

Висновок

Нейромережеві технології вирішують задачі розпізнавання образів та класифікації ситуацій, зокрема у системах керування, проте якість вирішення залежить від архітектури та алгоритму навчання. Запропонований підхід дозволяє класифікувати стани системи керування на основі оцінкових значень виходів системи.

Нейромережа забезпечує достатньо точне передбачення значень виходів системи керування та ставить їм у відповідність значення стану системи.

1. Васильев С.Н., Жерлов А.К., Федосов Е.А., Федунев Б.Е. Интеллектуальное управление динамическими системами. – М.: Физматлит, 2000. – 352с.
2. Ивахненко А.Г. Самонастраивающиеся системы с положительными обратными связями. – К.: Изд-во АН УССР, 1963. – 392с.
3. Widrow B., Lehr M. A 30 years of adaptive neural networks: perceptron, madaline, and backpropagation // Proceedings of IEEE. – 1990. – vol. 78, № 9. – P.1415 – 1442.
4. Білас О.Є., Томашевський О.М. Нейромережева класифікація станів та зашумлених даних у системах керування // Вісник Технологічного університету Поділля. – Т.1 (41). – 2002. – №3. – С.185 – 189.
5. Hopfield J.J. Neural networks and physical systems with emergent collective computational abilities // Proc. Natl. Acad. Sci. USA. – 1982. – № 79. – P.2554 – 2558.
6. Hinton G.E. Now Neural Network from experience // Scientific American. – September, 1992. – vol. 267, № 3. – P.185 – 234.
7. Agarwal M. A systematic classification of Neural-Network-Based control // Control Systems, 1997. – vol. 17, №2. – P.75 – 93.

С. Рендзіняк, Б. Крупський, І. Мурин
 Національний університет "Львівська політехніка"

УДК 621.372.061

ЗАСТОСУВАННЯ ПРЯМОГО ДІАКОПТИЧНОГО МЕТОДУ РОЗРАХУНКУ ДИНАМІЧНИХ РЕЖИМІВ НА ПАРАЛЕЛЬНИХ ОБЧИСЛЮВАЛЬНИХ СТРУКТУРАХ

© Рендзіняк С., Крупський Б., Мурин І., 2003

Запропоновано алгоритм паралельного прямого діакоптичного методу розрахунку динамічних режимів складних електричних кіл, реалізований в локальній комп'ютерній мережі.

The algorithm of parallel general diacoptic method of time-domain simulation of large-scale electric circuits with a local computer network is proposed.

1. Постановка проблеми

Прямі діакоптичні методи розрахунку динамічних режимів електричних кіл будуються при узгодженні розв'язків окремих підсхем на рівні дискретизованої та лінеаризованої математичної моделі досліджуваного об'єкта. Такі методи передбачають дискретизацію та лінеаризацію диференціальних нелінійних залежностей кожної підсхеми в однакових часових точках, у котрих відбувається узгодження підсхем шляхом розв'язання рівнянь зв'язку. Швидкість виконання алгоритму прямого діакоптичного методу можна суттєво підвищити, застосувавши принцип паралельних обчислень. Тому й проблема розробки програмного комплексу, який базується на паралельних діакоптичних методах і придатного для широкого класу користувачів, в наш час є актуальною.

2. Аналіз останніх досліджень

Теоретичні засади організації паралельних обчислень розроблено на достатньому рівні [1, 2], хоча питання алгоритмізації та реалізації обмежується тими обчислювальними засобами, якими володіє той чи інший дослідник. Проблема використання паралельних алгоритмів, побудованих на діакоптичних методах [3], недостатньо розроблена в науковій літературі, хоча в певних напрямках робота ведеться досить інтенсивно [2, 4].

3. Задачі досліджень

Математична модель певної динамічної системи, яка складається з нелінійних та керованих елементів, після дискретизації та лінеаризації набуває вигляду

$$\begin{pmatrix} J_{AX1} & 0 & \dots & 0 & J_{AV1} \\ 0 & J_{AX2} & \dots & 0 & J_{AV2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & J_{AXN} & J_{AVN} \\ A_{F1} & A_{F2} & \dots & A_{FN} & -1 \end{pmatrix} \begin{pmatrix} \Delta X_1 \\ \Delta X_2 \\ \vdots \\ \Delta X_N \\ \Delta V \end{pmatrix} = \begin{pmatrix} F_1(X_1^j, V_1^j) \\ F_2(X_2^j, V_2^j) \\ \vdots \\ F_N(X_N^j, V_N^j) \\ \sum_{i=1}^N A_{Fi} \cdot X^j - V^j \end{pmatrix}, \quad (1)$$

де X_i^j – внутрішні змінні i -ї підсхеми на деякому кроці інтегрування та j -й ітерації; V – зовнішні змінні, що встановлюють зв'язок між підсхемами, який описується останнім

рівнянням системи (1); $J_{AXi} = \left. \frac{\partial F(X_i^j, V^j)}{\partial X_i^j} \right|_{X_i^j}$, $J_{AVi} = \left. \frac{\partial F(X_i^j, V^j)}{\partial V^j} \right|_{V^j}$ – відповідні

якобіани ітераційного методу Ньютона-Рафсона; $i = 1, \overline{N}$, N – число підсхем. Зауважимо, що загалом рівняння стану підсхем містять залежності не від усіх змінних

V^j , а лише від певної підмножини $V^j \in R^{NV}$ елементів цього вектора ($NV = \sum_{i=1}^N NV_i$).

Однак рівняння зв'язку є функціями усіх компонент вектора V^j .

Задача досліджень полягає в побудові алгоритму і організації обчислень системи (2) на паралельній обчислювальній структурі, а саме в локальній мережі типу *Novell Netware* чи *Microsoft Windows*, і визначення їх ефективності.

4. Виклад основного матеріалу

Хоча з першого погляду алгоритм прямого інтегрування не є цілком придатним для виконання паралельними ЕОМ, його також можна адаптувати до роботи на паралельних обчислювальних комплексах. Так, у праці [5] розглядаються принципи побудови паралельних алгоритмів, які орієнтовані на конвеєрні обчислювальні системи. Водночас автори цих публікацій нехтують аналізом часової ефективності виконання запропонованих підходів. Не знаходимо у цих працях й описів алгоритмів для числового інтегрування систем алгебро-диференціальних рівнянь у формі (1). Тому проведемо адаптацію запропонованих у [6] підходів для розв'язання такої задачі на багатопроекторних обчислювальних системах, оцінюючи трудомісткість розробленого алгоритму.

Із перших N рівнянь (1) знаходимо прирости внутрішніх змінних стану підсхем

$$\Delta X_i^j = J_{AXi}^{-1} \left(F_i \left(X_i^j, V_i^j \right) - J_{AVi} \Delta V^j \right), \quad (2)$$

а рівняння зв'язку розв'яжемо відносно

$$\Delta V^j = \sum_{i=1}^N A_{Fi} \Delta X_i^j - \sum_{i=1}^N A_{Fi} X_i^j + V_i^j, \quad (3)$$

Із (2)–(3) можна знайти прирости значень зовнішніх змінних на кожній ітерації

$$\Delta V^j = \sum_{i=1}^N A_{Fi} J_{AXi}^{-1} \left(F_i \left(X_i^j, V_i^j \right) - J_{AVi} \Delta V^j \right) - \sum_{i=1}^N A_{Fi} X_i^j + V_i^j. \quad (4)$$

Об'єднаємо (2) і (4) у систему рівнянь

$$\left\{ \Delta X_i^j = J_{AXi}^{-1} \left(F_i \left(X_i^j, V_i^j \right) - J_{AVi} \Delta V^j \right), \right. \quad (5)$$

$$\left. \left\{ 1 + \sum_{i=1}^N A_{Fi} J_{AXi}^{-1} J_{AVi} \right\} \Delta V^j = \sum_{i=1}^N A_{Fi} \left\{ J_{AXi}^{-1} F_i \left(X_i^j, V_i^j \right) - X_i^j \right\} + V_i^j. \right. \quad (6)$$

Оскільки рівняння (5) відносно ΔX_i^j містять лише невідомі прирости зовнішніх змінних, то після знаходження ΔV^j їх розв'язання можливе одночасно для усіх $i = 1, \overline{N}$.

Розв'язання першого рівняння відносно ΔV^j також можна розпаралелити. Знаходити невідомі коефіцієнти

$$X_{Gi} = J_{AXi}^{-1} J_{AVi}, \quad (7)$$

$$P_{Vi} = A_{Fi} J_{AXi}^{-1} J_{AVi}, \quad (8)$$

$$D_{Vi} = A_{Fi} J_{AXi}^{-1} F_i \left(X_{m+1}^j, V_{m+1}^j \right) \quad (9)$$

для кожної із підсхем можна одночасно і незалежно від інших підсхем. Оскільки перемноження матриць, знаходження обернених матриць та якобіанів є досить трудомісткими операціями, можна очікувати значного пришвидшення роботи програми.

Матриці X_{G_i} , P_{V_i} , D_{V_i} мають розмірності: $X_{G_i} \in R^{N_X \times N_{V_i}}$, $P_{V_i} \in R^{N_V \times N_{V_i}}$, $D_{V_i} \in R^{N_V}$. В подальшому позначатимемо розміри векторно-масивних елементів так: $[X_{G_i}] = N_X \times N_{V_i}$, $[P_{V_i}] = N_V \times N_{V_i}$, $[D_{V_i}] = N_V$. Вирази ж типу

$$[K \times M] = [K \times L] \cdot [L \times M]. \quad (10)$$

розшифруватимемо як множення двох матриць з розмірностями $K \times L$, $L \times M$.

Як відомо, обчислення добутку двох прямокутних матриць (10) вимагає виконання $K \cdot L \cdot M$ дій множення та $K \cdot M \cdot (L - 1)$ операцій додавання. Будемо вважати, що часові затрати на виконання усіх елементарних математичних операцій однакові. Отже множення матриць потребує виконання $K \cdot M \cdot (2L - 1)$ математичних дій.

Для створення ефективних обчислювальних алгоритмів необхідно ретельно досліджувати усі способи зменшення кількості математичних операцій та розмірів додаткових змінних, зокрема зміною послідовності виконання математичних дій. Як можна помітити, сумісне обчислення матриць X_{G_i} , P_{V_i} , D_{V_i} можна проводити трьома способами, що характеризуються різними обчислювальними затратами. Проаналізуємо детальніше один із способів з найменшим числом математичних операцій.

Вважаємо, що значення змінної X_{G_i} знайдено за формулою (7). Тоді його можна використати для подальших обчислень так, щоб зменшити кількість дій для знаходження вектора P_{V_i} :

$$P_V = A_F X_G, \quad (11)$$

Спочатку виконуватимемо математичні дії над тими множниками у формулі (8), які знаходяться правіше.

$$B_V = J_{AX}^{-1} F(X, V), \quad (12)$$

$$D_V = A_F B_V. \quad (13)$$

У цьому випадку додаткова змінна B_V має розмірність $[N_X \times 1]$, а впорядкований таким чином алгоритм складається із

$$Z_2 = N_V N_{V_i} (2N_{X_i} - 1) + N_{X_i} (2N_{X_i} - 1) + N_V (2N_{X_i} - 1) \quad (14)$$

арифметичних дій.

Переваги цього способу організації алгоритму зберігаються при будь-яких розмірах аналізованих підсхем. Навіть у найтривіальнішому випадку, коли $N_V = 2$, а $N_{X_i} = 1$, одержимо виграш у швидкодії.

Отже, алгоритм прямого діакоптического методу містить чотири етапи виконання:

I. Обчислення коефіцієнтів для знаходження приростів зовнішніх змінних (паралельне виконання для кожної із N підсхем).

1. Знаходимо коефіцієнти матриці Якобі $J_{AX_i} \in (R^{N_{X_i} \times R^{N_{X_i}}})$ та $J_{AV_i} \in (R^{N_{X_i} \times R^{N_{V_i}}})$ для нелінійних підсхем.
2. Знаходимо значення вектора нев'язки $F_i(X_i^j, V_i^j)$.
3. Обчислюємо обернену матрицю Якобі $J_{AX_i}^{-1}$.
4. Знаходимо значення виразів: $X_{G_i} = J_{AX_i}^{-1} J_{AV_i} ([X_{G_i}] = N_{X_i} \times N_{V_i})$.
5. $P_{IT_i} = J_{AX_i}^{-1} F_i$ де вектор $P_{IT_i} \in R^{N_{X_i}}$.
6. $B_{V_i} = P_{IT_i} - X_i$, де розмірність вектора $B_{V_i} \in R^{N_{V_i}}$.
7. $D_{V_i} = A_{F_i} B_{V_i}$, вектор D_{V_i} містить N_{V_i} елементів.
8. $P_{V_i}^i = A_{F_i} X_{G_i}^i (P_{V_i}^i \in (R^{N_{V_i}} \times R^{N_{V_i}}))$.
9. Пересилання векторів $P_{V_i}, D_{V_i}, P_{IT_i}, X_{G_i}$ керуючому процесорові.

II. Послідовна робота алгоритму – знаходження приростів зовнішніх змінних підсхем.

1. Збираємо матрицю лінійної алгебраїчної системи рівнянь (6): $P_V = (P_{V_1}, P_{V_2}, \dots, P_{V_N})$, де $P_{V_i} \in R^{N_{V_i}} \times R^{N_{V_i}}$.
2. $P_V = P_V + I$, тут I – одинична матриця.
3. $D_V = D_{V_1} + \dots + D_{V_N}$, де $(D_{V_i} \in R^{N_{V_i}})$.
4. Знаходимо розв'язок рівняння $P_V \Delta V = D_V$.
5. Коректуємо значення вектора зовнішніх змінних згідно з аналогом формули (4): $V^{j+1} = V^j - \Delta V$.
6. Пересилаємо знайдені вектори V_i^{j+1} та ΔV_i відповідним процесорам обчислювальної системи.

III. Паралельна робота програми – розрахунок внутрішніх змінних стану підсхем та модулів суми приростів змінних.

1. $P_{IT_i} = X_{G_i} \Delta V_i - P_{IT_i}$;
2. $X_i^{j+1} = X_i^j + P_{IT_i}$;
3. Розраховуємо модуль сумарного приросту внутрішніх змінних підсхеми

$$P_{F_i} = \sum_{i=1}^{N_{X_i}} |P_{IT_i}|;$$

4. Знаходимо сумарний приріст вихідних зовнішніх змінних $V_F = \sum_{j=1}^{N_{V_i}} |\Delta V_j|$.

5. Передаємо X_i, P_{F_i}, V_F головному процесорові.

IV. Послідовне виконання – оцінка збіжності ітераційного процесу.

1. Якщо точність ітераційного процесу відповідає таким умовам: $\sum_{i=1}^N P_{Fi} \leq \varepsilon_X$

і $\sum_{i=1}^N V_{Fi} \leq \varepsilon_V$, то завершуємо ітерації у даній часовій точці і переходимо до

наступного кроку дискретизації рівнянь (1).

2. Розпочинаємо нову ітерацію виконанням пункту 1.1.

Запропонований алгоритм розпаралелення прямого діакоптического методу складається із чотирьох етапів, два з яких допускають незалежне і паралельне виконання на різних процесорних модулях багатопроцесорної системи. Ефективність паралельної реалізації залежить від співвідношень між тривалістю виконання послідовної та паралельної частин комп'ютерної програми. Порівняння запропонованої паралельної реалізації прямого діакоптического методу із подібними підходами дозволяє говорити про його повнішу адаптацію до роботи на кластерних та розподілених обчислювальних системах

внаслідок зменшення кількості сеансів обміну інформацією між паралельними процесорними модулями. Так, реалізація алгоритму інтегрування, запропонована у праці [7], передбачає виконання елементарної ітерації методу протягом шести етапів обчислень, побудованих за схемою паралельне – послідовне – паралельне – послідовне – паралельне – послідовне виконання. Зменшення кількості паралельних етапів виконання призводить до зменшення витрат на синхронізацію процесорів

паралельних ЕОМ, полегшує написання програмного коду, зменшує імовірність появи помилок програмування, пов'язаних із синхронізацією паралельних процесів та достовірністю даних.

Ефективність запропонованої паралельної реалізації алгоритму інтегрування прямого діакоптического методу, а також можливі способи підвищення ефективності використання обчислювальних ресур-

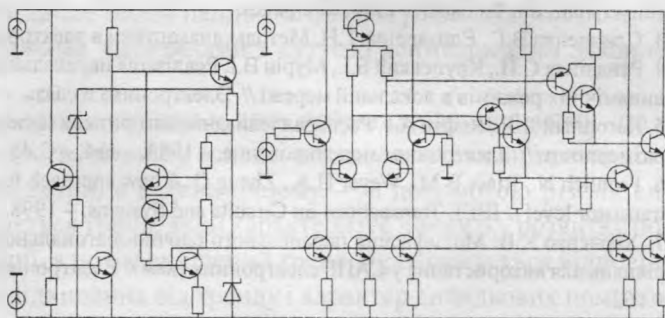


Рис. 1. Підсилювач напруги типу UA 727

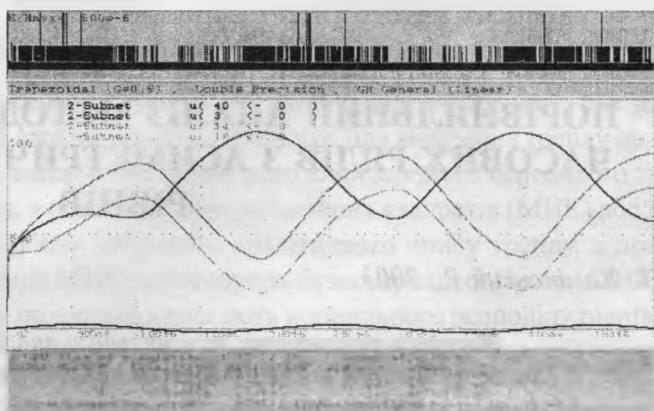


Рис. 2. Приклад розрахунку перехідного процесу в розщепленому на три підсхеми підсилювачі

сів багатопроцесорного комплексу при виконанні програмного коду досліджувалися за допомогою розробленого програмного комплексу зі застосуванням команд IPX/SPX-протоколів та системи вводу/виводу NetBIOS локальної мережі на прикладі розрахунку перехідного процесу складного підсилювача, розщепленого на три підсхеми (рис. 1, 2)

5. Висновки

Розроблений алгоритм діакоптичного методу на паралельних обчислювальних структурах дозволяє ефективно використовувати поширене комп'ютерне обладнання і відповідає сучасним тенденціям розвитку обчислювальних методів та техніки.

1. Параллельная обработка информации: В 5 т. // Под ред. А.Н.Свенсона. – К.: Наук. думка. 1985. – Т.1: Распаралеливание алгоритмов обработки информации. – 280 с.
2. D.Petcu. Parallelizs in solving ordinary differential equations, Mathematical Monographs 64, Tipografia Universitatii din Timisoara, 1998, 232 pages.
3. Слипенко В.Г., Елизаренко Г.Н. Методы диакоптики в электронике. – К.:Вища школа, 1981. – 207 с.
4. Рендзіняк С.Й., Крупський Б.І., Мурін В.І. Реалізація паралельних діакоптичних алгоритмів розрахунку динамічних режимів в локальній мережі // Електроніка і зв'язь. – 1998. – Вып. 4. – Ч.3. – С. 415 – 418.
5. Нагорный Л.Я., Кофто А.Г. Распаралеливание алгоритмов моделирования нелинейных систем большой размерности // Электронное моделирование. – 1983. – №4. – С.45 – 51.
6. Frohlich N., Riess B.M., Wever U.A., Zheng Q. A new approach for parallel simulation of VLSI circuits on a transistor level // IEEE Transactions on Circuits and Systems. – 1998. – Vol. CSI-45. – No.6. – pp. 601 – 613.
7. Харченко К.В. Модифікація паралельного блочно-діагонального методу розв'язання лінійних систем рівнянь для використання у САПР електронних схем // Електроніка та зв'язок. – 1999. – № 6. – Т. 1. – с. 156 – 162.

Р. Камінський

Національний університет "Львівська політехніка"

УДК 519.24

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ ЗГЛАДЖУВАННЯ ЧАСОВИХ РЯДІВ З АСИМЕТРИЧНИМ РОЗПОДІЛОМ РІВНІВ

© Камінський Р., 2003

Наведено результати порівняльного аналізу методів згладжування часових рядів з асиметричними розподілами рівнів методами ковзного середнього, експоненціального згладжування та медіанної фільтрації за критеріями: вибіркової медіани, загальної дисперсії, дисперсії різниці рівнів, критерію поворотних точок та показника фрактальності Херста.