УДК 004.56 DOI https://doi.org/10.35546/kntu2078-4481.2025.1.2.25

С. А. ПОЛОЖАЄНКО
доктор технічних наук, професор,
завідувач кафедри комп'ютеризованих систем та програмних технологій
Національний університет «Одеська політехніка»
ORCID: 0000-0002-4082-8270

А. Ю. ПРОКОФЬЄВ
аспірант кафедри комп'ютеризованих систем та програмних технологій
Національний університет «Одеська політехніка»
ORCID: 0000-0002-4520-4248

МЕТОДИ КОНТРОЛЮ ДОСТОВІРНОСТІ ОБЧИСЛЮВАЛЬНОЇ РЕАЛІЗАЦІЇ МОДЕЛЕЙ ДИНАМІЧНИХ СИСТЕМ, ЗАСНОВАНІ НА ЗАСТОСУВАННІ КОНТОЛЬНИХ АЛГОРИТМІВ

В роботі запропоновано методи контролю достовірності обчислювальної реалізації математичних моделей (ММ) динамічних систем в процесах їх моделювання та управління. Відмінною особливістю цих методів є побудова останніх із застосуванням процедур контрольних обчислень, причому алгоритм контрольних обчислень складає невід'ємну частину загального алгоритму певного методу. Перевірка достовірності виникає через необхідність усунення збоїв, які, з великою ймовірністю, з'являються при реалізації ММ динамічних систем в прикладних задачах моделювання та управління. Причому, актуальність перевірки достовірності розв'язування виразів, які складають ММ динамічних систем (суть – реалізації ММ), обумовлено включенням відповідних обчислювальних засобів безпосередньо в контур управління, а тому адекватність представлення ММ безпосередньо впливає, в кінцевому підсумку, на якість процесу управління. Організація контролю передбачає наявність еталонних величин, з якими порівнюються результати обчислень. В методах, що розглянуто в роботі, запропоновано, в якості еталонних величин, використовувати проконтрольовані результати попередніх обчислень або результати, при отриманні яких ймовірність збою мала. Виходячи з цього, очевидною є можливість організувати реалізацію ММ динамічних систем в процесах їх моделювання та управління зі зростаючою точністю обчислень, яка полягає у послідовному отриманні значення функції (або, іншими словами, вихідного сигналу динамічної системи, представленою відповідною ММ) на кроці обчислень з порядком точності локальної похибки, що збільшується. Крім метода зі зростаючою точністю, в роботі запропоновано екстраполяційний та інтерполяційний методи контролю, принципова відмінність яких полягає у тому, якого роду інформація використовується для контролю. В екстраполяційному методі для підвищення достовірності контролю на кроці обчислювального процесу, який контролюється, використовується інформація, вже проконтрольована на попередніх кроках обчислювального процесу. Інтерполяційний метод контролю, на відміну від екстраполяційного, враховує зміну розв'язку на кроці обчислень, який контролюється, що дозволяє здійснювати непрямий контроль обраної еталонної функції стану. Розв'язування тестових задач показало конструктивність запропонованих методів контролю достовірності обчислювальної реалізації ММ у випадках їх застосування до реальних динамічних систем.

Ключові слова: *математична модель, контрольні обчислення, достовірність обчислень, еталонне значення, екстраполяція, інтерполяція.*

S. A. POLOZHAENKO
Doctor of Technical Sciences, Professor,
Head of the Department of Computerized Systems and Software Technologies
National University "Odesa Polytechnic"
ORCID: 0000-0002-4082-8270

A. YU. PROKOFIEV
Postgraduate Student at the Department of Computerized Systems
and Software Technologies
National University "Odesa Polytechnic"
ORCID: 0000-0002-4520-4248

METHODS FOR CONTROLLING THE RELIABILITY OF COMPUTATIONAL IMPLEMENTATION OF DYNAMIC SYSTEMS MODELS BASED ON THE APPLICATION OF CONTROL ALGORITHMS

The paper proposes methods for checking the reliability of computational implementation of mathematical models (MM) of dynamic systems in the processes of their modeling and control. A distinctive feature of these methods is the construction of the latter using control calculation procedures, and the algorithm of control calculations is an integral part of the general algorithm of a particular method. Reliability checking arises due to the need to eliminate failures that, with a high probability, appear when implementing MM of dynamic systems in applied modeling and control problems. Moreover, the relevance of checking the reliability of solving expressions that make up MM of dynamic systems (the essence is the implementation of MM) is due to the inclusion of appropriate computational tools directly in the control loop, and therefore the adequacy of the MM representation directly affects, ultimately, the quality of the control process. The organization of control assumes the presence of reference values with which the results of calculations are compared. In the methods considered in the work, it is proposed to use, as reference values, the controlled results of previous calculations or results, when obtaining which the probability of failure is small. Based on this, it is obvious to organize the implementation of MM of dynamic systems in the processes of their modeling and control with increasing accuracy of calculations, which consists in sequentially obtaining the value of the function (or, in other words, the output signal of the dynamic system, represented by the corresponding MM) at the calculation step with an increasing order of accuracy of the local error. In addition to the method with increasing accuracy, the work proposes extrapolation and interpolation control methods, the fundamental difference of which is the type of information used for control. In the extrapolation method, to increase the reliability of control at the step of the computational process that is being controlled, information that has already been controlled at the previous steps of the computational process is used. The interpolation control method, unlike the extrapolation method, takes into account the change in the solution at the step of the calculations that is being controlled, which allows for indirect control of the selected reference state function. Solving test problems showed the constructiveness of the proposed methods for controlling the reliability of the computational implementation of the MM in cases of their application to real dynamical systems.

Key words: *mathematical model, control calculations, reliability of calculations, reference value, extrapolation, interpolation.*

Постановка проблеми

Відмінною ознакою сучасних систем управління є переважне застосування різного роду обчислювальних засобів (алгоритмічних та програмних), які використовуються – в якості модельної підтримки – для реалізації ММ динамічних об'єктів, а також алгоритмів управління останніми – при здійсненні, наприклад, технологічних процесів. Тому якість функціонування систем моделювання та управління, у значній мірі, визначається характеристиками обчислювальних засобів, що використовуються. Проблема забезпечення якості функціонування, зокрема управляючих систем з обчислювальною машиною в контурі управління, є дуже складною як з технічної, так і математичної точок зору. При вирішенні цієї проблеми практика висунула на одне з перших місць науковий напрямок, пов'язаний із розробкою методів та засобів, що забезпечують надійність процесів обчислень та обробки інформації [1–4].

У реальних системах моделювання чи управління процеси обчислень неминуче супроводжуються різного роду *завадами* та *похибками*. Так, у разі числового аналізу рівнянь (або інших математичних залежностей) динаміки (що утворюють ММ динамічної системи) на цифровій машині, похибки розв'язування цих рівнянь, крім усього, можуть бути спричинені *збоями* та відмовами в роботі апаратури. При цьому під *збоями* розуміються [5] спотворення оброблюваної та управляючої інформації під впливом первинних причин, що самоусуваються (неправильне спрацьовування двійкових елементів апаратури, завади, коливання напруги джерел живлення тощо). Причому більшість збоїв мають *випадковий характер*. Тому потрібен дієвий *поточний контроль*, що дозволяє виявити та усунути похибку до її поширення в контурі управління, що зумовлює *оперативність* як одну з основних характеристик засобів контролю похибок.

На тепер використовуються як *програмні* і *апаратурні* методи контролю процесів моделювання (чи управління), так і різноманітні їх поєднання [6, 7]. Програмні методи контролю, своєю чергою, можна поділити на *тестові* та *програмно-логічні* [8–10]. Тестовий контроль [11–15] призначено для перевірки відмов обладнання або програмного забезпечення у момент, коли не розв'язується робоча задача на обчислювально-управляючому пристрої або допускається подача тестових сигналів у ході розв'язування задачі.

В цьому випадку необхідно контролювати обчислювальний процес у робочому режимі функціонування системи моделювання або управління, а основним об'єктом контролю є збої обладнання. Більш раціональним вважаються програмно-логічні методи контролю, що дозволяють здійснювати контроль у процесі розв'язування задачі. Досить повний огляд існуючих методів програмно-логічного контролю (ПЛК) наведено у низці робіт [16–18]. Головним обмеженням застосування існуючих методів контролю числового розв'язування, зокрема, складних нелінійних диференціальних рівнянь (як основного математичного апарату, що описує динамічну поведінку

систем) у системах моделювання та управління є час, що відводиться на проведення контролю. Нижче, на прикладі диференціальних рівнянь, запропоновано методи контролю похибок, викликаних збоями при числовому розв'язуванні цих рівнянь (суть – ММ динамічних об'єктів) в часі, тобто, у темпі з реалізацією управляючого впливу в контурі управління.

Формулювання мети дослідження

Метою роботи є розробка методів, які забезпечують необхідні точність та достовірність розв'язків відповідних рівнянь як математичних моделей динамічних систем, а також виявлення збоїв (завад) в обчислювальному процесі.

Викладення основного матеріалу дослідження

Попередньо визначимо можливості контролю процедур та алгоритмів числового розв'язування математичних виразів, які складають ММ динамічних систем.

В основу числових методів реалізації ММ динамічних систем (в переважній більшості які представлено диференціальними рівняннями у повних або частинних похідних) покладено процедури *покрокового* отримання шуканої функції. При цьому обчислювальний процес природно розбивається на відносно постійні за часом частини, які відповідають визначенню шуканої функції на певному кроці інтегрування. Крім того, обчислювальний процес також розбивається всередині кроку інтегрування на постійні за часом частини, які обумовлено значеннями правої частини системи рівнянь ММ

$$\dot{Y} = f(Y,U,t), \quad Y(t_0) = Y_0, \tag{1}$$

де $Y = (y_1, y_2, \ldots, y_n)^T$ – вектор змінних стану, $U = (u_1, u_2, \ldots, u_m)^T$ – вектор сигналів управління, $f = (f_1, f_2, \ldots, f_n)^T$ – вектор-функція, t – незалежний параметр часу.

При будь-якому з методів числового розв'язування рівнянь динаміки виду (1) контроль збоїв можна організувати шляхом послідовного порівняння результатів, отриманих на поточному та наступному кроках інтегрування.

Наявність чи відсутність збоїв визначається перевіркою умови

$$\xi = \rho(y_i, y_{i+1}) \in D, \tag{2}$$

де ξ – міра близькості значень y_i та y_{i+1}, D – область допустимих значень цієї міри.

Для контролю похибок, викликаних збоями та відмовами, можуть використовуватися методи контролю методичної похибки числового розв'язку на етапі інтегрування. Так, у методах прогнозу та корекції [19] по різниці $\xi = |y_f - y_c|$ між прогнозованим y_f і скорегованим y_c значеннями шуканої функції можна робити висновки про наявності збою у роботі апаратури. Однак, існуючий взаємозв'язок між значеннями y_f та y_c знімає достовірність контролю.

Метод зі зростаючою точністю обчислень. На відміну від методів прогнозу та корекції, що, як зазначено вище, не забезпечують гарантованих оцінок достовірності обчислень, запропонуємо метод *зі зростаючою точністю обчислень*, сутність якого буде полягати у послідовному отриманні значення функції на кроці інтегрування з порядком точності локальної похибки, що збільшується. Очевидно, що для довільного порядку точності r, можна записати:

$$y_{i+1}^{(r)} = y_i^{(r)} + \frac{h}{2}\left[f(y_i^{(r)}, u_i, t_i) + f(y_{i+1}, u_{i+1}, t_{i+1}) \right], \tag{3}$$

$$y_{i+1}^{(r-1)} = y_i^{(r)} + hf(y_i^{(r)}, u_i, t_i), \tag{4}$$

де верхній індекс вказує на порядок локальної похибки наближеного значення. Порівнюючи значення $y_{i+1}^{(r-1)}$ та $y_{i+1}^{(r)}$ між собою можна зробити висновок щодо локальної похибки розв'язку. Цей самий прийом можна застосувати і для виявлення похибок від збоїв. Таким чином, якщо на попередньому кроці не було збою, то похибка при визначенні $y_{i+1}^{(r-1)}$ може відбутися тільки за умови реалізації виразу (4).

Інакше кажучи, на кожному кроці інтегрування маємо еталонне значення $y_{i+1}^{(r-1)}$, достовірність якого визначається ймовірністю збою при обчисленнях за формулою (4). І в цьому випадку величина $\xi = \left| y_{i+1}^{(r-1)} - y_{i+1}^{(r)} \right|$ визначається різницею залишкових членів і є малою величиною порядку $O(h^2)$. Якщо при визначенні $y_{i+1}^{(r-1)}$ відбувся збій та отримано значення з похибкою, то при визначенні $y_{i+1}^{(r)}$ буде використано значення похідної $\dfrac{\partial f(y + \Delta y, u, t_{i+1})}{\partial y}$, що може, в загальному випадку, знизити достовірність контролю, заснованого на порівнянні $y_{i+1}^{(r)}$ з $y_{i+1}^{(r-1)}$.

Алгоритм, що реалізує метод *зі зростаючою точністю обчислень* повинен відповідати наступним вимогам:

1. Процес обчислень на кроці інтегрування має починатися з визначення функції f в точці t_{i+1}. При цьому не обчислюється функція $f(y, u, t_{i+1})$, а використовується значення функції f, отримане та проконтрольоване при обчисленні найбільш точного значення розв'язку системи (1) на попередньому кроці інтегрування. Розрахункова формула для отримання першого наближеного значення $y(t_i + h)$ з локальною похибкою $O(h^2)$ має вигляд

$$y_{i+1}^{(2)} = y_i^{(r)} + hf(y_i^{(r-1)}, u_i, t_i),$$

де r – порядок локальної похибки наближеного значення розв'язку, $f(y_i^{(r-1)}, u_i, t_i)$ – оцінка похідної, яку використано при визначенні $y_i^{(r)}$.

2. Число розбиттів k (тобто число значень $y_{i+1}^{(r)}$, які отримано зі зростаючою точністю на відрізку $[t_i, t_{i+1}]$) можна обрати, виходячи з обмежень на величину $\xi = (y_{i+1}^{(l+1)} - y_{i+1}^{(l)})$.. Одначе, в практичних застосунках, доцільно, при синтезі алгоритму методу зі зростаючою точністю, обирати максимальне число розбиттів k, яке можливе для методу обраного порядку точності r.

Оцінимо головний член похибки, застосовуючи для цього правило Рунге [20]. Для деякої точки t відрізка $[t_0, t_N]$ методом k-го порядку точності послідовно можна отримати наближені значення функції y_{h_1} та y_{h_2} на кроках $h_1 \neq h_2$, відповідно.

Визначимо величину:

$$\xi(t) = \frac{y_{h_1} - y_{h_2}}{h_2^k - h_1^k}.$$

Тоді, з урахуванням величини головного члена похибки для кожного із значень $h = h_1$ і $h = h_2$

$$\xi_1 = h_1^k \frac{y_{h_1} - y_{h_2}}{h_2^k - h_1^k} \quad \text{та} \quad \xi_2 = h_2^k \frac{y_{h_1} - y_{h_2}}{h_2^k - h_1^k},$$

можна зробити висновок щодо точності обчислень.

При рівномірному кроці розбиття відрізка $[t_0, t_N]$, тобто, якщо $h_1 = h$, а $h_2 = 2h$, вираз для головного члена похибки набуде наступного вигляду:

$$\xi = y - y_h \approx \frac{y_h - y_{2h}}{2^k - 1}.$$

В залежності від обчисленої величини ξ, для отримання заданої точності, крок h слід або зменшувати, або збільшувати. Похибка, викликана збоєм та така, що перевищує допустиме значення оцінки головного члена похибки $\xi_{доп}$, буде виявлена та усунута шляхом перерахунку із зменшенням кроку h.

Основною перевагою запропонованого методу є простота алгоритму та незначний обсяг обчислень. Одначе достовірність точного контролю може виявитися незадовільною через *взаємовплив величин*, на підставі порівняння яких робиться висновок щодо наявності похибок в процесі обчислень.

Екстраполяційний метод контролю. В низці прикладних задач, зокрема при оцінці достовірності отриманого розв'язку, домінуюче значення мають *час прогнозу* (тобто швидкодія у його отриманні) та *простота* задіяних *обчислювальних процедур*. Зокрема, такі вимоги можуть виникати при розв'язуванні задач управління швидкоплинними динамічними процесами (об'єктами), особливо в умовах багатоваріантної постановки, наприклад, оптимізаційних задач.

Запропонуємо спрощений метод контролю, проблему побудови якого сформулюємо наступним чином: для певного класу задач Q отримати процедуру та алгоритм розв'язування A_k, результат виконання яких Y, у застосуванні до розв'язування задачі $q \in Q$ співпадає з точністю до ε з результатом, отриманим по основному алгоритму A_b. При цьому витрати часу та обчислювальні витрати (зокрема, машинної пам'яті) на реалізацію алгоритму A_k мають перебувати у заданих межах.

В разі обчислення по основному алгоритму A_b для класу задач Q моделювання динаміки динамічних об'єктів (або управління ними) більша частина часу витрачається на обчислення правої частини системи (1). Час обчислення за алгоритмом A_k можна значно скоротити у порівнянні з алгоритмом A_b, якщо в ньому використовувати значення функції $f(Y, U, t)$, отримані по алгоритму A_b. Оскільки обчислювальний процес за алгоритмом A_b також наражається на збої, то відмова від обчислювання функції $f(Y, U, t)$ значно зменшить ймовірність похибок першого роду – α-похибок. В даному випадку можна досягнути підвищення достовірності контролю на кроці обчислювального процесу, який контролюється, якщо у контрольній процедурі (контрольному алгоритмі) використовувати інформацію, яку вже проконтрольовано на попередніх кроках, тобто використати для контролю ідею *екстраполяції*.

Тоді, контрольне значення функції на $(i + 1)$-му кроці інтегрування, який контролюється, може визначатися за формулою

$$\overline{y}_{i+1} = \sum_{p=0}^{p_1} a_p y_{p-1} + h \sum_{s=1}^{s_1} b_s f\left(Y, U, t_{i-s}\right), \tag{5}$$

де $h = t_{i+1} - t_i$ – крок інтегрування, a_P, b_s – коефіцієнти.

У (5) використовуються тільки значення функції та її похідних, які проконтрольовано на попередніх кроках інтегрування. Верхні межі сум p_1 та s_1 обираються з умови потрібного порядку точності алгоритму, який

контролюється. Коефіцієнти a_P, b_s доцільно обирати, виходячи з умови мінімальної локальної похибки (тобто рівності нулю залишкового члена $r_i = 0$ між точним розв'язком та числовим розв'язком (5)).

Якщо при екстраполяції враховувати одну оцінку вказаних складових розв'язку (наприклад, похідної для диференціальних рівнянь динаміки), отриману за основним алгоритмом, то у виразі (5) додасться один доданок:

$$\overline{y}_{i+1} = \sum_{p=0}^{p_1} a_p y_{p-1} + h\left[\sum_{s=1}^{s_1} b_s f(Y,U,t_{i-s}) + b_0 f(Y,U,t_{i+l})\right], \tag{6}$$

де $f(Y, U, t)$ – оцінка складової (наприклад, похідної), b_0 – коефіцієнт.

Розглянемо отримання коефіцієнтів a_P та b_s для контрольної формули (6). Оскільки контрольна формула (6) являє собою по суті багатокроковий екстраполяційний метод розв'язування крайової задачі, то точний розв'язок $y = y(t)$ за умови, що на попередніх кроках обчислень отримано також «точний» розв'язок, буде визначатися наступним чином:

$$y(t_{i+1}) = \sum_{p=0}^{p_1} a_p y(t_{i-p}) + h\sum_{s=1}^{s_1} b_s f\left[Y(t_{i-s}), u_{i-s}, t_{i-s}\right] + hb_0 f\left[Y(t_{i-l}), u_{i-l}, t_{i-l}\right] + r_i, \tag{7}$$

де r_i – залишковий член.

Вибором коефіцієнтів a_P, $p = \overline{0, p_1}$ та b_0; b_s, $s = \overline{1, s_1}$ слід досягнути малості величини r_i, причому, для простоти вибору, можна припустити, що $r_i \to 0$ (або, навіть $r_i \equiv 0$, якщо розв'язок $y = y(t)$ може являти собою алгебраїчний багаточлен високого ступеня). Тоді, задавшись початковими значеннями a_0, b_0, b_1, визначається перше наближення для формули (7) та перевіряється умова

$$y(t) = y(t_{i+1}) + r_i\big|_{r_i \to 0} \tag{8}$$

і, якщо (8) не виконується, тоді послідовним перебором значень коефіцієнтів a_P, $p = \overline{1, p_1}$ та b_s, $s = \overline{2, s_1}$ досягається зазначена вище умова. В результаті обираються ті значення a_P та b_s, які забезпечують найменшу величину r_i.

В якості прикладу наведемо контрольну формулу, орієнтовану на контроль результатів, отриманих за алгоритмом (6), якщо дискретизація вузлових точок виконується за виразом $t_{i+1} = t_i + \lambda h$, де $\lambda = 2/3$:

$$\overline{y}_{i+1} = y_i + hy_{i-1+\lambda}. \tag{9}$$

Для матричного рівняння (1) n-го порядку формула (6) набуває вигляду:

$$\overline{y}_{i+1} = \sum_{p=0}^{p_1} a_{j,p} y_{j,p-1} + h\left[\sum_{s=1}^{s_1} b_{j,s} f_j(Y,U,t_{i-s}) + b_{j,0} f_j(Y,U,t_{i+l})\right], \quad j = \overline{1,n}.$$

Для прикладної задачі дослідження динаміки польоту літака на рис.1 та рис. 2 показано залежність математичного очікування та середнього квадратичного відхилення величини

$$\xi = \sum_{j=1}^{n} \frac{\left|y_{j,i} - \overline{y}_{j,i}\right|}{\max\limits_t \left|y_j\right|}, \quad j = \overline{1,6},$$

від кроку h при числовому розв'язуванні рівняння (1).

Криві (позначені цифрами 1 та 2) на рис. 1 і рис. 2 отримано при використанні для контролю виразу (9) з різними значеннями параметру λ.

Інтерполяційний метод контролю. При застосуванні для визначення достовірності екстраполяційного методу контролю є можливість виконувати корекцію розв'язку при похибках не повторними обчисленнями, а заміною y_{i+1} на $\overline{y}_{i+1}$, що є суттєвим при моделюванні та управлінні в реальному (або прискореному) масштабі часу. Одначе точність контролю в даному випадку залежить від точності екстраполяції розв'язку за допомогою формули (6) і може виявитися незадовільною.

З цієї причини, з точки зору підвищення точності контролю, більшу перевагу, у порівнянні з екстраполяційним методом, слід вважати метод, заснований на ідеї *інтерполяції*. Зазначимо, що в інтерполяційних формулах контролю, на відміну від екстраполяційних, враховується *зміна розв'язку на кроці обчислень, який контролюється*, що веде до *зменшення різниці* між величинами y та $\overline{y}$, які є результатом застосування алгоритмів A_0 та A_k, відповідно. Підвищення точності інтерполяції можна отримати, залучаючи результати обчислень за основним алгоритмом A_b, отримані до кроку обчислень, який контролюється.

Зважуючи на вище наведене, запропонуємо контрольну багатокрокову інтерполяційну формулу, за допомогою якої можна здійснювати контроль процесу обчислень на $(i + 1)$-му кроці:

$$\overline{y}_i = \sum_{p=-1}^{p_1} a_p y_{i-p} + h\sum_{s=-1}^{s_1} b_s f\left(Y,U,t_{i-s}\right), \tag{10}$$

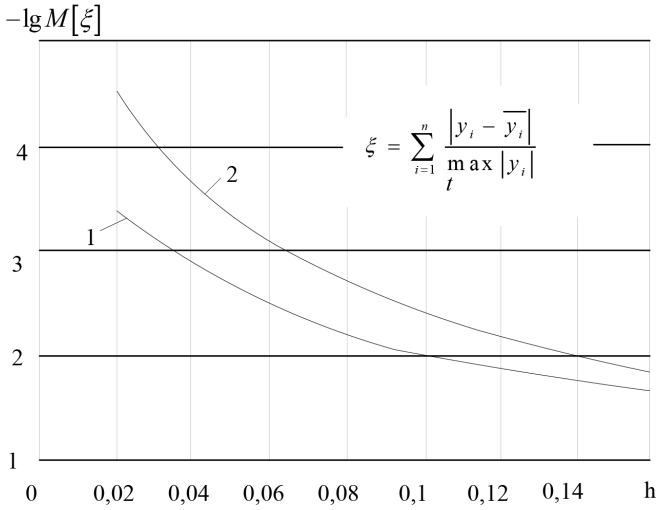

Рис. 1. Залежність математичного очікування *M*[ξ] величини ξ від кроку *h* (екстраполяційний метод)

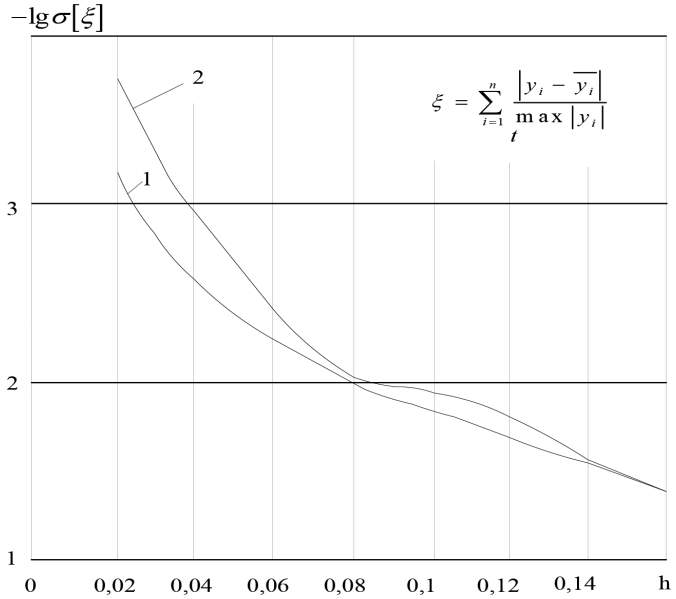

Рис. 2. Залежність середнього квадратичного відхилення σ[ξ] величини ξ від кроку *h* (екстраполяційний метод)

де a_P та b_s – дійсні коефіцієнти; $p \neq 0$ (це вироджений випадок, при якому $a_0 = 1$, а всі інші коефіцієнти дорівнюють нулю, причому виконується тотожність $\overline{y}_i \equiv y_i$).

Як і у випадку екстраполяційної формули (6) в (10) верхні границі підсумовування необхідно обирати у залежності від необхідного порядку точності контролю, а коефіцієнти a_P та b_s – з умови точності представлення поліномів відповідного ступеня. Таким чином, вираз (10) визначає процедуру *інтерполяційного методу* контролю обчислень.

Принципова відмінність інтерполяційного виразу (10) від екстраполяційного виразу (6), як контрольних алгоритмів, полягає в наступному. У разі екстраполяційних формул безпосередньо контрольованою величиною є значення функції, отримане за основним алгоритмом A_b на контрольованому кроці обчислень, а контрольне значення є результатом застосування виразу (6). У разі інтерполяційних формул контролю величина $\overline{y}_i$ обчислюється за виразом (10), а еталонним є значення функції, отримується за основним алгоритмом A_b на попередньому $(i - p)$-ому кроці обчислення, тобто здійснюється, по суті, *непрямий* контроль.

На рис. 3 та рис. 4 показано залежності математичного очікування та середнього квадратичного відхилення величини

$$\xi_i = \left[\sum_{j=1}^{p} \left(\frac{y_{i,j} - \overline{y}_{i,j}}{\max_i |y_j|} \right)^2 \right]^{1/2}$$

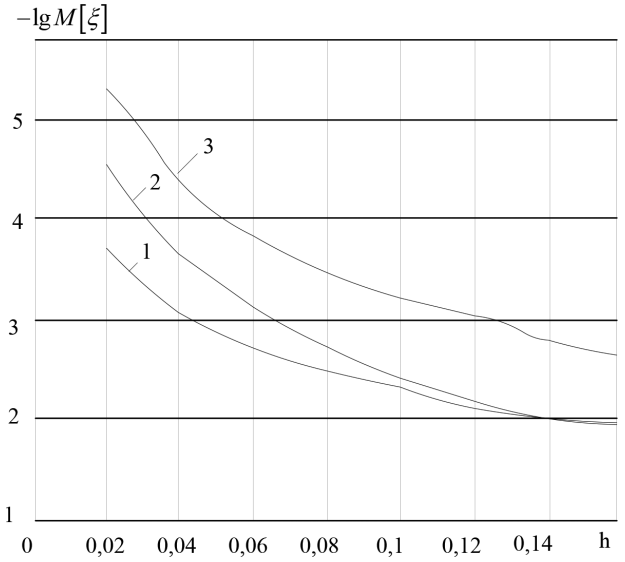

Рис. 3. Залежність математичного очікування *M*[ξ] величини ξ від кроку *h* (інтерполяційний метод)

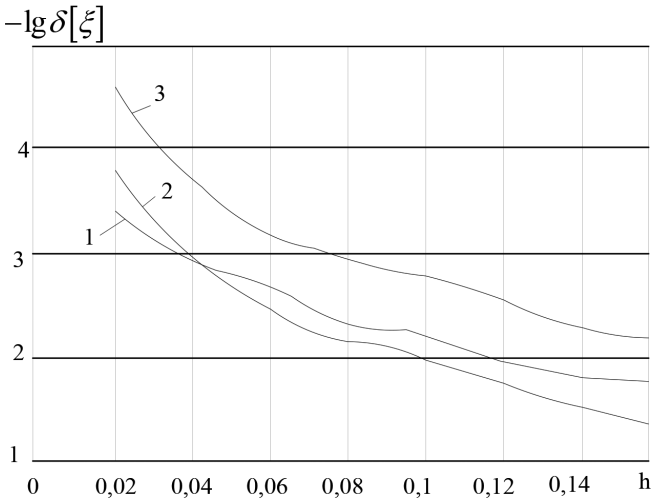

Рис. 4. Залежність середнього квадратичного відхилення σ[ξ] величини ξ від кроку *h* (інтерполяційний метод)

від кроку дискретності *h* у рівнянні (1), що як і у випадку екстраполяційного методу, описує динаміку польоту літака.

Для порівняння наведено криву 3, яка відповідає застосуванню для контролю екстраполяційної формули

$$\overline{y}_{i+1} = y_i + h(2,25\dot{y}_{i-1} - 1,25\dot{y}_{i-2}).$$

Можна бачити, що зі збільшенням кроку дискретності *h* точність інтерполяційного методу контролю, порівняно з екстраполяційним методом, збільшується.

Висновки

Проведено аналіз та визначено можливості контролю достовірності обчислювальної реалізації ММ динамічних систем в процесах їх моделювання та управління. Запропоновано метод зі зростаючою точністю обчислень, який, у порівнянні з методами «прогноз-корекція», дозволяє отримувати значення шуканої функції стану динамічних систем з порядком точності локальної похибки, що збільшується, і який (порядок) може бути завдано попередньо, до інтегрування рівнянь ММ динамічної системи.

Для певних класів прикладних задач, що потребують підвищеного контролю достовірності реалізації ММ динамічних систем, запропоновано екстраполяційний та інтерполяційний методи контролю. Методи ґрунтуються на застосуванні контрольного алгоритму, який є вбудованим в основний алгоритм на етапі числового розв'язування поставленої задачі (моделювання динаміки чи управління), але такого, що простіший за основний. При цьому екстраполяційний

метод дозволяє досягнути підвищення достовірності обчислень за рахунок застосування в контрольному алгоритмі інформації, яку вже проконтрольовано на попередніх кроках основного алгоритму, а за допомогою інтерполяційного методу можна організувати неявний контроль безпосередньо на кроці обчислень, який виконується.

При цьому екстраполяційний метод дозволяє досягнути підвищення достовірності обчислень за рахунок застосування в контрольному алгоритмі інформації, яку вже проконтрольовано на попередніх кроках основного алгоритму, а за допомогою інтерполяційного методу можна організувати неявний контроль безпосередньо на кроці обчислень, який виконується.

Розв'язування тестових задач, на прикладах моделювання динаміки літальних апаратів, показало конструктивність запропонованих методів контролю достовірності обчислювальної реалізації ММ.

Список використаної літератури

1. Ляшенко Б. М., Кривонос О. М., Вакалюк Т. А. Методи обчислень. Житомир: Вид-во ЖДУ, 2014. 228 с.

2. Шевченко С. В., Гужва В. О., Малиш В. Д., Морква І. Ю. Обґрунтування попереднього вибору архітектури системи обробки даних з використанням нечіткої логіки. *Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології*, 2019. № 2. С. 81–87. URL: https://doi.org/10.20998/2079-0023.2019.02.14

3. Парійскій Є. Ю. Порівняльний аналіз математичних моделей і підходів до моделювання та аналізу безперервно-дискретних систем. *Диференціальні рівняння та процеси керування*, 1997. № 1. С. 36–52.

4. Квєтний Р. Н., Богач І. В., Бойко О. Р. Комп'ютерне моделювання систем та процесів. Методи обчислень. Вінниця: ВНТУ, 2012. 193 с.

5. Костюшко І. А., Любашенко Н. Д., Третиник В. В. Методи обчислень. К. : НТТУ «КПІ ім. І. Сікорського», Вид-во «Політехніка», 2021. 243 с.

6. Гліненко, Л. К., Гліненко А. В., Сухоносов О. Г. Основи моделювання технічних систем. – Львів : Бескид Біт, 2003. 176 с.

7. Чабанюк Я. М., Яковина В. С., Федасюк Д. В. Побудова і дослідження моделі надійності програмного забезпечення з індексом величини проекту. *Інженерія програмного забезпечення*, 2010. Вип. 1. С. 1–6.

8. Яковина В., Смірнов В. Огляд основних підходів до аналізу надійності програмного забезпечення. *Комп'ютерні науки та інформаційні технології*, 2011 Вип. 719. С. 278–282.

9. Foidl H., Felderer M. Integrating software quality models into risk-based testing. *Software Quality Journal*, 2018. Vol. 26. P. 809–847.

10. Crispin L., Gregory J. Agile Testing: A Practical Guide for Testers and Agile Teams. New York : Addison-Wesley Signature Series (Cohn), 2010. 464 p.

11. Kaner K., Falk J., Nguyen H. Q. Testing Computer Software, Boston : International Thomson Publishing Press, 1999. 542 p.

12. Culbertson R., Brown C., Cobb G. Rapid Testing. Atlanta: Software Quality Institute Series, 2000. 374 p.

13. Катаєва Є. Ю., Одокієнко С. М., Люта М. В., Савченко Я. С. Практичний аналіз якості програмного забезпечення з відкритим кодом. *Управління розвитком складних систем*, 2020. № 44. С. 49–55. URL: https://doi.org/10.32347/2412-9933.2020.44.49-55

14. Говорущенко Т. О. Методологія оцінювання достатності інформації для визначення якості програмного забезпечення. Хмельницький: ХНУ, 2017. 310 с.

15. Галкін П. В., Ключник І. І. Програмування ПЛК в CODESYS. Харків : ФОП Панов А. М., 2019. 192 с.

16. Береза А. М. Основи створення інформаційних систем: 2 вид., перероб. і доп. К. : КНЕУ, 2001. 214 с.

17. Пупена О. М., Ельперін І. В., Луцька Н. М., Ладанюк А. П. Промислові мережі та інтеграційні технології в автоматизованих системах. К. : Ліра-К, 2011. 500с.

18. Hoffmann G. M., Rajnarayan D. G., Waslander S. L., Jang J. S. (2004) The Stanford Testbed of Autonomous Rotorcraft for Multi Agent Control (STARMAC-2004). *In the Proceedings of the 23rd Digital Avionics System Conference. Salt Lake City*, UT. P. 12–16.

19. Коссак О., Тумашова О. Методи наближених обчислень. Львів : Бак, 2003. 168 с.

20. Положаєнко С. А., Прокофьєв А. Ю. Параметри методу Рунге-Кутти з різним порядком точності при інтегруванні рівнянь динаміки в задачах моделювання нестаціонарних систем. *Інформатика та математичні методи в моделюванні*, 2024. Т. 14. № 1–2. С. 85–96. URL: https://doi.org/10.15276/imms.v14.no1-2.85

References

1. Liashenko B. M., Kryvonos O. M., Vakaliuk T. A. (2014) Metody obchyslen [Calculation methods]. Zhytomyr : Vyd-vo ZhDU. [in Ukrainian].

2. Shevchenko S. V., Huzhva V. O., Malysh V. D., Morkva I. Yu. (2019) Obgruntuvannia poperednoho vyboru arkhitektury systemy obrobky danykh z vykorystanniam nechitkoi lohiky [Justification of the preliminary choice of data

processing system architecture using fuzzy logic]. *Visnyk Natsionalnoho tekhnichnoho universytetu «KhPI». Seriia: Systemnyi analiz, upravlinnia ta informatsiini tekhnolohii,* no. 2, pp. 81–87. [in Ukrainian]. URL: https://doi.org/10.20998/2079-0023.2019.02.14

3. Pariiskii Ye. Yu. (1997) Porivnialnyi analiz matematychnykh modelei i pidkhodiv do modeliuvannia ta analizu bezpererervno-dyskretnykh system [Comparative analysis of mathematical models and approaches to modeling and analysis of continuous-discrete systems]. *Dyferentsialni rivniannia ta protsesy keruvannia*, no 1, pp. 36–52. [in Ukrainian].

4. Kvietnyi R. N., Bohach I. V., Boiko O. R. (2012) Kompiuterne modeliuvannia system ta protsesiv. Metody obchyslen [Computer modelling of systems and processes. Computational methods]. Vinnizja: VNTU. [in Ukrainian].

5. Kostiushko I. A., Liubashenko N. D., Tretynyk V. V. (2021) Metody obchyslen [Computational methods]. K. : NTTU "KPI im. I. Sikorskoho", Vyd-vo "Politekhnika". [in Ukrainian].

6. Hlinenko, L. K., Hlinenko A. V., Sukhonosov O. H. (2003) Osnovy modeliuvannia tekhnichnykh system [Fundamentals of modeling technical systems]. – Lviv : Beskyd Bit. [in Ukrainian].

7. Chabaniuk Ya. M., Yakovyna V. S., Fedasiuk D. V. (2010) Pobudova i doslidzhennia modeli nadiinosti prohramnoho zabezpechennia z indeksom velychyny proektu [Construction and research of a software reliability model with a project magnitude index]. *Inzheneriia prohramnoho zabezpechennia*, vol. 1, pp. 1–6. [in Ukrainian].

8. Yakovyna V., Smirnov V. (2011) Ohliad osnovnykh pidkhodiv do analizu nadiinosti prohramnoho zabezpechennia [Overview of the main approaches to software reliability analysis]. *Kompiuterni nauky ta informatsiini tekhnolohii*, vol. 719, pp. 278–282. [in Ukrainian].

9. Foidl H., Felderer M. (2018) Integrating software quality models into risk-based testing. *Software Quality Journal*, vol. 26, pp. 809–847.

10. Crispin L., Gregory J. (2010) Agile Testing: A Practical Guide for Testers and Agile Teams. New York : Addison-Wesley Signature Series (Cohn).

11. Kaner K., Falk J., Nguyen H. Q. (1999) Testing Computer Software, Boston: International Thomson Publishing Press.

12. Culbertson R., Brown C., Cobb G. (2000) Rapid Testing. Atlanta: Software Quality Institute Series.

13. Kataieva Ye. Yu., Odokiienko S. M., Liuta M. V., Savchenko Ya. S. (2024) Praktychnyi analiz yakosti prohramnoho zabezpechennia z vidkrytym kodom [Practical analysis of open source software quality]. *Upravlinnia rozvytkom skladnykh system*, no 44, pp. 49–55. [in Ukrainian]. URL: https://doi.org/10.32347/2412-9933.2020.44.49-55

14. Hovorushchenko T. O. (2017) Metodolohiia otsiniuvannia dostatnosti informatsii dlia vyznachennia yakosti prohramnoho zabezpechennia [Methodology for assessing the sufficiency of information to determine software quality]. Khmelnytskyi : KhNU. [in Ukrainian].

15. Halkin P. V., Kliuchnyk I. I. (2019) Prohramuvannia PLK v CODESYS [PLC programming in CODESYS]. Kharkiv : FOP Panov A. M. [in Ukrainian].

16. Bereza A. M. (2001) Osnovy stvorennia informatsiinykh system: 2 vyd., pererob. i dop. [Fundamentals of creating information systems: 2nd ed., revised and supplemented]. K. : KNEU. [in Ukrainian].

17. Pupena O. M., Elperin I. V., Lutska N. M., Ladaniuk A. P. (2011) Promyslovi merezhi ta intehratsiini tekhnolohii v avtomatyzovanykh systemakh [Industrial networks and integration technologies in automated systems]. K. : Lira-K. [in Ukrainian].

18. Hoffmann G. M., Rajnarayan D. G., Waslander S. L., Jang J. S. (2004) The Stanford Testbed of Autonomous Rotorcraft for Multi Agent Control (STARMAC-2004). *In the Proceedings of the 23rd Digital Avionics System Conference. Salt Lake City*, UT. pp. 12–16.

19. Kossak O., Tumashova O. (2003) Metody nablyzhenykh obchyslen [Approximate calculation methods]. Lviv : Bak. [in Ukrainian].

20. Polozhaienko S. A., Prokofiev A. Yu. (2024) Parametry metodu Runhe-Kutty z riznym poriadkom tochnosti pry intehruvanni rivnian dynamiky v zadachakh modeliuvannia nestatsionarnykh system [Parameters of the Runge-Kutta method with different orders of accuracy when integrating dynamics equations in modeling problems of non-stationary systems]. *Informatyka ta matematychni metody v modeliuvanni*, vol. 14, no 1–2, pp. 85–96. [in Ukrainian]. URL: https://doi.org/10.15276/imms.v14.no1-2.85