

УДК 004.415

А.М. Ковальчук, асист.
В.Г. Левицький, к.т.н., доц.
І.І. Самолюк, студ.

Житомирський інженерно-технологічний інститут

КОМП'ЮТЕРНИЙ ЛАБОРАТОРНИЙ ПРАКТИКУМ З ТЕОРІЇ АВТОМАТИЧНОГО КЕРУВАННЯ

В статті наведено опис деяких особливостей проектування, реалізації та використання програмного засобу для розробки комп'ютерних лабораторних практикумів з теорії автоматичного керування. Проведено порівняння характеристик створеного програмного забезпечення для лабораторного практикуму з відомими аналогами. Розглянуто ефективність використання програмного забезпечення в процесі підготовки студентів вищих навчальних закладів.

Швидкі темпи розвитку науки і техніки вимагають розробки великої кількості складних технічних систем, які виконують ті чи інші функції. В наш час неможливо уявити роботу цих систем без пристроїв автоматичного керування (САК). Теорія автоматичного керування (ТАК) є однією з фундаментальних дисциплін майже у всіх технічних вищих навчальних закладах. Викладання такої складної дисципліни практично неможливе без проведення лабораторних практикумів, які б добре ілюстрували теоретичний матеріал та надали можливість студентам на практиці ознайомитись з методами аналізу й дослідження САК.

Для побудови подібного лабораторного практикуму можна використати один або декілька з відомих систем чисельного аналізу та моделювання [1], наприклад, різноманітні версії програм MathCAD [2], MATLAB [3–6], Matematica [7, 8], Maple V [9, 10] та інші. Виявляється, однак, що дані програми не завжди зручно використовувати в навчальному процесі, оскільки вони, головним чином, орієнтовані на проведення певних математичних розрахунків. Існують також й інші причини, які ускладнюють використання даних систем програм в навчальному процесі: в першу чергу – це причини економічного та навчально-методичного характеру [11]. Зважаючи на це, було вирішено спроектувати та розробити власний лабораторний практикум з ТАК на основі програмного комплексу чисельного аналізу та моделювання “DSR Open Lab 1.0” (Dynamical systems research open laboratory – відкрита лабораторія дослідження динамічних систем) [12–15]. Основними особливостями програмної системи (ПС) “DSR Open Lab 1.0” є: по-перше, функціональна відкритість, що означає можливість добавляти різноманітні функціональні модулі – нові задачі, нові види початкових даних, види досліджень, бібліотеки математичних алгоритмів, види вихідних результатів; по-друге, в програмному комплексі реалізовано візуальні та інтерактивні засоби управління процесом чисельного аналізу та моделювання.

Для розв'язання задач ТАК сучасні системи чисельного аналізу та моделювання (Mathematica, MATLAB тощо) пропонують потужні бібліотеки алгоритмів, які, як правило, використовують базові сервіси математичного ядра ПС. Будь-яка спроба конкурувати з великими комерційними проектами, розробляючи власний лабораторний практикум “з нуля”, зазвичай не дає позитивних результатів у порівнянні з грандами індустрії математичного програмного забезпечення. Для вирішення цієї проблеми представлений спосіб організації лабораторного практикуму з ТАК, що ґрунтується на використанні лінгвістичних, математичних та сервісів інтерфейсу користувача програмної системи чисельного аналізу “DSR Open Lab 1.0”. Таким чином, відбувається вирішення проблем створення:

- 1) алгоритмічної бази – починаючи від визначення необхідних програмних структур даних (комплексні числа, дроби, дробово-раціональні функції тощо) і закінчуючи відомими з літератури алгоритмами чисельного диференціювання та інтегрування, розв'язання алгебраїчних і диференціальних рівнянь та інших необхідних розділів математики;
- 2) лінгвістичного забезпечення для опису характеристик систем, що досліджуються;
- 3) графічного інтерфейсу користувача з елементами адаптації до вигляду виду та методу чисельного аналізу чи моделювання в рамках задачі, що вирішується користувачем.

Подібний, загальний в деякому розумінні, підхід до створення комп'ютерних лабораторних практикумів, з одного боку, дозволяє сконцентруватися на алгоритмах вищого рівня ієрархії методів, а з іншого – забезпечує функціональну відкритість практикумів для подальшого їх нарощування новими методами та засобами. Проведемо короткий огляд функціональних можливостей для лабораторних практикумів з ТАК, що можуть бути створені на основі алгоритмів визначених вище систем чисельного моделювання та ПС “DSR Open Lab 1.0”. Також розглянемо деякі особливості організації процесу спілкування користувача та програмного засобу, а саме: лінгвістичного та графічного інтерфейсів користувача.

Для порівняння основні види чисельного аналізу класичної ТАК, що реалізовані в програмних системах, розглянутих в даній роботі, представлено в табл. 1.

Таблиця 1

Функціональні можливості систем чисельного аналізу та моделювання в предметній області ТАК

Вид аналізу	1*	2	3
Дослідження у часовій області			
Побудова перехідної характеристики	√	√	√
Побудова імпульсної характеристики	√	√	√
Побудова реакції системи на довільну вхідну дію	√	√	–
Дослідження у частотній області			
Побудова амплітудно-фазової характеристики	√	√	√
Побудова кривої Михайлова	–	–	√
Побудова логарифмічних частотних характеристик	√	√	√
Побудова дійсної та уявної частотних характеристик	–	–	√
Дослідження стійкості за алгебраїчними критеріями			
Метод Рауса	–	–	√
Метод Гурвіца	–	–	√
Безпосереднє дослідження коренів характеристичного рівняння	√	√	√

* 1 – Mathematica 4.0; 2 – MATLAB 6.1; 3 – DSR Open Lab 1.0.

Як видно з табл. 1, представлений лабораторний практикум з ТАК (3-й стовпчик таблиці) загалом не поступається, а іноді переважає в функціональності існуючі аналоги**, що дозволяє ефективно використовувати його в навчальному процесі та доводить вірність ідеї ієрархічної розробки алгоритмічної бази практикуму як надбудови над програмною системою “DSR Open Lab 1.0”.

Спосіб опису задач ТАК, застосований в представленому лабораторному практикумі, цілком відповідає стилю, прийнятому в програмній системі “DSR Open Lab 1.0”, принципово відрізняючись при цьому від описів з допомогою математичного програмного забезпечення типу MATLAB, Mathematica, Maple V тощо.

Всі згадані програмні інструменти для рішення математичних задач реалізують принцип використання предметно-орієнтованих програмних середовищ і мов. Однак MATLAB, Mathematica та Maple V схилиються при цьому до надмірного ускладнення опису математичної задачі практично на всіх етапах: в частині лексики (велика кількість зарезервованих слів, зокрема, імен функцій), синтаксису (громіздкі конструкції, запозичені з універсальних мов програмування) та семантики (складні принципи типізації і т.д.). І хоча бажання мати універсальний засіб визначення проблеми, незмінний для всього циклу вирішення математичної задачі, гнучкий та потужний, є цілком виправданим, якщо згадати широту завдань та функціональність програмних засобів для рішення математичних задач, та все ж застосування предметно-орієнтованих мов, ускладнених до рівня мови програмування загального призначення (MATLAB, Mathematica, Maple V тощо), є прийомом небажаним, а в певних випадках (наприклад, для використання в навчальному процесі) й абсолютно неприпустимим з огляду на специфіку кінцевого користувача, який може, взагалі кажучи, і не мати необхідних навичок програмування.

На відміну до такого сумнівного застосування предметно-орієнтованих мов, опис задачі програмної системи “DSR Open Lab 1.0” модифікує існуючі підходи до реалізації інтерфейсу людини та комп’ютера і забезпечує більш простий та прийнятний для непідготовленого користувача спосіб комунікації з програмним засобом. Це забезпечується реалізацією нової методики організації лінгвістичного забезпечення чисельного аналізу математичних задач – методики, що базується на двох принципах: 1) нетрадиційному для математичного програмного забезпечення розмежуванні функцій опису задачі між програмним графічним інтерфейсом і мовою та 2) використанні графічного введення математичних формул [16, 17]. Перший принцип звужує сферу застосування лінгвістичного забезпечення при описі математичної задачі та відповідно зменшує об’єм дій користувача, які не можуть бути інтерактивно контрольованими програмним комплексом (програмування з допомогою предметно-орієнтованої мови відноситься саме до таких неконтрольованих дій – збільшуючи кількість помилок та зменшуючи надійність роботи програми.) Завдяки другому принципу опис задачі чисельного аналізу максимально наближується до природної мови даної галузі – математичної нотації.

Новою також є схема організації діалогу користувача та ПС, яка була використана для створення інтерфейсу користувача ПС. За даною схемою процес аналізу задачі представляється за допомогою навігатора чисельного аналізу задачі та майстра чисельного аналізу і моделювання.

** Автори вважають за необхідне окремо відмітити відсутні в табл. 1 можливості програмного інструменту Simulink, який входить до складу програмної системи MATLAB. Згаданий інструмент суттєво покращує характеристики MATLAB як бази для лабораторного практикуму з ТАК, окреслюючи таким чином коло майбутньої роботи авторів у галузі програмного забезпечення курсу ТАК.

Навігатор представляє користувачеві множину видів аналізів, які можна проводити над вибраною математичною задачею. З цієї множини користувач системи може вибрати один або групу видів аналізів, які він хоче здійснити. Після цього за його вибором буде створена або активізована компонента майстра опису математичної моделі.

Для чисельного аналізу та моделювання в інтерфейсі користувача реалізовано режим майстра моделі. Кожний крок майстра чисельного аналізу та моделювання реалізовано у вигляді окремої програмної компоненти, що створена як сервер активного документу. Відображення даних з серверів (компонент) відбувається шляхом вміщення до інтерфейсу компоненти до клієнтської частини програми контейнера (та ж операція проводиться над панелями інструментів). Майстер опису математичної моделі допомагає користувачу у максимально наближеному до предметної області вигляді описати характеристики задачі, яку потрібно розв'язати або дослідити. Користувач може задати початкові умови та інші параметри, вибрати декілька додаткових видів аналізів даної задачі, переглянути результати своєї роботи та записати свої коментарі до проведеного розв'язку задачі. За елементи майстра моделі використовують компоненти: Equation Builder – для введення опису математичної моделі задачі; сервер додаткових вхідних даних; сервер вибору завдань на аналіз, він також управляє вибором видів вихідних результатів; компоненти для перегляду вихідних результатів (Equation Builder, Graph Server), сервер коментарів. Всі перераховані компоненти мають універсальний характер в тому розумінні, що вони можуть бути використані для різних типів математичних моделей та видів аналізу. В той же час, при виборі конкретної математичної моделі та виду аналізу кожна з компонент може бути спеціально настроєна для використання саме у даній задачі. Якщо, з певних міркувань, яка-небудь з компонент не може бути використана для математичної структури специфічного виду, то розробники створюють нову спеціальну програмну компоненту.

Панелі інструментів ПК “DSR Open Lab 1.0” є відображенням частини інтерфейсу користувача, який надає кожна програмна компонента. Панелі інструментів динамічно змінюють свій вигляд в залежності від типу математичної задачі, що досліджується, виду аналізу та компоненти, що є активною на даний момент.

На першому етапі розв'язку задачі чисельного аналізу в програмній системі “DSR Open Lab 1.0” відбувається визначення за допомогою навігатора моделі, типу математичної структури, яку збирається аналізувати користувач, та набору видів аналізу, які допускає обрана математична структура. Відмітимо, що користувач застосовує готове дерево вибору, побудоване на етапі розробки програмного комплексу чисельного аналізу. Це виключає можливість такого роду помилок взаємодії людини та комп'ютера, які трактуються в інших математичних програмах як помилки синтаксису та семантики відповідних предметно-орієнтованих мов.

Наступним етапом є опис конкретного екземпляру математичної структури (рис. 1). Вимоги до гнучкості програмного комплексу чисельного аналізу потребують застосування тут предметно-орієнтованих мов опису математичних задач. Однак запропоноване в даній роботі розмежування функцій опису задачі між програмним графічним інтерфейсом користувача та підсистемою лінгвістичного забезпечення (на основі використання типології математичних моделей технічних систем) дозволяє суттєво наблизити такі мови до природної мови математичного моделювання – математичної нотації. Це спрощує як вивчення, так і використання розробленого лінгвістичного забезпечення.

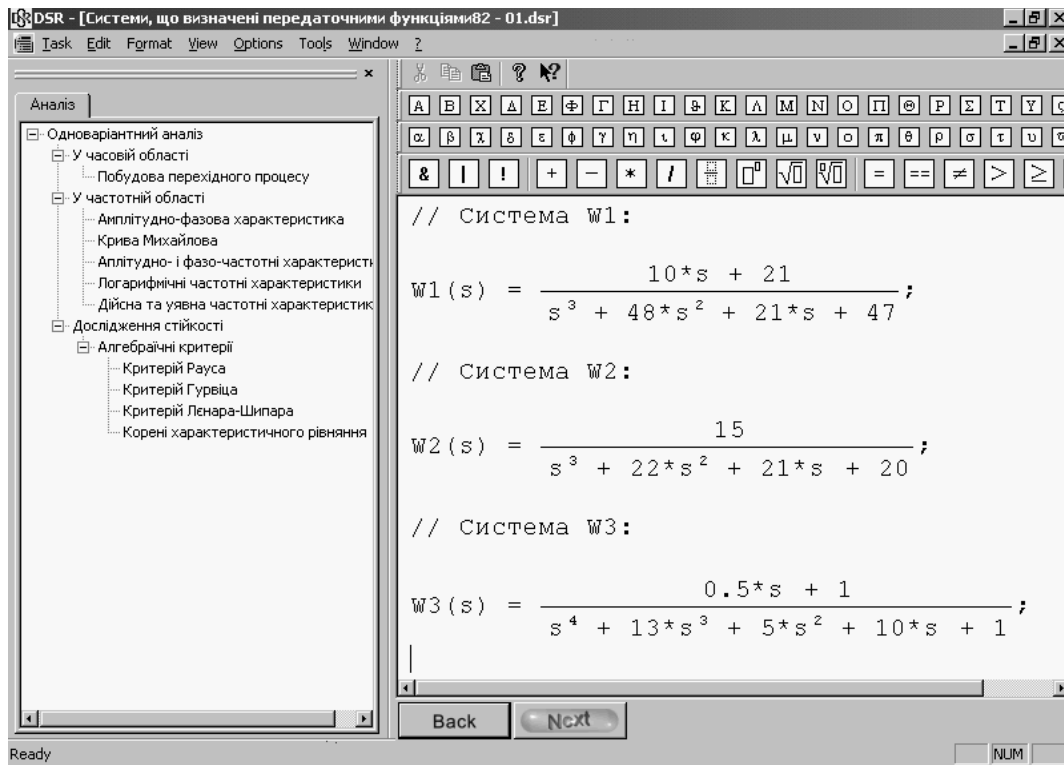


Рис. 1. Етап опису задачі для чисельного аналізу

Наступні стадії опису допомагають користувачу за допомогою графічного інтерфейсу вибрати методи розв’язку задачі, сформулювати умови, специфічні для конкретної задачі, та вибрати вид представлення результатів чисельного аналізу.

Останнім етапом розв’язку задачі чисельного аналізу є перегляд результатів у вигляді, визначеному на попередніх етапах (рис. 2).

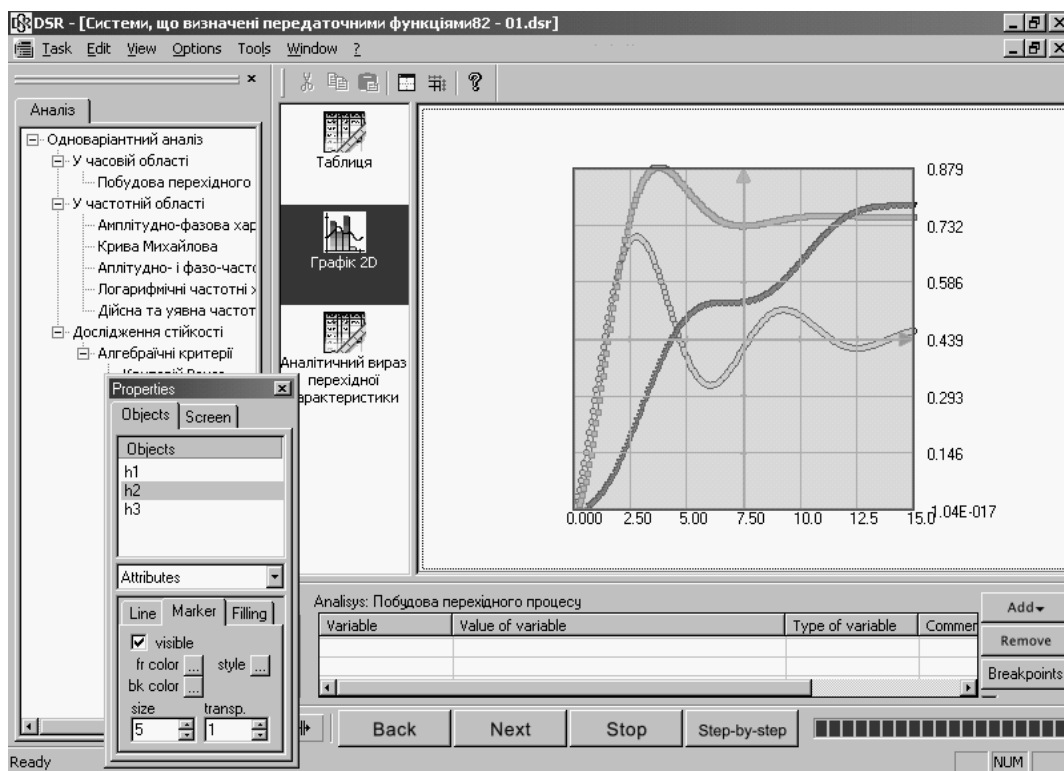


Рис. 2. Етап перегляду результатів

Як видно з даного короткого опису роботи користувача в програмній системі “DSR Open Lab 1.0”, завдяки звуженню функціональності лінгвістичного забезпечення формулювання умови задачі чисельного аналізу суттєво спрощується. Це стало можливим завдяки перенесенню структурної складності інтерфейсу з етапу виконання на етап розробки програмної системи чисельного аналізу.

Дійсно, використання для опису задачі чисельного аналізу більш складної універсальної мови (як це і зроблено в існуючих математичних програмах MATLAB, Maple V, Mathematica тощо) спростило б реалізацію графічного інтерфейсу програмної системи, однак примусило б користувача вивчати складну лексику, синтаксис та семантику мови та збільшила б вірогідність помилок. На противагу до такого підходу, кропітка розробка розгалуженого дерева вибору, яке спрямовує роботу користувача за допомогою меню, екранних форм та інших об’єктів графічного інтерфейсу, дозволяє приховати загальну складність задачі опису математичної структури та реалізувати її на рівні програмної архітектури.

Згідно із запропонованим підходом до організації взаємодії людина–комп’ютер в галузі чисельного аналізу математичних задач, предметно-орієнтовані мови опису “DSR Open Lab 1.0” є повністю декларативними та надають виключно засоби визначення певного математичного об’єкта, що, звичайно, суттєво полегшує вивчення та застосування програмного забезпечення в цілому та лабораторного практикуму з ТАК зокрема. Таким чином, запропонований поділ функцій між мовами опису математичних структур та рештою підсистем “DSR Open Lab 1.0”, що реалізують графічний інтерфейс з користувачем, дозволяє реалізувати прості та зрозумілі лексику, синтаксис і семантику мов у формі, максимально наближеній до звичної користувачам математичної нотації.

Переваги перенесення на програмний графічний інтерфейс функціональності, яка традиційно вважається в існуючих програмах чисельного аналізу математичних задач частиною предметно-орієнтованої мови, можна проілюструвати рядом прикладів роботи одного з найбільш потужних існуючих програмних інструментів Mathematica та представленого авторського лабораторного практикуму. Розглянемо далі та проаналізуємо шляхи розв’язку кількох характерних задач ТАК цими конкуруючими програмними продуктами (табл. 2).

Таблиця 2

Приклади опису задач класичної ТАК

Mathematica 1	“DSR Open Lab 1.0” 2
<i>Передаточна функція</i>	
Needs["ControlSystems`Master`"] g = TransferFunction[s, 20/(s(s+1)(s+4))]	$W(s) = \frac{20}{s(s+1)(s+4)}$;
<i>Побудова перехідної характеристики</i>	
Needs["ControlSystems`Master`"] g = TransferFunction[s, 20/(s(s+1)(s+4))] SimulationPlot[g, UnitStep[t], {t, 50}, PlotLabel -> "Step Response", PlotRange -> All]	$W(s) = \frac{20}{s(s+1)(s+4)}$;
<i>Побудова логарифмічних частотних характеристик</i>	
Needs["ControlSystems`Master`"] g = TransferFunction[s, 20/(s(s+1)(s+4))] BodePlot[g, {1, 10}, PlotPoints -> 200]	$W(s) = \frac{20}{s(s+1)(s+4)}$;
<i>Побудова амплітудно-фазової характеристики</i>	
Needs["ControlSystems`Master`"] g = TransferFunction[s, 20/(s(s+1)(s+4))] NyquistPlot[g, {0.01, 20}, PlotPoints -> 500]	$W(s) = \frac{20}{s(s+1)(s+4)}$;

З наведених вище прикладів можна зробити важливий для даної роботи висновок щодо порівняльної “дружності” інтерфейсу Mathematica та програмної системи “DSR Open Lab 1.0” (тобто власне інтерфейсу лабораторного практикуму з ТАК).

Розв’язок задачі за допомогою Mathematica (а зрештою й за допомогою решти подібних програмних інструментів) потребує від користувача: 1) вивчити задачу чисельного аналізу за допомогою існуючих підручників чи збірників задач; 2) вивчити складної структури предметно-орієнтовану мову Mathematica; 3) знайти потрібну функцію (це не так легко, як може здаватися на перший погляд – потрібні функції далеко не завжди знаходяться в одному розділі системи допомоги); 4) закодувати конкретну задачу користувача мовою Mathematica; 5) переглянути результати аналізу в жорсткому та непридатному для редагування вигляді, наперед визначеному розробником. При цьому, хоча б другий етап (вивчення мови), теоретично користувач проходить лише один раз (вперше освоюючи програму), відчутних плюсів це не приносить. По-перше, навіть одноразові затрати на вивчення предметно-орієнтованої мови Mathematica є занадто великими. По-друге, навіть розуміння загальної ідеології та правил синтаксису

предметно-орієнтованої мови не допоможе користувачу швидше знайти потрібну функцію серед багатьох сотень подібних функцій.

Розв'язок задачі за допомогою представленого лабораторного практикуму потребує лише визначення за допомогою програмного графічного інтерфейсу необхідного виду аналізу та скопіювати математичний запис, знайдений в підручнику, до вікна редагування опису задачі. При цьому не потрібно вивчати складну предметно-орієнтовану мову та марнувати час в пошуках серед статей системи допомоги потрібної лексики та синтаксису. Нам ввижається, що така ідеологія роботи з математичним програмним забезпеченням може здатися прогресом тим користувачам, яким набридло почувати себе кинутим напризволяще перед списком з тисячі функцій.

На наш погляд, порівняння роботи Mathematica та лабораторного практикуму на базі "DSR Open Lab 1.0" виявилось плідним та досить красномовним. За наведеним описом процедури розв'язку задачі чисельного аналізу кожен користувач вільний обрати шлях, який є близьким для нього. Авторська ж позиція в даному питанні є такою: можна стверджувати, що використання існуючих іноземних програмних систем чисельного аналізу не повністю відповідає вимогам до інтерфейсу користувача програмного забезпечення навчального курсу ТАК.

Таким чином, за результатами наведеної характеристики порівняльної ефективності застосування лабораторних практикумів з ТАК для представленого авторського програмного забезпечення та іноземних аналогів можна зробити висновки щодо придатності розробленого авторами програмного забезпечення для підготовки студентів вищих навчальних закладів. При цьому представлений авторський лабораторний практикум з ТАК загалом не поступається, а іноді переважає існуючі аналоги в зручності інтерфейсу користувача та функціональності.

ЛІТЕРАТУРА:

1. Колодницький Н.М., Левицький В.Г. Обзор основных программных средств для моделирования математических задач // САПР и графика, 1999. – № 10 – С. 56–65.
2. MathSoft, Inc. Mathcad User's Guide. Cambridge, MA: 2000.
3. Потемкин В.Г. Система MATLAB 5 для студентов. – М.: ДИАЛОГ-МИФИ, 1998. – 314 с.
4. Потемкин В.Г. Система MATLAB: Справочное пособие. – М.: ДИАЛОГ-МИФИ, 1998. – 350 с.
5. John H. Mathews, Kurtis Fink. Numerical Methods Using MATLAB. 3rd Edition, Pearson Education, 1999. – P. 680.
6. MATLAB User's Manual, MATHWORKS Inc, MA, USA, 1999.
7. Gaylord R.J., Kamin S.N. and Wellin P.R. Programming with Mathematica. 2nd Edition, Springer-Verlag, TELOS, Santa Clara, 1996.
8. Stephen Wolfram. The Mathematica Book, Fourth Edition. Cambridge University Press, 99, pp. 1470.
9. Monagan M.B., Geddes K.O., Heal K.M., Labahn G., Vorkoetter S.M. Maple V Release 5. Learning Guide. Springer-Verlag, 1998. – 284 p.
10. Monagan M.B., Geddes K.O., Heal K.M., Labahn G., Vorkoetter S.M. Maple V Release 5. Programming Guide. Springer-Verlag, 1998. – 379 p.
11. Колодницький М.М. Тривимірна компонентна архітектура прикладної програмної системи "DSR Open Lab 1.0" як втілення концепцій реінженерії // Теорія програмування. – 1998. – № 4. – С. 37–45.
12. Колодницький М.М. Програмна система "Dsr Open Lab 1.0": сучасний засіб автоматизації математичного моделювання // Вісник ЖІТІ. – 2000. – № 14. – С. 198–203.
13. Kolodnytsky M., Kovalchuk A., Kuryata S., Levitsky V. The Mathematical Software Implementation for Computational Algebra and Number Theory // Computer Mathematics. Proceedings of the 4th Asian Symposium. Chiang Mai, Thailand, December 17–21, 2000. / Lecture Notes Series on Computing. Volume 8. – World Scientific, Singapore. – P. 291–294.
14. Kolodnytsky M., Kovalchuk A., Kuryata S., Levitsky V. "A Theory of Approximation of Functions" Courseware // ATCM 2000. Proceedings of the 5th Asian Technology Conference in Mathematics, December 17-21, 2000. Chiang Mai, Thailand. / Published by ATCM, Inc., USA. – P. 86–95.
15. Kolodnytsky M., Kovalchuk A., Kuryata S., Levitsky V., Samolyuk I. Interactive Visual Software System for Problem Solving and Teaching in Finite Group Theory // Proceedings of the 23rd International Conference on Information Technology Interfaces, Pula, Croatia, 2001. – P. 409–416.
16. Левицький В.Г. Розробка лінгвістичного забезпечення спеціалізованої програмної системи чисельного аналізу // Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 01.05.03 – "Математичне та програмне забезпечення обчислювальних машин і систем". – Інститут проблем моделювання в енергетиці НАН України, Київ, 2001.
17. Левицький В.Г. Розробка мов опису задач математичного моделювання в програмному комплексі "Dsr Open Lab 1.0" // Вісник ЖІТІ/ – 2000. – № 15. – С. 195–202.

КОВАЛЬЧУК Андрій Михайлович – асистент кафедри автоматизації і комп'ютеризованих технологій Житомирського інженерно-технологічного інституту.

Наукові інтереси:

- комп'ютерні інформаційні технології;
- використання обчислювальної техніки в навчальному процесі;
- методи організації графічних інтерфейсів користувача прикладних програмних систем;
- комп'ютерна графіка.

ЛЕВИЦЬКИЙ В'ячеслав Георгійович – кандидат технічних наук, доцент кафедри програмного забезпечення обчислювальної техніки Житомирського інженерно-технологічного інституту.

Наукові інтереси:

- комп'ютерні інформаційні технології;
- моделювання та розв'язок задач за допомогою обчислювальної техніки;
- використання обчислювальної техніки в навчальному процесі;
- побудова компіляторів.

САМОЛЮК Ігор Іванович – студент V курсу факультету інформаційно-комп'ютерних технологій Житомирського інженерно-технологічного інституту.

Наукові інтереси:

- комп'ютерні інформаційні технології;
- моделювання та розв'язок задач за допомогою обчислювальної техніки;
- використання обчислювальної техніки в навчальному процесі;
- теорія автоматичного керування.

Подано 12.04.2002

Ковальчук А.М., Левицкий В.Г., Самолюк І.І. Комп'ютерний лабораторний практикум з теорії автоматичного керування

Ковальчук А.М., Левицкий В.Г., Самолюк И.И. Компьютерный лабораторный практикум по теории автоматического управления

Kovalchuk A.M., Levitsky V.G., Samolyuk I.I. Courseware on theory of automatic control

УДК 004.415

Компьютерный лабораторный практикум по теории автоматического управления / Ковальчук А.М., Левицкий В.Г., Самолюк И.И.

В статье представлено описание некоторых особенностей проектирования, реализации и использования программного средства для разработки компьютерного лабораторного практикума по теории автоматического управления. Проведено сравнение характеристик разработанного программного обеспечения лабораторного практикума с известными аналогами. Рассмотрено эффективность использования программного обеспечения в процессе подготовки студентов высших учебных заведений.

УДК 004.415

Courseware on theory of automatic control / A.M. Kovalchuk, V.G. Levitsky, I.I. Samolyuk

The description of author's courseware on theory of automatic control has been presented. The comparison of courseware characteristics with similar systems has been done and the efficiency of usage of the designed software in students training is considered.