

## РОЗРОБКА ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ MATLAB/SIMULINK ДЛЯ ПРОЕКТУВАННЯ КЕРУЮЧИХ ПРИСТРОЇВ СИСТЕМ УПРАВЛІННЯ НА БАЗІ ПЛІС

У статті розглядається проблема створення інструментальних засобів для використання методології швидкого прототипування та сумісного моделювання при розробці керуючих пристроїв систем управління. Детально розглянуто мету, засоби і переваги підходу швидкого прототипування. Також розглянуто зв'язок швидкого прототипування та сумісного моделювання, переваги їх використання у комплексі. Як рішення запропоновано універсальний спосіб зв'язку між середовищем моделювання та апаратним прототипом.

Результатом роботи є розроблена програма, що дозволяє використовувати апаратні прототипи керуючих пристроїв, створені на ПЛІС, для сумісного моделювання з моделлю системи створеної у програмному пакеті Matlab.

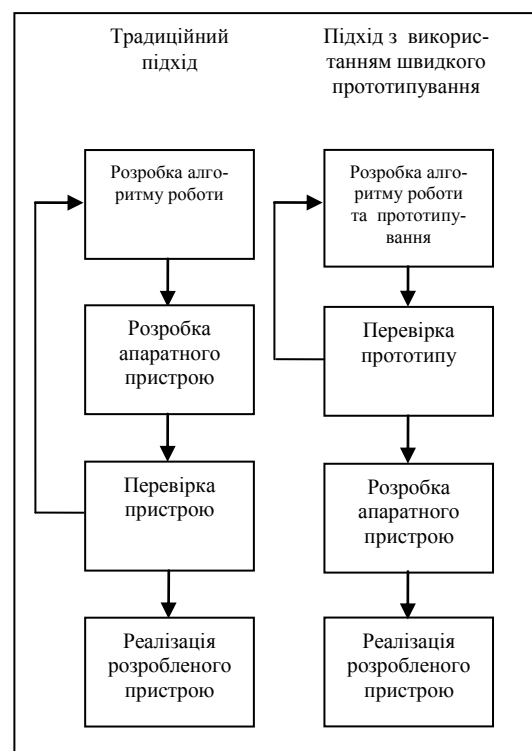
This article addresses the problem of creating a design environment for rapid prototyping and co-simulation to be used in field of control systems. Purposes, techniques and advantages of rapid prototyping approach are covered in details. As well a co-simulation approach is covered in conjunction with rapid prototyping; the advantages of using them in combination are presented. A universal way of communication between a simulation environment and a prototype is proposed as a part of the solution.

The study resulted in a development of a program that allows hardware prototypes implemented in FPGA to be used for co-simulation with a model of a system created in MATLAB/Simulink environment.

### Вступ

Метою підходу швидкого прототипування є створення прототипів нових виробів на ранніх етапах розробки. Прототипи, зазвичай, реалізують тільки частину потрібної функціональності, або не відповідають усім вимогам до кінцевого виробу, щоб прискорити процес їх створення. Використання прототипів дозволяє, поперше, виявити можливі недоліки та виправити їх на ранніх стадіях розробки, а по-друге, скорегувати подальші плани розробки в залежності від змін у вимогах проекту, які можуть бути сформульовані на основі випробувань прототипу. При такому підході розробка ведеться ітеративно, що додає гнучкості процесу. Загалом це пришвидшує цикл розробки кінцевого продукту та дозволяє зменшити час потрібний для завершення проекту, що є вигідним при дрібносерійному виробництві та виробництві у висококонкурентних галузях промисловості. Різницю між традиційним підходом та підходом з використанням швидкого прототипування можна побачити на рисунку 1.

Розглянемо технологію швидкого прототипування у зв'язку з розробкою електронної апаратури. Є декілька засобів, які можуть бути використані для створення прототипів електронної апаратури. Це мікроконтролери, програмовані логічні пристрої та програмовані логічні інтегральні схеми.



**Рис. 1. Схеми традиційного підходу та підходу з використанням швидкого прототипування**

Від спеціалізованих інтегральних схем, які зазвичай використовуються при виробництві кінцевої продукції, їх відрізняє одна спільна особливість – можливість швидкої зміни функціональності, що і є необхідним при створенні прототипів. Завдяки стабільному прогресу у області надвеликих інтегральних схем та облас-

ті засобів проектування електронних пристроїв, ПЛІС досягли рівня при якому їх можна вважати передовим засобом розробки електронних пристроїв. Вони мають переваги над іншими засобами прототипування, маючи більшу швидкість та ефективність роботи відносно до вартості. Саме тому, в останні роки ПЛІС використовуються, як у промисловості, для створення прототипів, так і в науці, для дослідження ефективності їх використання у різних областях застосування.

Саме можливість запрограмувати у ПЛІС довільну функціональність, поєднана з високою швидкістю отриманого пристрою робить їх надзвичайно вигідними для створення регуляторів для систем управління та перевірки роботи складних алгоритмів управління.

Отримавши прототип пристрою необхідно перевірити його функціонування та виявити недоліки у його роботі. Для цього необхідно перевірити його працездатність у робочих умовах, наприклад, використавши у справжній системі або макеті системи. Однак, коли існує досить детальна модель системи, розроблена у спеціалізованих програмних пакетах, або при занадто дорогій розробці макету системи, можливо провести моделювання програмної моделі системи разом із розробленим прототипом.

Такий підхід до моделювання отримав назву сумісного моделювання. Він полягає у поєднанні обчислювальних засобів різного походження, для створення загальної моделі. Апаратний блок обчислень, наприклад, створений раніше прототип, за допомогою каналу зв'язку підключається до моделі складної системи, і, коли потрібно провести обчислення закладені у апаратній частині, через канал зв'язку на неї надсилаються вхідні данні, пристрій проводить складні обчислення та через канал зв'язку відсилає назад вихідні данні. Каналом зв'язку може бути будь-який з інтерфейсів, які підтримують апаратний засіб та середовище моделювання.

### Огляд матеріалів

Проблемі створення прототипів пристроїв, а також розробці середовища для сумісного моделювання, яке задовольнило б вимогам науковців та інженерів присвячені наступні роботи.

У роботі [1] у загальному вигляді описано процес використання MATLAB/Simulink як засобу розробки пристроїв на ПЛІС. Як приклад розглянуто створення передавача стандарту

HSDPA, описано загальний процес перетворення моделі MATLAB/Simulink у код на мовах опису апаратного забезпечення.

Стаття [2] містить приклад розробки регулятора для асинхронного електричного двигуна у MATLAB/Simulink. Як засіб перетворення моделі на код на мові опису апаратного забезпечення використано програмний пакет Xilinx System Generator, який дозволяє створити модель для ПЛІС фірми Xilinx.

У [3] описується приклад зв'язку між MATLAB/Simulink та демо-платою з ПЛІС фірми Altera за допомогою програмних засобів фірми Celoxica.

У статтях [4], [5] наведено приклади побудови ПІД-регуляторів за допомогою програмного пакету Xilinx System Generator. У [5] детально розглянута побудова ПІД-регулятора та проведено аналіз використання ресурсів ПЛІС таким регулятором.

Робота [6] детально розглядає використання ПЛІС для прототипування регуляторів. Розглянуто методику створення регуляторів з можливістю обробки аналогових даних за допомогою Xilinx System Generator, а також інтеграцію створених регуляторів у Систему-на-чипі (SoC) з написанням інтерфейсу для обміну даними з ПК з ОС Linux.

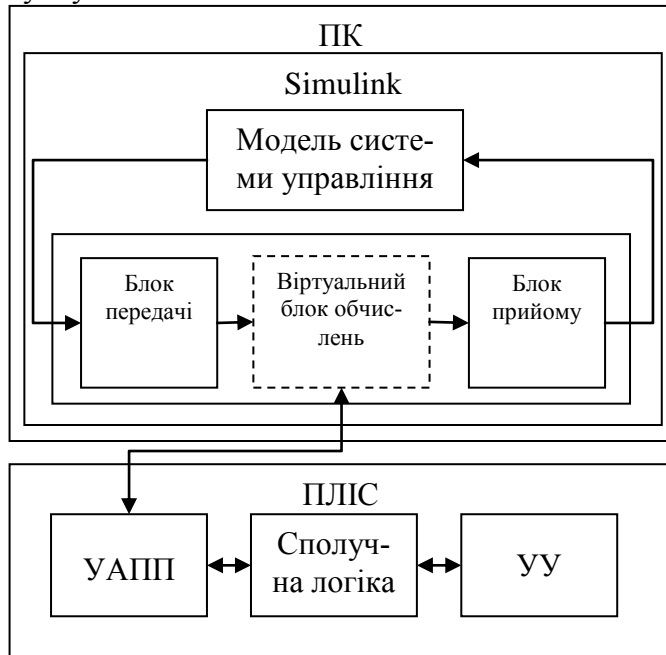
### Постановка проблеми

Як видно з огляду приведених вище робіт, на даний момент не існує єдиного загального способу забезпечення комунікацій між апаратним пристроєм, реалізованим на ПЛІС, та моделлю на ПК, реалізований у програмному пакеті MATLAB. Існуючі способи обмежені у своєму використанні до ПЛІС окремих виробників, або до демонстраційних плат з ПЛІС окремих моделей. До того ж більшість середовищ сумісного моделювання вимагають використання окремих програмних засобів, ускладнюючи створення такого середовища. В даній роботі буде розглянуто спосіб створення середовища для сумісного моделювання на основі програмного пакету MATLAB, що є універсальним у сенсі незалежності від моделі та марки використовуваної ПЛІС, та дозволить провести сумісне моделювання з прототипом створеним за допомогою MATLAB\Simulink HDL Prototype Control Toolbox[7].

## Вирішення проблеми

Основи цієї розробки закладено у статті [7]. У ній описано створення засобів та загальної методики перетворення векторно-матричних моделей керуючих пристроїв у код на мові Verilog, який потім може використовуватись для програмування ПЛІС.

Для сумісного моделювання разом з моделлю у Simulink код регулятора поєднується з підсистемою, що забезпечує обмін даними з ПК. Структурна схема рішення представлена на рисунку 2.



**Рис. 2. Структурна схема середовища сумісного моделювання на базі MATLAB/Simulink**

На боці комп'ютера знаходиться пакет Simulink із зібраною в ньому схемою передачі і прийому квантованих за часом значень величин впливів. У цей час ПЛІС налаштована на виконання таких функцій:

- прийом і передача даних;
- обробка (обчислення) даних.

За прийом і передачу відповідає модуль універсального асинхронного приймача-передавача (УАПП – UART). Параметри режиму передачі задаються до конфігурування пристрою, на стадії опису апаратного рівня.

Сполучна логіка (СЛ) об'єднує заздалегідь задану кількість послідовно переданих байт в одне слово, яке являє собою одне, квантоване за часом, значення, прийняте від ПК. Даний модуль також проводить зворотні перетворення над обробленими даними для їх передачі на ПК. Після формування одного слова даних керуючий модуль виставляє це значення на вхідну шину обчислювального модуля і видає одиноч-

ний імпульс, тривалістю в один період тактової частоти, на вхід дозволу обчислення. Вже на наступному такті обчислювальний модуль виставляє на вихідну шину обчислене значення, яке згодом розбивається на байти і передається на ПК. Після виділення одного байта зі слова даних модуль сполучної логіки виставляє на вхідні шину модуля УАПП даний байт і формує одиночний імпульс дозволу передачі. По завершенні передачі чергового байта модуль УАПП інформує модуль СЛ про закінчення передачі, що дає останньому можливість вибору наступної дії: або передачі наступного байта або переходу в режим очікування, якщо все слово даних було передано.

Модуль сполучної логіки реалізований у вигляді кінцевого автомата типу Мілі для забезпечення синтезування вищеописаного алгоритму.

Параметр довжини слова в байтах, який є умовою для переходу в станах прийому і передачі даних, встановлюється у конфігураційному файлі при генерації моделі.

Обчислювальним модулем виступає блок «HDL State-Space» трансльований в код на мові Verilog[7] й об'єднаний з керуючою програмою.

Для доведення можливості функціонування приведеної вище системи її було реалізовано у експериментальному режимі. При цьому, як за-сіб зв'язку було обрано широко поширений інтерфейс RS-232, що забезпечує обмін даних між ПЛІС та ПК. До недоліків такої реалізації слід віднести невелику швидкість передачі даних, яка не дозволяє системі працювати в режимі реального часу. Але, слід зазначити, що структура системи побудована таким чином, що, по-перше, прототип не залежить від реалізації інших частин системи, тобто він може використовуватись, як з моделлю на ПК, так і з моделлю створеною на апаратних компонентах, та, по-друге, при необхідності легко може бути змінений інтерфейс, що поєднує прототип з іншими частинами системи, тобто можливо реалізувати інтерфейси, які дозволять всій системі працювати в режимі реального часу, і це не буде вимагати внесення змін до структури системи.

## Приклад використання

Для прикладу проведемо сумісне моделювання двовимірної системи з наявними неконтрольованими збуреннями (із застосуванням фільтру Калмана).

Об'єкти системи задамо у векторно-матричному вигляді, яке буде відповідати рівнянням (1), де  $A, B, C, D$  – матриці,  $X(t)$  – вектор станів об'єкту,  $U(t)$  – вектор управління,  $Y(t)$  – вектор виходів об'єкту.

$$\begin{aligned} \dot{X}(t) &= AX(t) + BU(t), \\ Y(t) &= CX(t) + DU(t). \end{aligned} \quad (1)$$

Система задана за допомогою векторно-матричних моделей об'єкта управління та керуючого пристрою (розрахунок виконувався окремо), (2) та (3), відповідно.

$$\begin{aligned} A &= \begin{bmatrix} -3 & 2 \\ 2 & -6 \end{bmatrix}, B = \begin{bmatrix} 3 & 2 \\ 4 & 1 \end{bmatrix}, \\ C &= \begin{bmatrix} 0.4 & 0 \\ 0 & 1.2 \end{bmatrix} \end{aligned} \quad (2)$$

$$\begin{aligned} A_r &= \begin{bmatrix} 0 & 0 & -33.3 & 0 \\ 0 & 0 & 0 & -100 \\ 3 & 0 & -20.547 & -3.5921 \\ 0 & 1 & -0.3991 & -23.355 \end{bmatrix}, \\ B_r &= \begin{bmatrix} 83.333 & 0 & 0 & 0 \\ 0 & 83.333 & 0 & 0 \\ 0 & 0 & 1.2617 & 2.9934 \\ 0 & 0 & 0.9978 & 2.7125 \end{bmatrix}, \\ C_r &= \begin{bmatrix} -0.6 & 0.4 & 2.6 & -5.2 \\ 2.4 & -0.6 & -12.4 & 6.8 \end{bmatrix}, \\ D_r &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (3)$$

Модель системи у MATLAB\Simulink матиме вигляд як на рисунку 3.

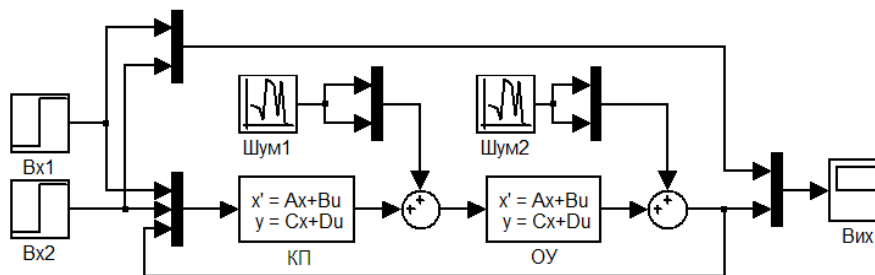


Рис. 3. Схема системи з фільтром Калмана

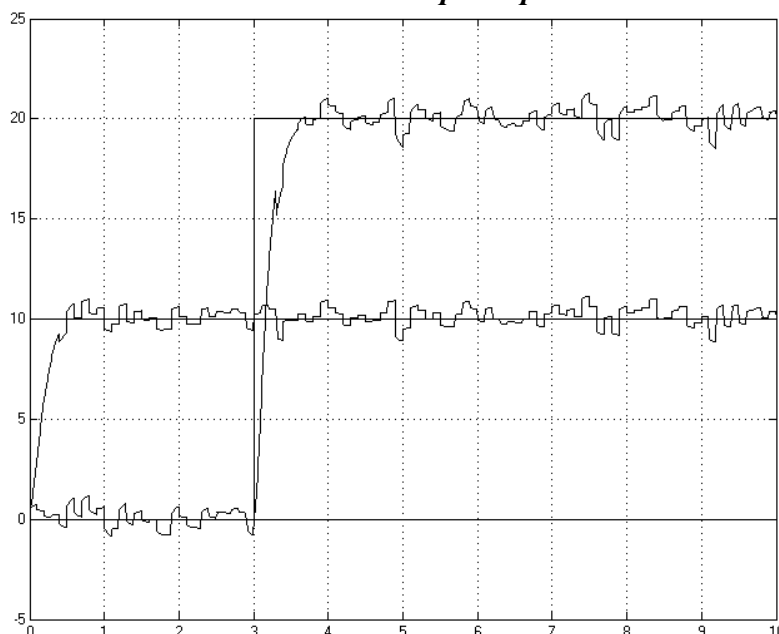


Рис. 4. Перехідні процеси у системі

Наступним кроком додамо до моделі блок з розробленої бібліотеки інструментальних засобів MATLAB\Simulink HDL Prototype Control Toolbox, за допомогою якого створимо програму для ПЛІС. Далі отримаємо можливість генерувати програму функціонування керуючого

пристрою для ПЛІС за допомогою Simulink HDL Coder[7], об'єднаємо отриману програму із сполучною логікою та згенеруємо файл прошивки для ПЛІС. Під час синтезу логіки для ПЛІС було отримано дані про витрати апаратних ресурсів ПЛІС обраної моделі, що приве-

дені у таблиці 1. Ці дані свідчать про те, що ресурси ПЛІС було використано приблизно на 75%, та подальше ускладнення регулятора є

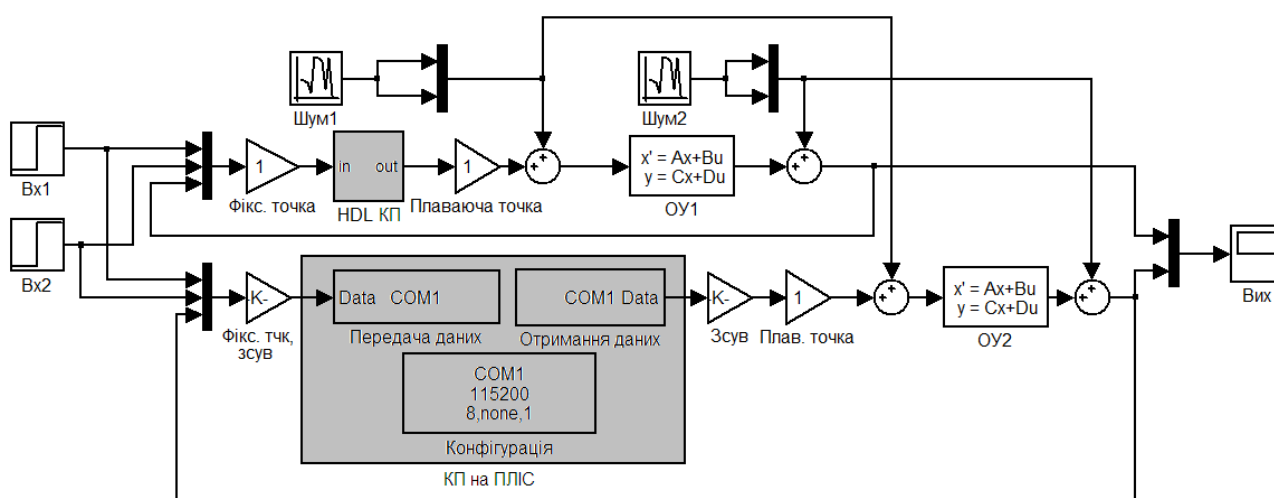
неможливим без певної оптимізації, або використання мікросхеми ПЛІС з більшими апаратними ресурсами.

**Табл. 1. Витрати апаратних ресурсів ПЛІС та обмеження тактової частоти**

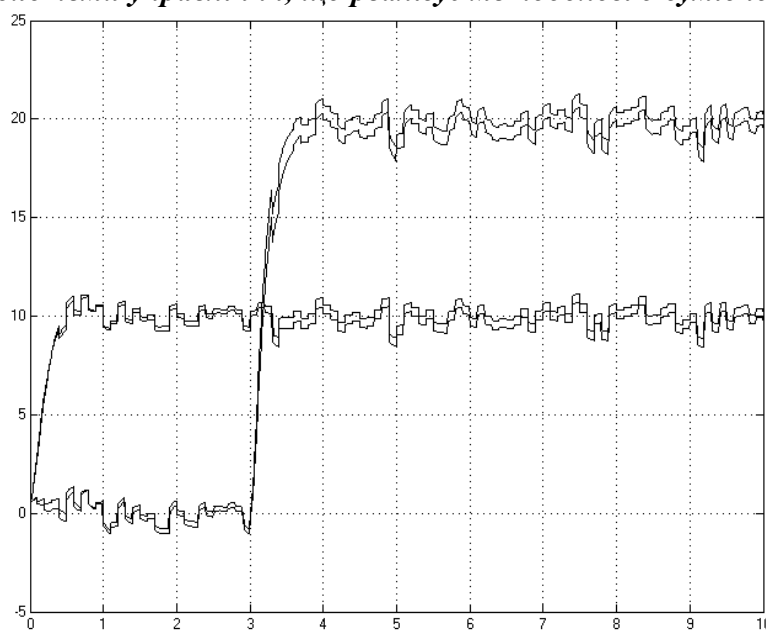
Ресурс	Використано	Доступно	Процент використання
Логічні таблиці	5 119	7 168	71%
Конфігуровані блоки	2 685	3 584	74%
Блоки множення	16	16	100%
Мультиплексори	2	8	25%
Блоки вводу-виводу	12	264	4%
Тактова частота	22 МГц	33 МГц	66%

Отримавши бінарний файл, запрограмуємо ПЛІС, у нашому випадку Xilinx Spartan 3E, створеною програмою.

Після цього модифікуємо модель Simulink таким чином, щоб забезпечити обмін даними між Simulink та ПЛІС, та виконаємо моделювання.



**Рис. 5. Модель системи управління, що реалізує методологію сумісного моделювання**



**Рис. 6. Результати сумісного моделювання системи**

Результати моделювання свідчать, що апаратна система веде себе аналогічно до моделі у Simulink з невеликою розбіжністю, що обумовлена використанням у ПЛІС типів даних з обмеженою точністю.

## Висновки

Результатом роботи є середовище інструментальних засобів MATLAB\Simulink для розро-

бки прототипів керуючих пристроїв, та проведення їх сумісного моделювання на базі програмного пакету MATLAB. Запропоновані засоби забезпечують роботу в рамках одного програмного середовища для роботи з моделлю системи та незалежність від виробника та марки ПЛІС.

Подальшою перспективою досліджень може бути розробка інфраструктурних блоків для Simulink та покращення сполучної логіки для додання інших інтерфейсів комунікації між Simulink та ПЛІС, а також автоматизація програмування ПЛІС з загального середовища розробки.

### Список літератури

1. Tejas M. Bhatt, Dennis McCain. Matlab as a Development Environment for FPGA Design // Proceedings of the 42nd annual Design Automation Conference.– New York, NY, USA.– 2005.– 607.
2. R.Rajendran, Dr.N.Devarajan. FPGA Based Implementation of Space Vector Modulated Direct Torque Control For Induction Motor Drive // International Journal of Computer and Electrical Engineering. – June, 2010. – Vol. 2, No. 3. – 1793-8163
3. Bartosinski, R., Kadlec, J. Hardware Co-simulation with Communication Server from MATLAB/Simulink // KONFERENCE TECHNICAL COMPUTING. – PRAGUE. – 2006.
4. Yuen Fong Chan, Moallem M., Wei Wang. Design and Implementation of Modular FPGA-Based PID Controllers // Industrial Electronics, IEEE Transactions on. – 2007. – Volume 54 Issue:4 – 1898 – 1906
5. The use of FPGA in field-oriented control of an induction machine, Ozkan AKIN, Irfan ALAN
6. Stephane Simard, Jean-Gabriel Mailloux, and Rachid Beguenane. Prototyping Advanced Control Systems on FPGA // EURASIP Journal on Embedded Systems. – 2009.– Volume 2009.– 12.
7. Писаренко А.В., Репнікова Н.Б., Цвелодуб А.О. Розробка Verilog/VHDL орієнтованого блоку MATLAB/Simulink // Наукові вісті Галицької академії. – Івано-Франківськ: ГА, 2010. – № 17(1). – С.115-121.