УДК 519.72

А. М. Одейчук, аспірант кафедри інформаційних управляючих систем
Харківського національного університету радіоелектроніки
А. Ю. Гуд, аспірант кафедри інформаційних управляючих систем
Харківського національного університету радіоелектроніки

ІМІТАЦІЙНА МОДЕЛЬ ДЛЯ ДОСЛІДЖЕННЯ ГЕТЕРОСКЕДАСТИЧНИХ ЧАСОВИХ РЯДІВ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

У статті розглянуто проблему вибору та побудови адекватної моделі прогнозування в ІС з урахуванням статистичних характеристик вихідних даних. Розроблено імітаційну модель для генерації гетероскедастичних часових рядів із заданими статистичними характеристиками.

В статье рассмотрена проблема выбора и построения адекватной модели прогнозирования в ИС с учетом статистических характеристик исходных данных. Разработана имитационная модель для генерации гетероскедастических временных рядов с задаваемыми статистическими характеристиками.

In work the problem of a choice and construction of adequate model of forecasting in IMS taking into account statistical characteristics of initial data is considered. The simulation model is developed for generation heteroscedasticity time series with set statistical characteristics.

Ключові слова. Інформаційні системи, імітаційна модель, моделі прогнозування.

Вступ. Нині аналіз даних, що полягає в побудові моделей прогнозування техніко-економічних показників, поданих у вигляді часових рядів, є невід’ємною частиною сучасних ІУС, ДІС, АСУ тощо [1]. Через існування великої кількості методологій, що використовуються для прогнозування різних видів часових рядів, дуже складно визначити структуру моделі прогнозування. Зокрема, багато вчених, інженерів і наукових груп зосередили свою увагу на проблемі вибору та побудови адекватних моделей прогнозування гетероскедастичних часових рядів (ГЧР) [2–4].

Постановка завдання. Один з перспективних напрямків дослідження для розв’язання цієї задачі – імітаційне моделювання. Тому досить актуально розробити імітаційну модель для генерації гетероскедастичних часових рядів (ГГЧР) з високою швидкістю розрахунку. Вона дозволить науковцям детальніше проаналізувати якісні характеристики гетероскедастичних часових рядів і розробити аналітичну процедуру вибору моделі прогнозування.

Результати дослідження.

Імітаційна модель генерації гетероскедастичних часових рядів

Розробимо імітаційну модель, що дозволяє здійснити генерацію гетероскедастичних часових рядів із заданими статистичними характеристиками на основі GARCH моделі. З огляду на те, що дослідження статистичних характеристик часових рядів пов’язано з генерацією досить великих обсягів даних, щоб висновки, які грунтуються на їх аналізі, були статистично значущими і достовірними, основною вимогою до створюваної імітаційної моделі є мінімізація часу, що витрачається на виконання розрахунків.

Узагальнена структура

Узагальнена блок-схема запропонованої імітаційної моделі може бути зображена у вигляді сукупності функціональних блоків (рис. 1), на базі яких виконується генерація гетероскедастичних часових рядів із заданими статистичними характеристиками:

1. Блок завдання початкових значень.
2. Блок генерації псевдовипадкових чисел з рівномірним розподілом.
3. Блок генерації псевдовипадкових чисел із заданим законом розподілу.
4. Блок генерації часового ряду з умовною змінною дисперсією.
5. Блок формування результатів імітаційного моделювання.

У блоці 1 виконується ініціалізація змінних моделювання і встановлюються початкові значення параметрів, наведених у табл. 1, на основі яких буде виконано генерацію гетероскедастичного часового ряду.

У блоках 2 і 3 здійснюється генерація псевдовипадкових чисел.

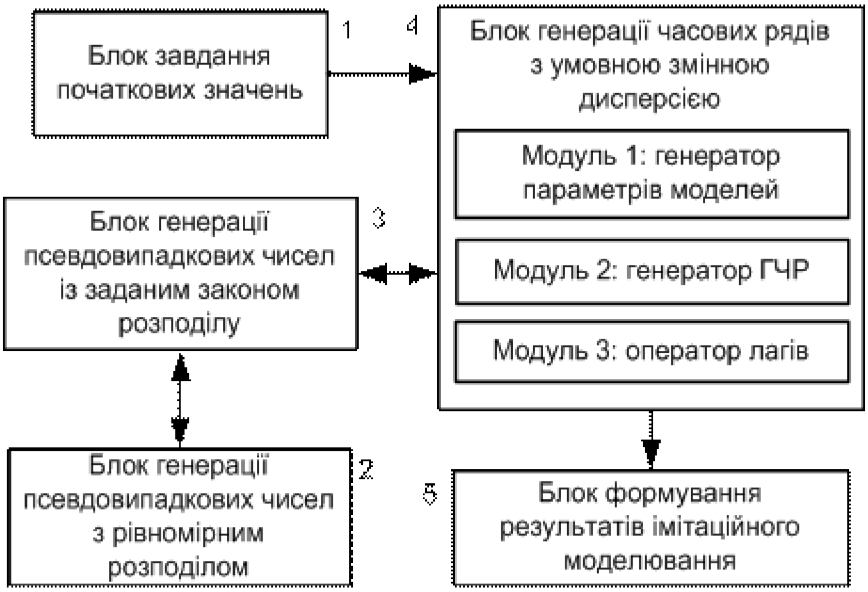

Рис. 1. Структура імітаційної моделі

У блоці 4 виконується генерація гетероскедастичного часового ряду, що відповідає заданим значенням моделювання. Згенерований часовий ряд потім передається у блок 5, в якому формуються результати проведеного моделювання.

Програмна реалізація імітаційної моделі виконана в математичному пакеті MatLab. Її блоки і відповідні алгоритми описано нижче.

Блок завдання початкових значень

У цьому блоці формується матриця плану експерименту на основі варійованих статистичних характеристик, перелік яких та діапазони набутих ними значень наведено в табл. 1, де *Z, Q* – множини цілих і раціональних чисел відповідно.

Блок генерації псевдовипадкових чисел

Для генерації випадкових величин, розподілених за рівномірним законом, існує ряд генераторів псевдовипадкових чисел (ГПВЧ), які реалізовано практично в усіх компіляторах мов програмування і статистичних програмних пакетів, а також на програмному рівні або з використанням апаратних ГПВЧ.

Найбільшого поширення одержали ГПВЧ, що базуються на лінійному конгруентному методі (період 2^{31}) та методі Фібоначчі із запізненнями [5]. У математичному пакеті MatLab 7 стандартний ГПВЧ використовує лінійно-конгруентний метод, який ґрунтується на рекурентному співвідношенні (1), де кожне наступне число ξ_t залежить від попереднього ξ_{t-1} [6, 7].

Таблиця 1

Параметри моделювання

№	Найменування параметра	Діапазон значень
1	Математичне сподівання випадкової величини ξ_t	$(-\infty, +\infty) \in Q$
2	Дисперсія випадкової величини ξ_t	$(0, +\infty) \in Q$
3	Ступеневий параметр λ	$(0, +\infty) \in Q$
4	Коефіцієнт середнього рівня дисперсії	$(0, +\infty) \in Q$
5	Коефіцієнт завдання скінченності умовної дисперсії	$(0, 1) \in Q$
6	Кількість α_j параметрів (*a*)	[illegible]
7	Кількість β_j параметрів (*b*)	$(1, +\infty) \in Z$
8	Сума α_j параметрів	$(0, 1] \in Q$
9	Сума β_j параметрів	$[0, 1] \in Q$
10	Довжина гетероскедастичного часового ряду	$[1, +\infty) \in Z$

$$\xi_t = \left(7^5 * \xi_{t-1}\right) * \mathrm{mod}\left(2^{31} - 1\right). \qquad (1)$$

Серед сучасних ГПВЧ відомий "Вихор Мерсенна" ("Mersenne Twister"), розроблений у 1997 р. Макото Мацумото і Такудзі Нісімура [8]. Цей ГПВЧ ґрунтується на властивостях простих чисел Мерсенна ($M_n = 2^n - 1$, де n – натуральне число); алгоритм наведено в праці [9]. Він забезпечує швидку генерацію псевдовипадкових чисел та ефективно використовує пам'ять ЕОМ (порівняння ГПВЧ "Вихор Мерсенна" з іншими сучасними ГПВЧ описано в працях [9, 10]). Цей ГПВЧ позбавлений багатьох недоліків, властивих іншим, а саме малого періоду, передбачуваності, статистичної залежності, що легко виявляється. Він має період, який дорівнює числу Мерсенна $2^{19937} - 1$, а кореляція між послідовними значеннями у вихідній послідовності "Вихору Мерсенна" дуже мала [9].

Тому блок ГПВЧ в імітаційній моделі доцільно базувати на генераторі "Вихору Мерсенна". У MatLab 7.4 набір ГПВЧ розширено "Вихором Мерсенна" [11], у версії MatLab 7.6 це стандартний ГПВЧ. Для більш ранніх версій MatLab можна скористатися безкоштовною бібліотекою, що реалізує цей ГПВЧ з інтернет-ресурсу [8], і командою loadlibrary.

Блок генерації псевдовипадкових чисел із заданим законом розподілу

Для одержання послідовності чисел з функцією розподілу $F(x)$ на основі псевдовипадкових чисел, що генеруються, розподілених за рівномірним законом на інтервалі [0, 1], використовуватимемо типовий метод обернених функцій [12]. Суть цього методу полягає в знаходженні оберненої функції розподілу $\xi = F^{-1}(\eta)$ щодо відомої $\eta = F(\xi)$. Генеруючи рівномірно розподілені величини $\eta_1, \eta_2, \ldots, \eta_k$ в інтервалі [0, 1] і використовуючи обернену функцію, одержимо відповідну послідовність $\xi_1, \xi_2, \ldots, \xi_k$, яка має необхідний закон розподілу.

З огляду на те, що аналітичне вираження оберненої функції нормального розподілу не можна знайти через відсутність аналітичного вигляду для $\xi = F^{-1}(\eta)$, то для одержання нормально розподілених випадкових чисел пропонується використовувати методи, які грунтуються на апроксимації функції нормального розподілу (ziggurat method) [13]. Цей метод реалізує функція MatLab – randn.

Блок генерації часового ряду з умовною змінною дисперсією

Нелінійний процес GARCH (p, q, λ) можна подати у вигляді співвідношень (2–3) [3]:

$$\xi_t \sim i.i.d(0, 1),$$
$$\varepsilon_t = \xi_t \sigma_t, \tag{2}$$

$$\sigma_t^2 = \omega + \sum_{j=1}^{p} \alpha_j \sigma_{t-j}^2 + \sum_{j=1}^{q} \beta_j |\varepsilon_{t-j}|^2, \tag{3}$$

де σ_t^2 – умовна за передісторією дисперсія ε_t, яку також часто називають волатильністю процесу. Абревіатура *i.i.d* (independent, identically disturbed sample) означає, що величини ξ_t незалежні та мають однаковий розподіл.

Щоб умовна дисперсія залишалась додатною, потрібно виконати співвідношення $\omega > 0$, $\alpha_1, \ldots, \alpha_p \geq 0$, $\beta_1, \ldots, \beta_q \geq 0$, а щоб вона була скінченною, сума параметрів α_j і β_j має бути меншою за 1.

Послідовність дій, що виконуються в даному блоці генерації часових рядів з умовною змінною дисперсією, така:

1. Уведення вихідних даних з блоку 1.
2. Генерація α_j та β_j параметрів моделі GARCH у модулі 1.
3. Генерація величини ξ_t з використанням ГПВЧ "Вихору Мерсенна".
4. Генерація гетероскедастичного часового ряду та його умовної змінної дисперсії на основі α_j та β_j параметрів, отриманих на кроці 2, і величини ξ_t – на кроці 3 у модулях 2 і 3.

Перейдемо до розгляду операцій, реалізованих у модулі 1, в якому виконується послідовна генерація параметрів моделі GARCH за таким алгоритмом.

1. Уведення вихідних даних: коефіцієнт завдання скінченності умовної дисперсії *sl;* кількість і сума параметрів (α_j або β_j), які позначимо: z і sum$Z$.

2. Ініціалізація змінних:

zParam = [ ] – вектор параметрів, що генерується. Якщо кількість параметрів дорівнює 0, повертається порожня множина; $k = 0$ – лічильник циклу за параметрами.

3. Якщо $z < 0$, то відбувається перехід на крок 4, інакше завершується робота модуля 1.

4. Виконуємо розрахунок параметрів моделі на основі вагового коефіцієнта $\psi_k = 2 \cdot \dfrac{z+1-k}{z \cdot (z+1)}$ для моделі ARCH, запропонованого Енглом [3], де k – номер поточного параметра. З урахуванням виразу, що задає скінченність умовної дисперсії для кожної множини параметрів моделі, розрахунок параметрів виконується так:

$$zParam(k) = \mathrm{sum}Z \cdot (1 - sl) \cdot 2 \cdot \frac{z+1-k}{z \cdot (z+1)}.$$

5. Збільшення лічильника k на одиницю.

6. Якщо $k > z$, то завершується робота модуля 1, інакше відбувається перехід на крок 4.

Розглянемо операції, реалізовані в модулі 2, в якому виконується генерація гетероскедастичного часового ряду і його умовної змінної дисперсії, за таким алгоритмом.

1. Уведення вихідних даних: w, значення параметрів α_j та β_j, розрахованих у модулі 1 і позначених далі як *aParam* і *bParam* відповідно, λ, ξ_t, T.

2. Ініціалізація змінних: $k = 1$ – лічильник циклу; *sigma* = [ ] та *et* = [ ] – вектори значень умовної змінної дисперсії та гетероскедастичного часового ряду, що генерується; *aSum* = 0 та *bSum* = 0 – змінні для зберігання суми α_j та β_j параметрів.

3. З огляду на те, що модель GARCH обчислюється за рекурентними формулами (2–3), то як початкове значення умовної дисперсії в праці [3] пропонується використовувати безумовну дисперсію процесу ε_t, яка може бути розрахована за формулою:

$$\sigma_1^2 = \frac{\omega}{1 - \sum_{j=1}^{p} \alpha_j - \sum_{j=1}^{q} \beta_j}. \tag{4}$$

Слід зазначити, що цей вираз (4) можна використати, коли виконується нерівність $\sum_{j=1}^{p} \alpha_j - \sum_{j=1}^{q} \beta_j < 1$, однак для моделей сім'ї IGARCH ця нерівність перетворена на рівність і дорівнює 1 [3]. Тому вираз (4) не може бути застосовано для визначення первісного значення умовної дисперсії для моделей IGARCH.

Крім того, вибір первісного значення некритичний при генерації умовної дисперсії, враховуючи те, що значення умовних дисперсій, початкові значення яких задано різними виразами, збігаються через $\left(\sum_{j=1}^{p} \alpha_j + \sum_{j=1}^{q} \beta_j\right) \Big/ \left(1 - \sum_{j=1}^{p} \alpha_j + \sum_{j=1}^{q} \beta_j\right)$ лагів (наприклад, значення лага збіжності 0,95/(1 – 0,95) = 19), що дорівнює середній довжині геометричного лага, наявного в моделях ARCH. Унаслідок зазначених зауважень початкове значення умовної дисперсії запропоновано розраховувати як величину $E(\xi_t)$, де $E(\xi_t)$ – безумовна дисперсія ξ_t.

Таким чином, на кроці 3 виконується обчислення $\sigma_1^2 = E(\xi_t)^2$.

4. Розрахунок $e_1 = \xi_1 \sigma_1^{1/2}$.

5. Виконання добутку матриць параметрів *aParam* і лагових значень умовної дисперсії, що розраховуються в модулі 3. Ця дія рівнозначна обчисленню суми $\sum_{j=1}^{q} \alpha_j \sigma_{t-j}^2$.

6. Виконання добутку матриць параметрів *bParam* і лагових значень гетероскедастичного ряду, що розраховуються в модулі 3. Ця дія рівнозначна обчисленню суми

7. Розрахунок k-го значення умовної дисперсії та гетероскедастичного часового ряду. Збільшення лічильника $k = k + 1$.

8. Якщо $k < T$, то перехід до кроку 5, інакше – завершення роботи модуля 2.

У модулі 3 реалізується функціональність оператора лагів (L), використання якого досить зручне в маніпулюванні змінними лагів. Основними властивостями оператора лагів є $Lc = c$, якщо c – константа, $L^p x_t = x_{t-p}$, $L^q(L^p x_t) = L^{p+q} x_t = x_{t-p-q}$ і $(L^p + L^q)x_t = x_{t-p} + x_{t-q}$ [2]. За визначенням, $L^0 x_t = I x_t = x_t$.

Використовуючи лаговий оператор, умовну змінну дисперсію в моделі GARCH (3) можна подати у вигляді:

$$\sigma_t^2 = w + \alpha(L)\sigma_t^2 + \beta(L)|\varepsilon_t|^2.$$

У модулі 3 виконується така послідовність дій.

1. Уведення вихідних даних: x – матриця, щодо якої обчислюються значення лагів; t – індекс стовпця матриці x, щодо якого розраховується оператор лагів; n – глибина занурення; *zer* – нульові значення для елементів, які виходять за межі отриманого ряду, тобто мають індекс 0, –1 і т. д.

2. Ініціалізація змінної $C = zeros(size(x, 1), n)$, що є підкладкою для оберненої матриці.

3. Якщо $t = 1$ або $t < n$, то перехід на крок 4, інакше – на крок 5.

4. Присвоєння кожному елементу матриці C значення *zer*.

5. Якщо $t < 1$, то завершення роботи лагового оператора.

6. Якщо $t \le n$, то перехід на крок 7, інакше – на крок 8.

7. Виконання $C(:, size(c, 2) - t + 1 + 1 : size(c, 2)) = x(:, 1 : t - 1)$ та завершення роботи.

8. Виконання $C = x(:, t - n : t - 1)$ та завершення роботи алгоритму.

Для перевірки адекватності роботи блоку генерації гетероскедастичних часових рядів виконаємо моделювання часового ряду відповідної моделі GARCH (2, 2).

Для генерації параметрів моделі GARCH (2, 2) використовуємо такі значення змінних: $s1 = 0{,}05$; $\lambda = 2$; $a =$

2; b = 2; sum$A$ = 0,5; sum$B$ = 0,5. Отримані значення параметрів дорівнюють $\alpha_1 = \beta_1 = 0{,}3167$ та $\alpha_2 = \beta_2 = 0{,}1583$.

Випадкову величину ξ_t одержуємо за допомогою ГПВЧ з нормальним законом розподілу, математичним сподіванням – нуль, дисперсією, що дорівнює одиниці, та довжиною вектора – 300.

Генерацію гетероскедастичного часового ряду проведемо, використовуючи такі значення змінних: w = 0,05; λ = 2; T = 300.

Результати моделювання зображено на рис. 2.

Для перевірки гіпотези, що згенерований часовий ряд гетероскедастичний, скористаємося тестом Енгла, який підтверджує висунуту гіпотезу, з рівнем значущості 0,001.

Таким чином, проведений аналіз результатів функціонування розроблених блоків імітаційної моделі підтверджує правильність роботи запропонованих алгоритмів для генерації часових рядів з умовною змінною дисперсією і параметрів відповідних їм моделей.

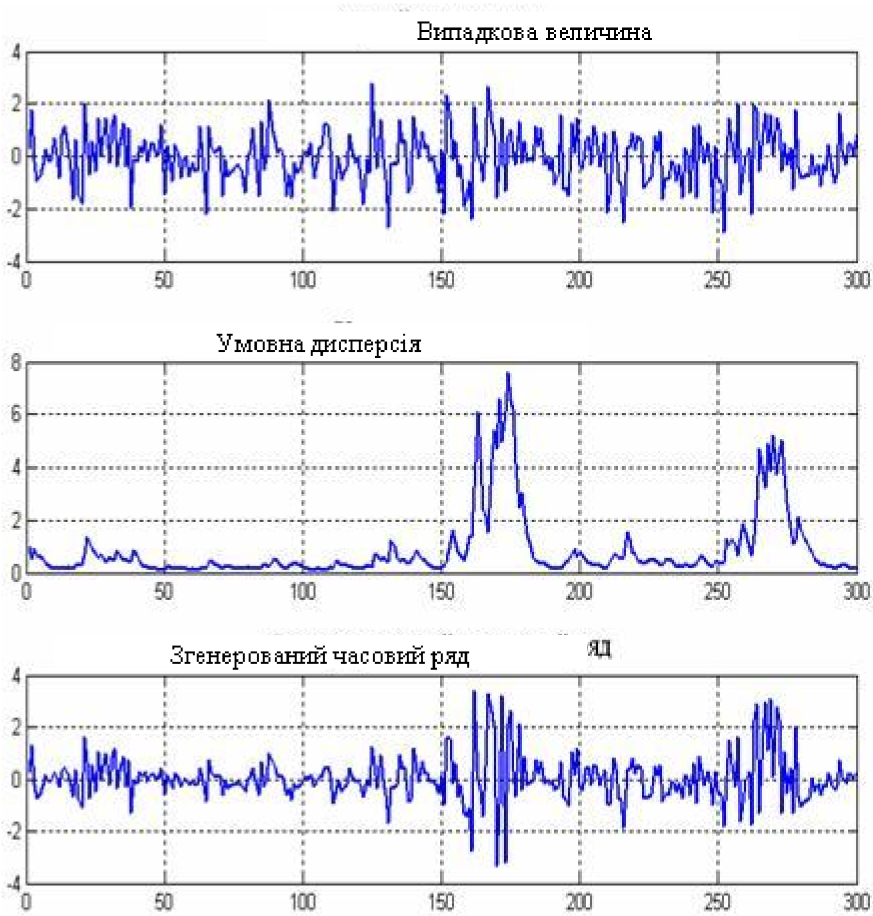

Рис. 2. Результати моделювання

Блок формування результатів імітаційного моделювання

У цьому блоці формується матриця, що містить значення випадкової величини ξ_t, умовної змінної дисперсії, часового ряду та гетероскедастичний часовий ряд.

Порівняння з аналогами

Порівняємо розроблений блок генерації гетероскедастичних часових рядів з аналогом, реалізованим у математичному пакеті MatLab (команда *garchsim*). Для цього скористаємося процедурою профілювання програмного коду, реалізованого командою *profile*, а також командами *tic* і *toc* для визначення витраченого на обчислення часу.

У ході порівняння виконувалась послідовна генерація 100 ГЧР довжиною 500 значень і визначався час, що витрачається на виконання розрахунку.

Порівняння виконано на ПК, продуктивність якого оцінено математичним пакетом MatLab (команда *bench*) як: LU = 0,3167; FFT = 0,6472; ODE = 0,4271; $Sparse$ = 0,7128; 2 – D = = 1,023; 3 – D = 0,7682.

Результати порівняння подано в табл. 2.

Таблиця 2

Результати порівняння з аналогами

Найменування	Розроблений блок	Стандартний блок MatLab

Витрачений час, с	7,0955	12,2185

Програмний код основних функціональних блоків імітаційної моделі

Нижче подано функції мовою математичного пакета MatLab, що реалізують основну функціональність розробленої імітаційної моделі.

Функція *getGarchParam* реалізує розрахунок параметрів моделі GARCH за алгоритмом модуля 1, функція *getGarchTimeSeries* – генерацію ГЧР (модуль 2) і функція *lagOper* – лаговий оператор (модуль 3) з блоку 4.

Для скорочення обсягу викладеного з програмного коду вилучено перевірку обмежень на параметри.

```
function [zParam] = getGarchParam (sl, z, sum)
zParam = [];
if z > 0, zParam(1:z) = sum*(1 – sl)*2*(z + 1 – 1 – (1:z))/(z*(z + 1)); end;
function varargout = getGarchTimeSeries(ksi, c, w, aParam, bParam, lambda)
aParam = flipdim(aParam, 2);
bParam = flipdim(bParam, 2);
sigm = zeros(1, length(ksi));
et = zeros(1, length(ksi));
sigm(1) = var(ksi).^lambda;
et(1) = c + ksi(1)*(sigm(1).^(1/lambda));
for t = 2:length(ksi)
if (size(aParam, 2) = = 0), aSum = 0;
else
aSum = aParam*lagOper(sigm, t, size(aParam, 2), sigm(1))';
end;
if (size(bParam, 2) = = 0), bSum = 0;
else
bSum = bParam*lagOper(abs(et).^lambda, t, size(bParam, 2), et(1)^lambda)';
end;
sigm(t) = w + aSum + bSum;
et(t) = c + ksi(t)*(sigm(t).^(1/lambda));
end;
varargout{1} = sigm;
varargout{2} = et;
function c = lagOper(x, t, n, zer)
c = zeros(size(x, 1), n);
if (t = = 1 || t <= n), c(:,:) = zer; end;
if t > 1
if t <= n
c(:,size(c, 2) – t + 1 + 1:size(c, 2)) = x(:,1: t – 1);
else
c = x(:, t – n:t – 1);
end;
end;
```

Висновки. Розроблено структуру, блоки, модулі та алгоритми імітаційної моделі, що дозволяють розрахувати значення параметрів ARCH – GARCH моделей з урахуванням обмежень, виконати генерацію часових рядів з умовною змінною дисперсією відповідно до заданих статистичних характеристик, що і є науковою новизною роботи.

Практичну новизну статті становить програмна реалізація імітаційної моделі мовою математичного пакета MatLab. Порівняння швидкості розрахунку блоку генерації ГЧР зі стандартними алгоритмами MatLab показало, що розроблений блок імітаційного моделювання в 1,72 раза менше витрачає часу на проведення розрахунку, ніж стандартний.

Результати роботи можуть бути використані в розробці інформаційних систем побудови моделей прогнозування, а також для розв'язання задач складання прогнозів техніко-економічних показників і прийняття рішень у різних галузях господарської діяльності й науки.

Література

1. Математичне забезпечення інформаційно-управляючих систем [Текст] : підручник для студентів вищих навчальних закладів / Б. В. Шамша, А. М. Гуржій, З. В. Дудар, В. М. Левикін. – Х. : СМІТ, 2005. – 448 с.

2. Грін Вільям Г. Економетричний аналіз [Текст] / Вільям Г. Грін / пер. з англ. А. Олійник, Р. Ткачук ; наук. ред. пер. О. Комашко ; передм. О. І. Черняка, О. В. Комашка. – К. : Основи, 2005. – 1197 с.

3. Эконометрия [Текст] / В. И. Суслов, Н. М. Ибрагимов, Л. П. Талышева, А. А. Цыплаков. – Новосибирск : СО РАН, 2005. – 744 с.

4. Одейчук А. Н. Интеллектуальная система выбора метода прогнозирования стохастических рядов в условиях гетероскедастичности [Текст] / А. Н. Одейчук, Б. В. Шамша, Е. Г. Федоров // АСУ и приборы автоматики. – 2007. – Вып. 138. – С. 9–14.

5. Економіко-статистичне моделювання і прогнозування [Текст] : навчальний посібник / В. П. Кічор, Р. В. Фещур, В. В. Козик та ін. – Львів : Видавництво Національного університету "Львівська політехніка", 2007. – 156 с.

6. Формирование массива элементов, распределенных по равномерному закону [Электронный ресурс]. –

Режим доступа : http:// www.exponenta.ru/soft/matlab/potemkin/ book2/chapter5/rand.asp.
7. Park S. K. Random Number Generators: Good ones are hard to find [Текст] / S. K. Park, K. W. Miller // Comm. ACM. – 1998. – № 10. – Vol. 32. – P. 1192–1201.
8. SIMD-oriented Fast Mersenne Twister (SFMT): twice faster than Mersenne Twister [Електронний ресурс]. – Режим доступу : http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/ SFMT/index.html
9. Matsumoto M. Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudorandom Number Generator [Текст] / M. Matsumoto, T. Nishimura // ACM Transactions on Modeling and Computer Simulation. – 1998. – № 1. – Vol. 8. – P. 3–30.
10. Tor Roneid. Collusion-Secure Fingerprinting. A simulation of the Boneh and Shaw Scheme. Coding theory and cryptography [Електронний ресурс] / Roneid Tor. – Bergen : Department of Informatics Universitas Bergensis, 2005. – 119 p. – Режим доступу : http://www.ii.uib.no/ ~georg/coding/Students/Tor/ collusion_secure_fp.pdf
11. Mathematics, MATLAB Version 7.1 (R14SP3) Release Notes [Електронний ресурс]. – Режим доступу : http://www.mathworks.com/access/ helpdesk/ help/techdoc/index.html
12. Серая О. В. Имитационное моделирование [Текст] : учебное пособие / О. В. Серая. – Х. : НТУ "ХПИ", 2003. – 80 с.
13. Normal Behavior. Ziggurat algorithm generates normally distributed random numbers [Електронний ресурс]. – Режим доступу : http://www.mathworks.com/company/newsletters/ news_ notes/clevescorner/spring01_cleve.html.