У дизельний має надходити більше кисню ніж в бензиновий, однак під час роботи повітря стискається, температура його підвищується, оскільки підвищується тиск, тому починається швидке руйнування кисню. Дизель повинен отримувати охолодження, щоб не пошкодилися поршні, кільця

або головки блоку. Також тепле повітря призводить до зниження потужності двигуна.

На сьогоднішній день є дизельні двигуни які окрім турбіни обладнані інтеркулером. Однією з головних функцій якого є зниження температури стисненого повітря. Також інтеркулер дозволяє зменшити детонацію, яка виникає коли процес згоряння стає нестабільним. Тривалий наддув двигуна неможливий, якщо не буде встановлений інтеркулер.

Крім того слід враховувати, що у всіх дизелів за рахунок внутрішнього сумішоутворення для досягнення повноти згорання палива слід забезпечити досить значний надлишок повітря.

Альтернативою системи з турбіною та інтеркулером може бути використання турбіни з приводом від електродвигуна, що пов'язаний з педалью акселератора автомобіля за допомогою реостата, а охолодження повітря в даній системі буде відбуватися за рахунок вприскування і розпилення води у впускний колектор подачі повітря в двигун. Крім підвищення потужності турбонадув забезпечує економію палива в розрахунку на одиницю потужності і зниження токсичності відпрацьованих газів за рахунок більш повного згорання палива. Подача води в двигун, також забезпечує збільшує потужність двигуна, зменшує витрату пального та викиди вуглекислого газу та тяжких металів в атмосферу.

Для подачі та розпилення води у впускному колекторі двигуна використовується спеціально розроблена форсунка та бачок з насосом від системи омивання скла автомобіля. Крім того підвищення потужності двигуна можливе при структуруванні повітря за допомогою іонізатора.

Окрім цього, для очищення від механічних домішок та пилу повітря, що подається в дизельний двигун використовуються паперовий або ж вугільний фільтр, пропускна здатність якого зменшується при зростанні часу експлуатації двигуна після його заміни, що негативно вплине на кількість повітря, що подається у впускний колектор, і роботу самої турбіни. Тому замість нього використовується повітряний фільтр нульового опору.

Принцип роботи розробленої системи наступний. Під час роботи двигуна повітря, що надходить до турбіни з атмосфери, очищується фільтром нульового опору. Турбіна вмикається лише при досягненні колінчастим валом заданих оборотів. Повітря проходячи по впускному колекторі буде змішуватися з водою, що розпилюється форсункою. Далі повітряна суміш надходить безпосередньо в циліндри двигуна внутрішнього згорання.

Іонізатор-активатор кисню повітря (рис. 1) призначений для якісного покращення характеристик паливно-повітряної суміші в камері згорання бензинових або дизельних двигунів внутрішнього згорання, шляхом активації кисню повітря високою напругою промислових частот без виникнення електричного розряду навколо електрода-активатора.



Рис. 1. Іонізатор-активатор кисню повітря

За допомогою датчика моделі BMP180 (рис. 2) вимірюється тиск та температура повітря у впускному колекторі.

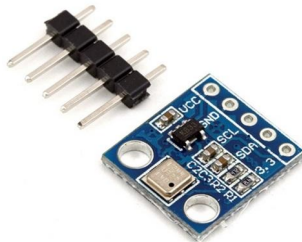


Рис. 2. Датчик тиску та температури BMP180

За допомогою датчика Хола моделі Hall KY-003 (рис. 3) визначимо кількість обертів за хвилину колінчастого валу.



Рис. 3. Датчик Хола Hall KY-003

Для вимірювання витрати води, що подається до форсунки, використовується витратомір моделі YF-S401 (рис. 4).



Рис. 4. Витратомір YF-S401

Чотирьохканальний модуль реле (рис. 5) використовується для управління приладами, що працюють з напругою AC 220 В або DC 30 В за допомогою Arduino.



Рис. 5. Модуль реле чотирьохканальний

Регулювання та керування розробленою системою здійснює мікроконтролер на базі плати Arduino Uno (рис. 6).

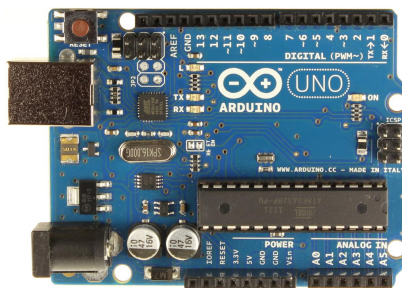


Рис. 6. Плата Arduino Uno

Живлення електричної турбіни, насоса подачі води та плати Arduino буде здійснюватись безпосередньо від акумулятора автомобіля через змінний резистор, опір якого змінюється залежно від положення акселератора педалі газу автомобіля.

На рис. 7 наведено електричну схему, а на рис. 8 – макетну схему підключення елементів системи до плати Arduino Uno.

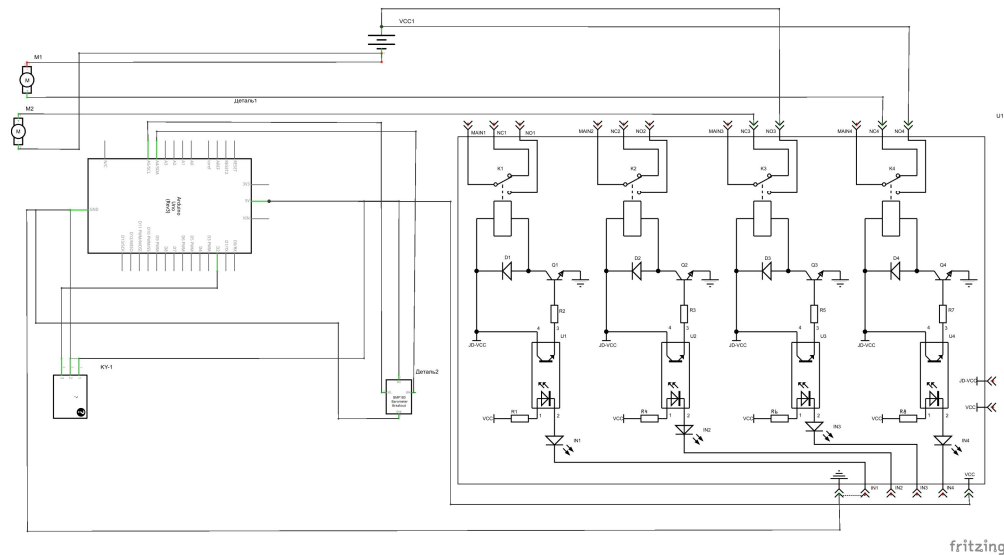


Рис. 7. Електрична схема підключення датчиків та виконавчих механізмів до плати Arduino Uno

Живлення плати Arduino Uno напряму підключаємо до акумулятора автомобіля. Датчики та реле підключаємо до плати Arduino. Датчик BMP 180 підключаємо до живлення 5V плати, «-» датчика підключаємо до «землі» (GND) та виходи SDA та SCL на відповідні входи плати Arduino. Аналогічно підключаємо датчик холода KY-003, тільки сигнал подаємо на дискретний вхід 2. Також підключаємо витратомір води до живлення, GND і сигнал подаємо на дискретний вихід 8. Реле підключаємо відповідно до датчиків. Електричний двигун підключаємо послідовно в реле на розімкнутий контур реле (так як реле має один замкнутий контур, а другий розімкнутий). GND підключаємо на акумулятор, сигнал також подаємо на дискретний вихід плати Arduino. За таким принципом підключаємо і насос подачі води.

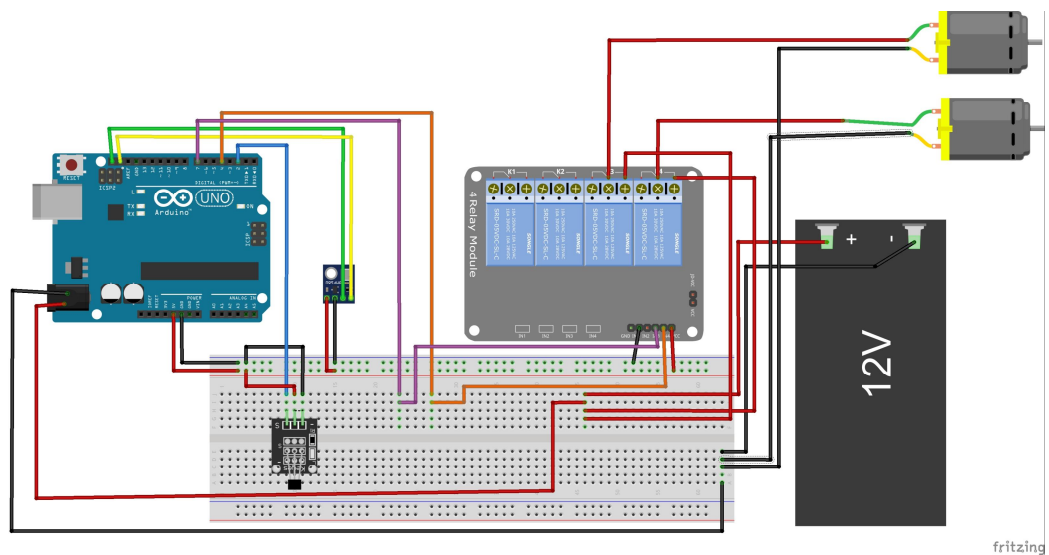


Рис. 8. Макетна схема підключення елементів системи до плати Arduino Uno

В середовищі Arduino 1.5.6-r2 написано скетч, яким записано в плату Arduino.

Зміни температури повітря залежно від тиску повітря при різних ступенях ефективності турбіни наведена на рис. 9.

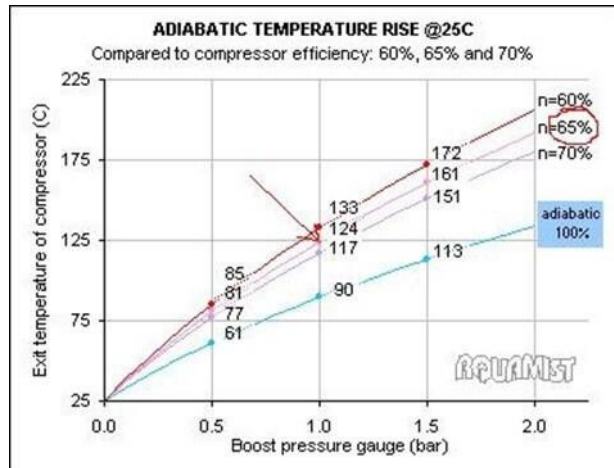


Рис. 9. Графік залежності температури повітря від тиску при різних ступенях ефективності турбіни

У першому тесті в повітряну суміш розпилювалось 100 мл./хв. води. Результати тесту наведені на рис. 10.

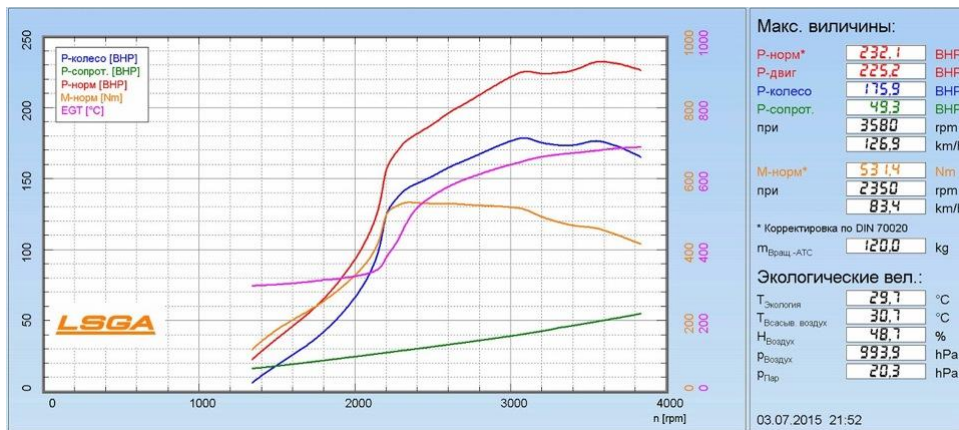


Рис. 10. Результати першого тесту

Потужність двигуна становить 232 к.с., крутний момент 531 Н·м, максимальне його значення склало 685°. Тобто, є великий резерв для збільшення потужності в безпечному режимі.

В другому тесті збільшили подачу води до 240 мл./хв. (при повному газі на максимальній потужності). Результати тесту наведені на рис. 11.

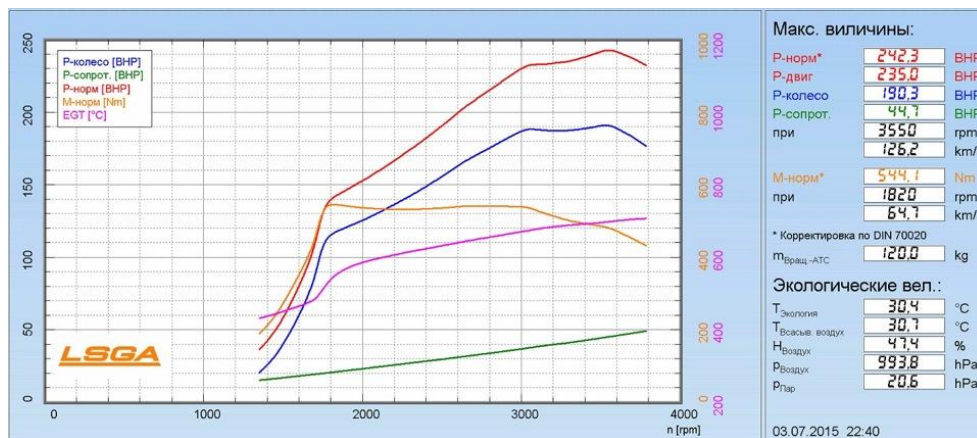


Рис. 11. Результати другого тесту

В третьому тесті використовували 50% води та 50% метанолу. Результати тесту наведені на



рис. 12.

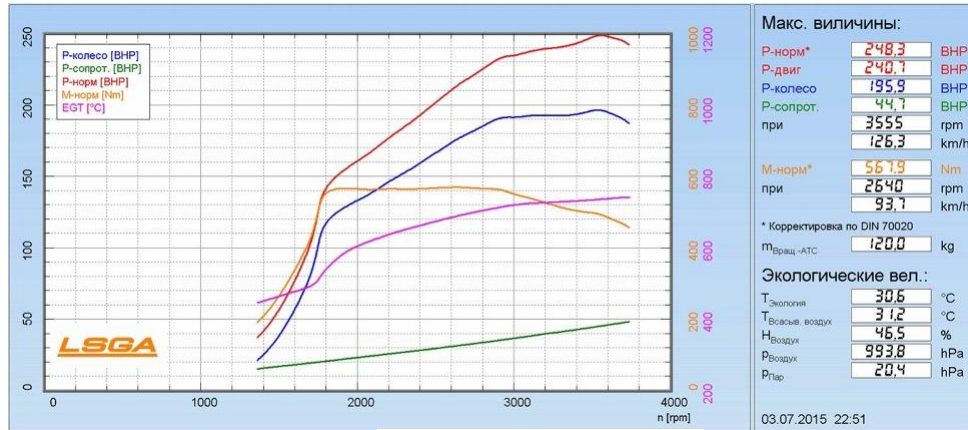


Рис. 12. Результати третього тесту

Метанол - це паливо, і природно містить енергію, на відміну від води. Відповідно зросла не тільки потужність двигуна до 248 к.с., а й крутний момент склав 568 Н·м, але і значно піднялася температура ЕГТ (740 °С).

Однак, використання метанолу, як засобу для підвищення потужності в дизельних двигунах, не є доцільним. Додавання більше 50% метанолу, може призвести до детонації, тому краще просто збільшити подачу палива шляхом класичного "чіп тюнінга". А ось вприскування води відкриває нові можливості і сильно розширює межі безпечного збільшення крутного моменту і максимальної потужності двигуна. Виняток становить зимовий час, коли при значному часі перебування автомобіля на морозі при непрацюючому двигуні вода замерзає, і час нагріву дизельного двигуна значно більший ніж бензинового, необхідно постійно підігрівати воду, або ж додавати, як мінімум 20% метанолу.

Масло в двигуні внутрішнього згорання змащує поверхні тертя та очищує їх від різних відкладень. Вприскування води також виконує функцію очищення поверхні циліндрів та поршнів, тому масло буде служити довше.

**Висновки.** Використання розробленої автоматизованої системи керування якістю повітря, що подається в паливну систему дизельного двигуна внутрішнього згорання автомобіля, збільшить потужність двигуна до 20%, зменшить викиди вуглекислого газу до 25% та зменшить розхід палива на 10...15%.

1. <http://uk.wikipedia.org/wiki/>

2. В.Ф. Кисликов, В.В. Лушчик, Будова й експлуатація автомобілів. – Київ: 2009.

3. В.М. Складаров, В.П. Волков, Р.М. Кузнецов, М.В. Складаров, В.В. Стельмашук, Конструкція автомобіля. Частина перша: " Особливості конструкції двигунів". – Харківський національний автомобільно-дорожній університет, Луцький національний технічний університет, 2012.

УДК 510.589+681.587.72

Доц., к.т.н. Чабан О.В.\*, ас., к.т.н. Костючко С.М., ас. Міскевич О.І.\*\*\*, студ. Киричук А.А.\*\*\*

\*Національний університет «Львівська політехніка»

\*\*Луцький національний технічний університет

\*\*\*Національний педагогічний університет імені М.П. Драгоманова

## ВЕРИФІКАЦІЯ НА ОСНОВІ ЧИСЛОВОГО ЕКСПЕРИМЕНТУ ДВОХ МЕТОДІВ ДОСЛІДЖЕННЯ ПАРАМЕТРИЧНОЇ ЧУТЛИВОСТІ

**Чабан О.В., Костючко С.М., Міскевич О.І., Киричук А.А. Верифікація на основі числового експерименту двох методів дослідження параметричної чутливості.** Теоретичні та практичні доробки теорії аналізу параметричної чутливості є застосовними лише до найпростіших задач. Вони не дають змоги диференціювати рівняння стану за вектором невідомих у складних нелінійних системах. Метод допоміжної параметричної чутливості, в свою чергу, дозволяє обійти цю процедуру методом введення допоміжних рівнянь стану, які відіграють важливу роль при виведенні рівнянь чутливості. Матриця параметричних чутливостей обчислюється, як добуток матриці коефіцієнтів диференціальних рівнянь стану, записаних у нормальній формі Коші, та матрицю допоміжних параметричних чутливостей. В даній статті продемонстровано верифікацію класичного методу та методу допоміжної параметричної чутливості.

**Ключові слова:** нелінійна система, параметрична чутливість, нелінійність, допоміжна модель, алгоритм.

**Чабан О.В., Костючко С.Н., Міскевич О.І., Киричук А.А. Верификация на основе численного эксперимента двух методов исследования параметрической чувствительности.** Теоретические и практические наработки теории анализа параметрической чувствительности является приемлемым только к простейшим задачам. Они не позволяют дифференцировать уравнения состояния по вектору неизвестных в сложных нелинейных системах. Метод вспомогательной параметрической чувствительности, в свою очередь, позволяет обойти эту процедуру методом введения вспомогательных уравнений состояния, которые играют важную роль при выводе уравнений чувствительности. Матрица параметрических чувствительностей исчисляется как произведения матрицы коэффициентов дифференциальных уравнений состояния, записанных в нормальной форме Коши, и матрицы вспомогательных параметрических чувствительности. В данной статье продемонстрировано верификацию классического метода и метода вспомогательной параметрической чувствительности.

**Ключевые слова:** нелинейная система, параметрическая чувствительность, нелинейность, вспомогательная модель, алгоритм.

**Tchaban O.V., Kostiuchko S.M., Miskevich O.I., Kyrychuk A.A. The verification of two methods research of parametric sensitivity using numerical experiment.** Theory and practice of analysis of parametric sensitivity are applied only to the simplest problems. They do not allow to differentiation equation of state by the vector of unknowns in complex nonlinear systems. The method of auxiliary parametric sensitivity allows bypassing this procedure by the introduction of auxiliary equations of state, which play a supportive role in finding equations of sensitivity. Matrix of parametric sensitivities is calculated as the product of matrix of coefficient differential equations of state, which are written in the normal Cauchy's form, and matrix of auxiliary parametric sensitivity. In this article is demonstrated verification of the classical method and method of auxiliary parametric sensitivity.

**Keywords:** non-linear systems, parametric sensitivity, non-linearity, auxiliary model, algorithm.

**Постановка наукової проблеми.** У наш час досить великою є потреба розвитку методів математичного моделювання, які б давали змогу обчислювати параметричну чутливість. Це важливо як для теорії, так і для практики проектування та експлуатації нелінійних технічних систем. Значний вплив на розв'язок задач оптимального проектування відіграють сталі параметри системи, адже вони впливають на її характеристики.

**Аналіз досліджень.** Більшість сучасних розробок, при дослідженні параметричної чутливості нелінійних систем, базуються використанні загальної теорії звичайних диференціальних рівнянь. Практичний аспект цього застосування має трудність побудови диференціальних рівнянь чутливості. Через це класичні підходи для реальних задач дають неприйнятні результати. Це звужує коло застосування класичної теорії до найпростіших задач. В математичному моделюванні задача аналізу усталеної параметричної чутливості нелінійних фізичних систем ще не знайшла свого належного розв'язання. Як кінцевий результат пропонується підміна понять залежності тих чи інших невідомих від дискретних змін тих чи інших параметрів.

**Виклад основного матеріалу й обґрунтування результатів досліджень.** Застосування класичної теорії параметричної чутливості до складних нелінійних систем неможливе. Метод допоміжної параметричної чутливості знімає обмеження класичної теорії та ґрунтується на ідеї використання допоміжних рівнянь стану. Основою цього методу є загальна теорія нелінійних диференціальних рівнянь. Продемонструємо на простому прикладі на скільки спрощується аналіз на підставі запропонованого методу.

Диференціальне рівняння стану пристрою запишемо в загальному вигляді

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}_1(\mathbf{x}, \boldsymbol{\lambda}, t), \quad \mathbf{x}(t)|_{t=0} = \mathbf{x}(0), \quad (1)$$

$\mathbf{x}$  – колонка невідомих,  $\boldsymbol{\lambda}$  – колонка сталих параметрів.

На випадок допоміжної колонки невідомих

$$\mathbf{y} = \mathbf{y}(\mathbf{x}), \quad (2)$$

рівняння стану матиме вигляд

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}_2(\mathbf{x}, \boldsymbol{\lambda}, t). \quad (3)$$

Початкові умови  $\mathbf{x}(0)$ , які дозволяють обійти перехідну реакцію, виступатимуть аргументом рівняння періодичності з періодом  $T$

$$\mathbf{f}(\mathbf{x}(0)) = \mathbf{x}(0) - \mathbf{x}(\mathbf{x}(0), T) = 0 \quad (4)$$

Розв'язання рівняння (4) проводять ітераційним методом Ньютона

$$\mathbf{x}(0)^{(s+1)} = \mathbf{x}(0)^{(s)} - \mathbf{f}'(\mathbf{x}(0)^{(s)})^{-1} \mathbf{f}(\mathbf{x}(0)^{(s)}). \quad (5)$$

Продиференціювавши цільову функцію (4) за  $\mathbf{x}(0)$  отримаємо матрицю Якобі

$$\mathbf{f}'(\mathbf{x}(0)) = \mathbf{I} - \Phi(T); \quad \Phi(T) = \left. \frac{\partial \mathbf{x}(\mathbf{x}(0), t)}{\partial \mathbf{x}(0)} \right|_{t=T}. \quad (6)$$

Матриця  $\Phi(T)$  – матриця монодромії у момент часу  $t = T$ . Її мультиплікатори дають повну відповідь про статичну стійкість знайденого періодичного стану [16]. Розв'язання рівнянь (4) та (5) становить двоточкову крайову задачу для диференціальних рівнянь стану.

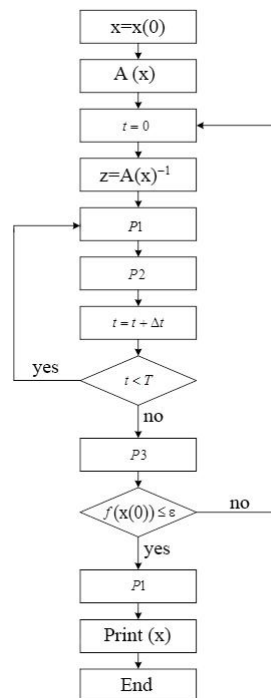


Рис. 1. Блок-схема входження в усталений стан виконавчого компонента комп'ютеризованої системи

Введемо позначення параметричних чутливостей відповідних невідомих:



$$\frac{\partial \mathbf{x}}{\partial \lambda} = \mathbf{C}_x, \quad \frac{\partial \mathbf{y}}{\partial \lambda} = \mathbf{C}_y. \quad (7)$$

Продиференціювавши за  $\lambda$  вираз (2) можна встановити зв'язок між  $\mathbf{C}_x$  і  $\mathbf{C}_y$

$$\mathbf{C}_x = \mathbf{A}\mathbf{C}_y, \quad (8)$$

де обернена матриця до  $\mathbf{A}$  матиме вигляд

$$\mathbf{A}^{-1} = \left( \frac{\partial \mathbf{y}(\mathbf{x})}{\partial \mathbf{x}} \right)^{-1}. \quad (9)$$

Диференціюючи за  $\lambda$  (1) і (3) отримаємо рівняння відповідних параметричних чутливостей

$$\frac{\partial \mathbf{C}_x}{\partial t} = \frac{\partial \mathbf{f}_1(\mathbf{x}, \lambda, t)}{\partial \mathbf{x}} \mathbf{C}_x + \frac{\partial \mathbf{f}_1(\mathbf{x}, \lambda, t)}{\partial \lambda}, \quad (10)$$

$$\frac{\partial \mathbf{C}_y}{\partial t} = \frac{\partial \mathbf{f}_2(\mathbf{x}, \lambda, t)}{\partial \mathbf{x}} \mathbf{A}\mathbf{C}_y + \frac{\partial \mathbf{f}_2(\mathbf{x}, \lambda, t)}{\partial \lambda}. \quad (11)$$

Для прикладу застосування рівнянь чутливостей (10) і (11) скористаємося диференціальними рівняннями стану насиченого дроселя ( $i = x$ )

$$\frac{di}{dt} = A(u - Ri). \quad (12)$$

За характеристикою намагнечування магнетопровода  $\psi = ai + bi^3$  із формул (2) та (9) встановлюємо значення коефіцієнта  $A$

$$A = \frac{I}{a + 3bi^2}. \quad (13)$$

Допоміжне рівняння дроселя матиме вигляд ( $\psi = y$ )

$$\frac{d\psi}{dt} = u - Ri. \quad (14)$$

В даній роботі розглядатимемо періодичну чутливість до опору резистора  $R$ . Підставляючи (13) у (12) і диференціюючи за  $\lambda$ , отримаємо (10)

$$\frac{\partial \mathbf{C}_x}{\partial t} = - \left[ \frac{R(a + 3bi^2) + (u - Ri)6bi}{(a + 3bi^2)^2} \mathbf{C}_x + \frac{i}{a + 3bi^2} \right]. \quad (12)$$

Диференціюючи за  $\lambda$  (14), одержимо (11)

$$\frac{\partial \mathbf{C}_y}{\partial t} = - \left[ \frac{R}{a + 3bi^2} \mathbf{C}_y + i \right]. \quad (13)$$

Зв'язок між чутливостями  $\mathbf{C}_x$  і  $\mathbf{C}_y$  можна знайти за формулами (8) та (13)

$$\mathbf{C}_x = \frac{\mathbf{C}_y}{a + 3bi^2}. \quad (17)$$

Задамо вихідну інформацію для розв'язання двоточкової крайової задачі для неоднорідних диференціальних рівнянь (15), (16):  $u = 100\sin(314t + \pi/6)$ ;  $R = 10$ ;  $a = 0.2$ ;  $b = 0.01$ . Відповідність результатів симуляції рівнянь (15) та (16) показана на рис. 3. Це підтверджує математичну обґрунтованість запропонованого методу. У найпростішому випадку бачимо, що

рівняння (15) є складнішим за рівняння (16). У випадку, коли коефіцієнт  $A$  набирає матричного змісту із складними взаємозв'язками між елементами матриці, то побудувати рівняння (15) неможливо.

Таблиця 1

$t, c$	$i_k, A$	$i_s, A$
0.0000E+00	-1.147138	-1.147138
5.5556E-04	-0.969788	-0.969787
1.6667E-03	-0.539052	-0.539051
2.2222E-03	-0.281953	-0.281952
3.3333E-03	0.262434	0.262434
3.8889E-03	0.522196	0.522196
5.0000E-03	0.962481	0.962481
5.5556E-03	1.13204	1.13204
6.6667E-03	1.36133	1.36133
7.2222E-03	1.42082	1.420819
8.3333E-03	1.43008	1.430076
8.8889E-03	1.37967	1.379670
9.9999E-03	1.16736	1.167359
1.0556E-02	1.00531	1.005312
1.1667E-02	0.576593	0.576593
1.2222E-02	0.319901	0.319902
1.3333E-02	-0.226146	-0.226145
1.3889E-02	-0.487892	-0.487891
1.4999E-02	-0.932913	-0.932912
1.5556E-02	-1.10467	-1.104669
1.6667E-02	-1.33745	-1.337454
1.7222E-02	-1.39822	-1.398216
1.8333E-02	-1.40915	-1.409148
1.8889E-02	-1.35919	-1.359193
1.9999E-02	-1.14714	-1.147138

Таблиця 2

$t, c$	$S$	$A\chi$
0.0000E+00	1.2754481E-02	1.2937506E-02
5.5556E-04	1.5861539E-02	1.5994631E-02
1.6667E-03	2.0506294E-02	2.0526893E-02
2.2222E-03	2.1750918E-02	2.1714665E-02
3.3333E-03	2.0768825E-02	2.0632748E-02
3.8889E-03	1.8624833E-02	1.8444171E-02
5.0000E-03	1.2623498E-02	1.2366969E-02
5.5556E-03	9.3618501E-03	9.0784589E-03
6.6667E-03	2.9923802E-03	2.6814227E-03
7.2222E-03	-7.0767004E-05	-3.8321465E-04
8.3333E-03	-6.0580457E-03	-6.3507301E-03
8.8889E-03	-9.0412721E-03	-9.3124555E-03
9.9999E-03	-1.5041226E-02	-1.5243614E-02
1.0556E-02	-1.7975548E-02	-1.8130289E-02
1.1667E-02	-2.2935783E-02	-2.2977958E-02
1.2222E-02	-2.4352683E-02	-2.4339039E-02
1.3333E-02	-2.3581881E-02	-2.3471912E-02
1.3889E-02	-2.1425027E-02	-2.1271167E-02
1.4999E-02	-1.5225075E-02	-1.4991916E-02
1.5556E-02	-1.1838963E-02	-1.1575701E-02
1.6667E-02	-5.2664322E-03	-4.9687168E-03
1.7222E-02	-2.1383308E-03	-1.8359367E-03
1.8333E-02	3.8982376E-03	4.1865609E-03
1.8889E-02	6.8657817E-03	7.1351584E-03
1.9999E-02	1.2754471E-02	1.2959572E-02

Таблиці 1 та 2 демонструють результати симуляції усталеного стану технічного пристрою за класичним ( $i_k$ ) і запропонованим ( $i_s$ ) методами та періодичну чутливість струму реального технічного пристрою до опору, за класичним ( $S$ ) та запропонованим ( $A\chi$ ) методами відповідно. Збіг результатів характеризується точністю числових обрахунків.

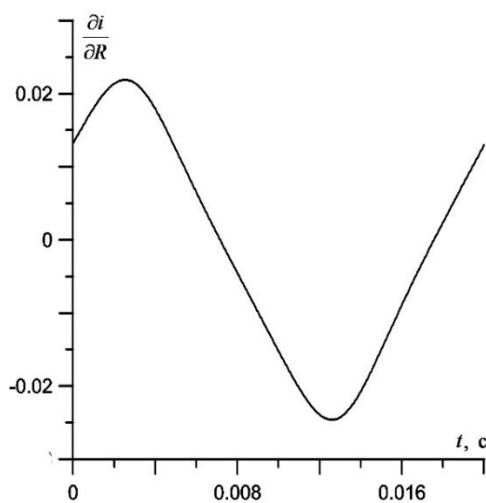


Рис 2. Усталена параметрична чутливість

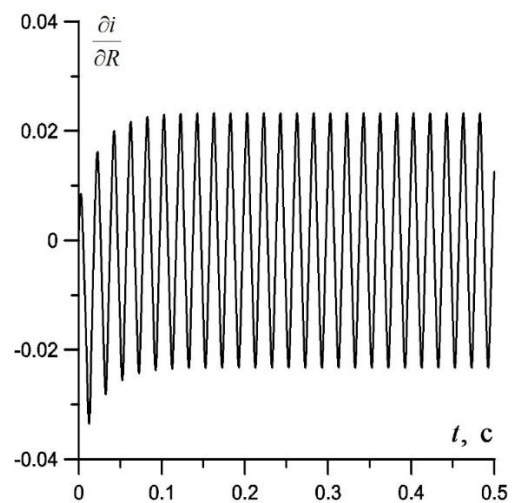


Рис 3. Перехідна параметрична чутливість

Результати комп'ютерних обрахунків перехідної параметричної чутливості класичного та запропонованого методів збігаються.

**Висновки та перспективи подальших досліджень.** В даній статті не йшла мова про переваги одного методу над іншим з погляду точності отриманих результатів. Робота показує складність їх застосування до нелінійних систем. Запропонований метод допоміжної чутливості дає змогу виключити операцію диференціювання матриці коефіцієнтів  $\mathbf{A}$ . Ця задача досягнута з усією математичною строгістю, що підтверджують результати комп'ютерної симуляції.

1. Aprile T.I. A computer algorithm to determine the steady-state response of non-linear oscillators / Aprile T.I., Trick T.N. // IEEE, Trans. Circuit Theory. – 1972. – vol. 19. – P. 354-360.
2. Donald G. Fink, H. Wayne Beaty Standard handbook for electrical engineers / McGRAW-HILL BOOK, twelfth edition. – NY. – 1987. – 1600 p.
3. Khusainov D. Introduction in modeling of dynamical systems (in Ukrainian) – Kyiv, 2010. – 130 p.
4. Korn G.A., Korn T.M. Mathematical handbook for scientists and engineers / McGRAW-HILL BOOK. – N.Y., Toronto, London. – 1961. – 720 pp.
5. Kozoriz V. Transport and the future (in Ukrainian) // Technical news, 2005 / 1(20), 2(21). – PP. 29-34.
6. Lyashko S.I. Generalized Optimal Control of Linear Systems with Distributed Parameters / Dordrecht, Boston, London. – Kluwer Academic Publishers. – 2002. – 455 p.
7. Melnik A. Architecture of computer (in Ukrainian) – Lutsk, 2008. – 470 p.
8. Tavares M., Portela C. Transmission system parameters optimization–sensitivity analysis of secondary arc current and recovery voltage // Power Delivery, 2004. – V. 19, № 3. – PP. 1464-1471.
9. Tchaban V. Mathematical modeling in electrical engineering (in Ukrainian). – Lviv: Taras Soroka's publisher home, 2010, 508 p.
10. Tchaban V. Mathematical modeling of electromechanical processes (in Russian). – Lviv, 1997, 344 p.
11. Tchaban V., Bily L. Mathematic modelling of periodic regimes of electric machines / Modelling, simulation and control. – 1984. – Vol. 1, № 1. – P. 35-39.
12. Tchaban V., Kostyuchko S. Parametric sensitivity of three-phase induction motor // Technical news, 2014 / 1(39), 2(40). – PP. 26-30.
13. Tchaban V., Kostyuchko S., Tchaban Z. Auxiliary model of parametric sensitivity // Computational problems of electrical engineering. – Lviv, 2012. – V.2, № 2. – PP. 105-111.
14. Tchaban V., Kozoriz V. Mathematical modelling of interaction of superconductors magnetic systems / Proceedings Intern. AMSE Conference "Systems". – London. – 1993. – Vol. 3. – P. 149-153.
15. Tchaban V., Tchaban O. Derivative of matrices of parameters over independent variables (in Ukrainian) // Technical news, 2014 / 1(39), 2(40). – P. 51
16. Tchaban V., Tchaban O., Kostyuchko S., Tchaban Z. About one two-point boundary value problem // Technical news, 2011 / 1(33), 2(34). – PP. 10-12.
17. Tchaban V., Tchaban O., Kostyuchko S., Tchaban Z. The theory of electromagnetic circuits. – In monograph: Computing in Science and Technology, Rzeszow: Wyd-wo Uniwersytetu Rzeszowskiego, ISBN 978-83-7338-895-6, 2012/13, 172 p. (pp. 34-55).
18. Tchaban V., Tchaban Z. Simulation of static stability of three phase induction motor. – Przegląd elektrotechniczny, R. 87, NR 9a/2011, pp. 295–297.

УДК 004.73:37.013.77

Чеб С.С., Мельник В.М., Мельник К.В., Самарчук В.Ф.  
Луцький національний технічний університет

## ХАРАКТЕРИСТИКА ДЕЯКИХ ЗАКОНОМІРНОСТЕЙ ВПЛИВУ СОЦІАЛЬНИХ МЕРЕЖ ІНТЕРНЕТ НА ОСОБИСТІСНИЙ РОЗВИТОК СТУДЕНТІВ

**Чеб С.С., Мельник В.М., Мельник К.В., Самарчук В.Ф.** Характеристика деяких закономірностей впливу соціальних мереж інтернет на особистісний розвиток студентів. В роботі розглянуто окремі закономірності впливу соціальних мереж на особистісний розвиток студентів. Здійснено аналіз результатів отриманих за допомогою проведеного анкетування, бесіди та інтерв'ю. Проаналізовано формування інтегративної думки студентів про соціальні мережі.

**Ключові слова:** соціальна мережа, залежність, соціальне опитування, Інтернет.

**Чеб С.С., Мельник В.М., Мельник К.В., Самарчук В.Ф.** Характеристика некоторых закономерностей влияния социальных сетей интернет на личностное развитие студентов. В работе рассмотрены некоторые закономерности влияния социальных сетей на личностное развитие студентов. Осуществлен анализ результатов полученных с помощью проведенного анкетирования, беседы и интервью. Проанализировано формирование интегративного мнения студентов о социальных сетях.

**Ключевые слова:** социальная сеть, зависимость, социальный опрос, Интернет.

**Cheb S.S., Melnyk V.M., Melnyk K.V., Samarchuk V.F.** Some conformity characterisation with social networks influence to the personality growing of students. The conformities with the social networks influence to the personality growing of students are considered in the article. The analysis of the results, which are accepted after made questionnaires, conversations and interviews, is considered here. There also the integrated mean of the students about social networks is discussed.

**Keywords:** Social network, dependence, social poll, Internet.

**Постановка проблеми.** Що являє собою соціальна мережа сьогодні – відомо практично всім, хто хоч якимось чином вважає себе мініатюрною частинкою сучасного суспільства із доступом (постійним чи частковим) до мережі Інтернет. По суті, соціальна мережа – це віртуальне об'єднання людей, де можна обмінюватися певною інформацією, що в широкому розумінні є характеристикою самого поняття „Інтернет”. Причини її існування очевидні: сьогодні люди проводять величезну кількість часу за комп'ютером і звикли обмінюватися інформацією одне з одним, насамперед у електронному вигляді, адже це, як мінімум, економить час [1]. Першим веб-сайтом, який пропонував можливості роботи із соціальними мережами, був classmates.com, який з'явився в 1995 році. Вслід за ним, в 1997 році з'явився SixDegrees.com. Починаючи з 2001 року почали з'являтися сайти, в яких використовувалась технологія під назвою «Коло друзів». Ця форма соціальних мереж, яка широко використовується у віртуальних спільнотах, набула широкої популярності та розквітнула з появою сайту Friendster. Насьогодні існує велика кількість сайтів з можливостями організації соціальних мереж. 4 лютого 2004 року почала працювати соціальна мережа FaceBook. Популярність цих сайтів постійно зростала, і в 2005 році було більше переглядів сторінок сайту MySpace, ніж сайту Google. Google також пропонує веб-сайт з можливостями роботи із соціальними мережами. Соціальні мережі почали розглядатись як складова Інтернет стратегії приблизно в той самий час: в березні 2005 року Yahoo запустила Yahoo! 360°, а в липні 2005 News Corporation запустила MySpace. 1 жовтня 2006 року було запущено відому російську соціальну мережу «В Контакте», що набула навіть великої популярності на теренах Росії, України, Білорусії, Казахстану. Що стосується українських соціальних мереж, то одна з найвідоміших вітчизняних мереж Connect.ua була створена у грудні 2007 року [2].

На даний час більшість соціальних мереж розвиваються дуже високими темпами, що доводить постійно зростаючу кількість зареєстрованих користувачів. Цифри говорять самі за себе. Навесні 2010 року кількість користувачів Facebook досягла 470 млн. чоловік. Усього лише за рік об'єм аудиторії FaceBook подвоївся. Батьківщина цієї соціальної мережі – США. Twitter – третя за величиною соціальна мережа у світі, створена в 2006 році в США. На сьогодні користувачами Twitter є 74 млн. чоловік по всьому світу. «Vkontakte» – це найбільша соціальна мережа в Рунеті, яка займає 9-е місце в списку найбільших соціальних мереж світу. Вона активізована в 2006 році. Сервіс отримав широку популярність безпосередньо в Росії, на Україні, у Білорусі і Казахстані. Станом на 2010 рік більшість користувачів сягає понад 95 млн. чоловік [3].

За даними дослідників соціальні мережі мають свої переваги і недоліки. Користувачі різного віку знаходять всляку інформацію, яка може в чомусь допомогти і, навпаки, знищити певні набуті морально-духовні цінності. Однак, попри певні переваги, хоча з цим трудно погодитися, соціальні

мережі породжують і шкідливі наслідки. Особливим є те, що там криються численні небезпеки, про котрі ми не раз навіть і не здогадуємось до кінця, чи не хочемо це визнати. Проблеми можуть стрінути кожного: Інтернет може немовби прив'язати до себе та відібрати, певною мірою, потрібну свободу або спокій [4]. За даними досліджень від пагубного впливу соціальних мереж найбільше страждає молодь і студенти [5,6], а тому в даній роботі буде досліджено вплив соціальних мереж саме на цю групу населення, як найбільш впадаючу в залежність від них.

**Аналіз останніх досліджень.** Аналіз соціальних мереж (має відношення і теорія мереж) перетворився на основний метод досліджень в сучасній соціології, антропології, географії, соціальній психології, інформатиці та дослідженні організацій, а також поширену тему для досліджень та дискусій. Дослідження в декількох академічних галузях показали, що соціальні мережі діють на багатьох рівнях, починаючи від родин, і закінчуючи «цілими націями». Вони відіграють важливу роль в тому, як розв'язуються проблеми, працюють організації, та досягають успіху на шляху до власних цілей індивіди. Учені Британського інституту біології Biologist встановили, що фізіологічні процеси людини, яка спілкується з друзями у віртуальній реальності, не ідентичні тим, що виникають при реальному контакті. «Вам здається, наче ви спілкуєтесь, але ваш організм поводить ся так, наче ви цілком самотні», – говорить біолог Арік Сігман [5].

Залежність від соціальних мереж схожа до наркотичної. До такого висновку прийшли американські дослідники з університету штату Меріленд (США) [7]. Вчені поставили «жорстокий» досвід: на 24 години позбавили 200 американських студентів засобів комунікації: телефонів, Інтернету і соціальних мереж. Продемонстровані молодими людьми симптоми, були, як дві краплі води, схожі на ознаки наркотичних «ломок» – нервозність, підвищене занепокоєння і зниження рівня активності. Багато хто з студентів заявив, що втрата засобів зв'язку рівносильна для них втраті друзів та сім'ї. Тобто віртуальні мережі є неодмінною умовою соціального функціонування американської молоді. Просте спілкування з періодами очікування зустрічі та відповіді вже нікого не задовольняє.

Перейдено до аналізу статистики соціальних мереж. Українські дослідники Форкун та Білик в своїй роботі [8] говорять, що більшість користувачів соціальних ресурсів відвідує їх з метою спілкування з друзями або пошуку людей. Нові знайомства готовий завести на Інтернет-просторах кожен. Ставлять за мету розповідь про себе іншим людям 4% респондентів. Переважна більшість користувачів відвідує соціальні ресурси заради спілкування з друзями та знайомими – 86%, готові до нових зустрічей – 5%, в рівній мірі спілкуються і зі старими друзями, і шукають нових – біля 9% відвідувачів. Більше двох третин опитаних підтримують в соціальній мережі тільки особисті контакти, які направлені тільки на підтримку ділових контактів. Це 1,5% респондентів, кожен третій з яких поєднує особисті і ділові контакти (30%). Відвідувачі соціальних мереж довіряють інформації, яку публікують про себе люди. Беззастережно такій інформації довіряють 5% респондентів, швидше довіряють, ніж не довіряють – 68%, вважають, що дана інформація швидше невірна – 14% опитаних, і 2% користувачів соціальних ресурсів не довіряють інформації, розміщеній на цих сайтах.

Кожен другий користувач соціальної мережі (53%) не помічає реклами, розміщеної на сайті, а іноді звертає на неї увагу 42% користувачів, із яких ~3% – помічають рекламу завжди. При цьому жінки і чоловіки відповідали на це питання однаково. Не залежав розподіл відповідей і від віку. Викликає інтерес зміст рекламних повідомлень, що лише цікавий для ~10% відвідувачів і не цікавий зміст реклами – для 64% респондентів. При цьому, її наявність на сайті їх не дратує. Досить гостро реагує на присутність реклами кожен п'ятий відвідувач соціальних мереж (22%), а їхня реклама в соціальних мережах за словами респондентів дратує і відволікає їх.

Творець концепції WWW, Тім Бернерс-Лі, впевнений [10], що соціальні мережі становлять небезпеку для Інтернету і його користувачів, так як останні залишають там дуже багато персональних даних та інших відомостей особистого характеру, які потім залишаються в мережах. Більше того, мережі розпоряджаються цими даними і не вважають за потрібне з ким-небудь ділитися своїм контентом. «Великі соціальні мережі постійно завантажують в себе масу даних, які набираються користувачами в іншому інтернеті», – говорить він.

Бум соціальних мереж проник у всі куточки планети. Однак не всі країни допустили до себе це «ноу-хау». Спираючись на достовірні джерела [11], відомо, що влада Китаю в попередні роки, наприклад, заборонила такі світові соціальні мережі, як Twitter, Flickr, Bing, Live.com, Hotmail.com, Wordpress, YouTube, Blogger. Окрім позитивного впливу, всі сайти неформального спілкування несуть і негативні впливи. Розповсюдження неправдивої інформації, провокації, шантаж, – все це також здійснюється через соціальні мережі. Поширення недостовірної інформації відбувається також

через розміщення її на сайтах, блогах, незареєстрованих засобах масової інформації в мережі Інтернет (Інтернет-порталах) тощо. [15]. Якщо глобальні соціальні мережі не закриють, то вони матимуть великий шанс втратити актуальність, або дати поштовх до створення нових технологій спілкування на більш вищому рівні.

**Постановка задачі.** Психологи твердять: молодь, живучи у віртуальному світі соціальних мереж, втрачає поняття реальних життєвих цінностей, а то і інтелектуально деградує. Замість виконання домашнього завдання, допомоги по дому, школярі годинами передивляються сторінки "друзів" (так називаються контакти користувача), у тому числі часто нецензурні фото та відео. Найважливіші події свого життя тепер публікують спочатку на веб-сторінці. Завдяки цьому психіка молоді може порушуватися, що призводить до депресій та суїцидів [12]. Тому основною задачею даної роботи є дослідження проблеми залежності особистості нинішнього студента від соціальних мереж. Виникає любопитне питання – це дізнатися, чи користуються нинішні студенти соціальними мережами, яким соціальним мережам надають перевагу, як часто знаходяться вони на даних ресурсах Інтернет і яка основна мета постає для них при відвідуванні даних сайтів.

**Основна частина.** Для вирішення поставлених задач було проведено анкетування, під час якого опитуваним (в даному опитуванні прийняло участь 105 респондентів) потрібно було дати відповіді на питання, представлені нижче і подібні ним:

1. Чи зареєстровані Ви в соціальних мережах?
2. Якій соціальній мережі Ви надасте перевагу?
3. Як часто Ви заходите на дані Інтернет сайти?
4. Яка Ваша основна мета при відвідуванні соціальних мереж

Одержані результати дослідження представлені на рис.1.

На питання: "*Чи зареєстровані Ви в соціальних мережах?*" відповіли:

- «Так» – 85,71 % опитуваних,
- «Ні» – 14,29 % опитуваних (рис. 1).

Отже, за результатами першого питання бачимо, що дуже велика частка опитаних респондентів є користувачами соціальних мереж (більше 86 %). Однак це не має викликати великого занепокоєння, якщо спостерігати за цим зі сторони. Але така велика кількість зареєстрованих користувачів говорить про надзвичайний вплив соціальних мереж на студентів, їх «віртуальне» віддалення від реального світу.



Рис. 1 – Результати опитування по першому питанню

Друге питання звучало: "*Якій соціальній мережі Ви надасте перевагу?*" На це питання отримано наступні результати:

- «Вконтакте» – 62,86 % опитуваних;
- «Однокласники» – 3,8 % опитуваних;
- «Facebook» – 14,29 % опитуваних;
- «My Space» – 0,95 % опитуваних;
- «Twitter» – 0,95 % опитуваних;

- «Мой Мир» – 2, 86 % опитуваних;
  - «Не зареєстрований ніде» – 14, 29 % опитуваних.
- Аналіз результатів приведений в графічному варіанті на рис. 2.



Рис. 2 – Результати опитування по другому питанню

Як бачимо, великий відрив тут здобула соціальна мережа Павла Дурова «Vkontakte», яка містить 62,86 % опитуваних респондентів, задіяних в нашому дослідженні.

Третім запитанням було наступне: "Як часто Ви заходите на дані Інтернет сайти?" Наведемо результати відповідей респондентів:

- «Один раз на день» – 28,57 % опитуваних;
  - «Два-три рази на день» – 27,6 % опитуваних;
  - «Три-п'ять разів на день» – 12,4 % опитуваних.
  - «Частіше ніж п'ять разів на день» – 17,14 % опитуваних.
- Результати дослідження за даним питанням подані на рис. 3.



Рис. 3 – Результати опитування по третьому питанню

Приведений аналіз дослідження по третьому питанню показує, що на відміну від двох попередніх питань, в цьому випадку результати розділились більш-менш рівномірно. Тут не проявляється яскраво вираженого домінування. Відштовхуючись від даного результату можна сказати, що опитані респонденти не так вже й часто заходять в соціальній мережі на протязі дня, що, зважаючи на результати інших опитувань [9], являється значно нижчим і дозволяє формувати думку про те, що нинішня студентська молодь не живе тільки віртуальним, а децю і реальним життям.



Наступне запитання: "Яка Ваша основна мета при відвідуванні соціальних мереж?" привело до одержання наступних результатів:

- «Спілкування» – 57,14 % опитуваних;
  - «Перегляд відео» – 8,57 % опитуваних;
  - «Прослуховування музики» -15,24 % опитуваних;
  - «Перегляд соціальної та іншої реклами» – 4,8 % опитуваних.
- Графічно дані результати приведені на рис.4, нижче.



Рис. 4 – Результати опитування по четвертому питанню

Згідно наведених результатів саме спілкування є основною метою відвідування даних і подібних сайтів студентською молоддю в нашому територіальному регіоні. Спілкування є дуже важливим елементом в житті кожної молодої особистості. Оскільки в спілкуванні завжди є момент пізнання людини людиною, а також пізнання кожним самого себе, то спілкування завжди є сильним засобом формування в людини ставлення до інших людей, а також до самої себе. На сьогодні важливо розвивати в особистості такі якості, як доброзичливість, толерантність, симпатія, повага до інших [13]. Тільки ці риси не завжди доводиться проявляти під час віртуального спілкування. Віртуальному спілкуванню за результатами наших досліджень віддають перевагу невпевнені в собі, сором'язливі люди, яким важко спілкуватися контактувати з іншими віч-на-віч. Нерідко вони відчують самотність, відчуженість. З частіша їм здається, що оточуючі їх не розуміють. Все це – серйозні психологічні проблеми. Та замість того, щоб шукати шляхи їх розв'язання, вони втікають у світ ілюзій, де почувають себе набагато впевненіше [14].

З метою бажання з'ясувати, що ж так приваблює молодь в соціальних мережах та виявити думку студентів про це, декільком студентам було задано запитання: "Що для Вас являють собою соціальні мережі?" Ось деякі коментарі з цього приводу.

Студент віком 21 рік відповів: «Негативно відношусь до соціальних мереж, але вони добре підходять для спілкування з людьми, яких давно не бачив...»

Студентка віком 19 років прокоментувала: «Я досить негативно відношусь до соціальних мереж, оскільки спілкування в них "затягує" і не дає спокійно жити...»

Студентка віком 22 роки сказала такі слова: «Соціальні мережі досить вагома і потрібна річ в наш час, без якої обійтись неможливо...»

Студент, віком 20 років відповів: «Соціальні мережі – це найзручніший засіб спілкування, перегляду відео, прослуховування музики та організації масових зустрічей.»

Студентка віком 19 років заявила: «Перш за все – це засіб пошуку інформації, яку неможливо знайти в бібліотеці, ... можливість познайомитись з цікавими людьми, ну і, звичайно, спілкування з друзями та прослуховування музики...»

Інші відповіді приблизно в однаковій кількості від числа опитуваних з розрахунком на представлені відповіді домінували одну із них. Спираючись на отримані коментарі, можна зробити висновок, що ставлення студентів до соціальних мереж є неоднозначним. Більшість вважає, що вони є досить потрібна річ, необхідна для спілкування, обміну та пошуку інформації, можливістю для придбання нових знайомств та організації масових заходів, але є і негативні відгуки, які базуються на



тому, що спілкування в соціальних мережах «затягує» і відсторонюють людини від реального світу. Тому їх вплив на особистісний розвиток є неоднозначним та двояким.

**Висновки і перспективи.** Провівши дане дослідження було отримано наступні цікаві соціологічні результати. Велика кількість респондентів є користувачами соціальних мереж (більше 86 % опитаних), а найпопулярнішою соціальною мережею за наведеними результатами опитування стала мережа «Facebook» (62,86 % опитуваних). Більшість опитуваних респондентів заходять в соціальні мережі один раз на день (28,57 % опитуваних) і, згідно результатів дослідження, головною метою більшості користувачів соціальних мереж є спілкування (57,14 % опитуваних). Згідно результатів проведеної бесіди та інтерв'ю зі студентською молоддю можна зробити висновок, що не всі студенти позитивно відгукуються про соціальні мережі, що дає змогу говорити про їх сподівання негативного впливу мереж в першу чергу на них і на особистість в цілому.

Соціальна мережа на думку іншої позиційної сторони студентів – це допомога, яку можна отримати сьогодні й негайно. Як було акцентовано в процесі дослідження: "Експерти, співробітники, партнери – усіх цих людей можна знайти через Інтернет". Отже, залишається лише повторити улюблену фразу батьків американських дітей: «Дозволяючи нашим дітям бути присутніми у соціальних мережах, ми сподіваємося на те, що вони знайдуть там не тільки те, що потрібно їм» [14].

1. Дубенко Л. Соціальні мережі: реальні загрози віртуального світу [Електронний ресурс]. - 2011. - Режим доступу до статті: [http://www.golosua.com/main/article/politika/20110221\\_sotsialni-mereji-realni-zagrozi-virtualnogo-svitu](http://www.golosua.com/main/article/politika/20110221_sotsialni-mereji-realni-zagrozi-virtualnogo-svitu).
2. Історія виникнення глобальних соціальних мереж [Електронний ресурс]. – Режим доступу до статті: [http://wiki.fizmat.tnpu.edu.ua/index.php/Лекція:\\_Основні\\_поняття.\\_Історія\\_виникнення\\_глобальних\\_соціальних\\_мереж](http://wiki.fizmat.tnpu.edu.ua/index.php/Лекція:_Основні_поняття._Історія_виникнення_глобальних_соціальних_мереж).
3. Статистика соціальних мереж [Електронний ресурс]. – Режим доступу до статті: <http://money-ukraine.at.ua/forum/2-13-1>.
4. Позитивне і негативне соціальних мереж [Електронний ресурс]. – Режим доступу до статті: <http://dyvensvit.org/blogy/221-blog-omykoly-mykosovskogo/2695-pozytyvne-j-negatyvne-socialnyh-merezh>.
5. Ковальова Л. Соціальні мережі б'ють по реальному життю [Електронний ресурс]. – 2010. - Режим доступу до статті: <http://lady.tochka.net/ua/3836-sotsialnye-seti-byut-po-realnoy-zhizni>.
6. Соціальні мережі небезпечні для здоров'я [Електронний ресурс]. – 2010. - Режим доступу до статті: <http://www.credo-ua.org/2010/01/13088>.
7. Залежність від СМС та соціальних мереж побідна до наркотичної [Електронний ресурс]. – 2010. - Режим доступу до статті: <http://www.credo-ua.org/2010/04/22346>.
8. Форкун Ю.В. Дослідження факторів впливу на рейтинг соціальних мереж та соціальних сайтів в Інтернеті. / Ю. В. Форкун В. В. Білик. - Восточно-Европейский журнал передовых технологий. - 2010. - с. 25-28.
9. Бернерс-Лі Т. Соціальні мережі фрагментують Інтернет [Електронний ресурс]. – Режим доступу до статті: <http://starodub.org.ua/2010/11/23/timothy-john-berners-lee>.
10. Митрошина Н. Соціальні мережі : за чи проти? [Електронний ресурс]. – Режим доступу до статті: <http://www.rtkor.com/news/2010/02/09/98777.new>.
11. Павлице Н. Соціальна мережа залежності [Електронний ресурс]. – 2010. - Режим доступу до статті: [http://www.beregovo.uz.ua/ua/index.php?option=com\\_content&view=article&id=509:2010-08-03-12-49-51&catid=11:2009-05-03-21-00-45](http://www.beregovo.uz.ua/ua/index.php?option=com_content&view=article&id=509:2010-08-03-12-49-51&catid=11:2009-05-03-21-00-45).
12. Нор К.Ф. Спілкування студентів у груповій навчальній діяльності. – 2009. – с. 149-154.
13. Набока І. Віртуальне спілкування: гра чи втеча від реального світу? [Електронний ресурс]. – 2005. - Режим доступу до статті: <http://www.vox.com.ua/data/publ/2005/09/28/virtualne-spilkuvannya-gra-chy-vtecha-vid-realnogo-svitu.html>.
14. Протасова Г. У мережі як вдома [Електронний ресурс]. – 2011. - Режим доступу до статті: <http://osvita.com/publications/23-02-2011/1298454690>.
15. Соболевський А. Спростування недостовірної інформації, яка поширена через Інтернет [Електронний ресурс]. – 2010. – Режим доступу до статті: <http://crp.in.ua/ua/articles/id-29>.