

ОЦІНКА ЕФЕКТИВНОСТІ МІКРОПРОГРАМУВАННЯ ЙМОВІРНИХ ПЕРЕХОДІВ ПОСЛІДОВНОСТІ ЗАЛИШКІВ

Я.М.Николайчук, Г.Я.Ширмовський, О.І.Іванкевич, Н.Г.Ширмовська

ІФНТУНГ, 76019, м. Івано-Франківськ, вул. Карпатська, 15, тел.(0342) 504521

e-mail: public@nimg.edu.ua

Предлагается один из подходов оценки эффективности микропрограммирования вероятности изменений значений вычетов ординат представления системы остаточных классов. Постановка задачи связана с возможностью реализации метода на специализированном вычислительном устройстве, требующем микропрограммного математического обеспечения.

One of approaches of estimation of effective microprogramming probability of changes of values of take-outs of ordinates of presentation of the system of remaining classes is offered in the article. Raising of task is related to marketability method on the specialized computing device, requiring the microprogramming mathematical providing.

Із розвитком рівня інтелекту інформаційно-вимірjuвальних систем (ІВС) бурових виробництв все більшого визнання набуває організація інформаційного забезпечення їх виробничо-господарської діяльності на сучасній основі [1]. Інформаційне забезпечення необхідне розглядати в тісному взаємозв'язку з технічним рівнем засобів ІВС, програмними засобами і технологією вирішення задач управління буровим виробництвом.

Пропонується мікропрограмний підхід обчислення значень ординат системи залишкових класів, що є математично-інформаційним підходом реалізації задачі на спеціалізованому обчислювальному пристрої з використанням спеціалізованого математичного забезпечення.

Під мікропрограмою будемо вважати впорядковану сукупність мікрокоманд (керуючих слів), що описують алгоритм виконання операцій, реалізація яких дасть змогу моделювати процес зміни значень залишків b_{ij} , представлених деревом переходів (рис. 1) [2].

Нехай граф алгоритму представляється відповідно до правил, описаних [2]. Тоді граф алгоритму ймовірних переходів послідовності залишків буде містити одну початкову й кінцеву вершини, а також сукупність операторних і умовних вершин. Операторна вершина (Y-оператор) містить набір сумісних мікрооперацій $Y_{\varphi} = \{y_1, \dots, y_n\}$, які підтверджують незмінність значень $b_{i-ij}, b_{ij}, b_{i+ij}$ послідовності залишків через часові інтервали Δt . Умовна вершина (X-оператор) містить єдину логічну умову з множини $X = \{x_1, \dots, x_L\}$, яка визначає можливість переходу обчислення b_{ij} в бік збільшення або зменшення на величину $b \cdot p_j$.

Розглянемо загалом мікрокоманду як слово із структурою, зображеною на рис. 2, де y_1, y_2, \dots – операційна частина (поле Y), мікрокоманди x_1, x_2, \dots , та A – адресна частина мікрокоманди.

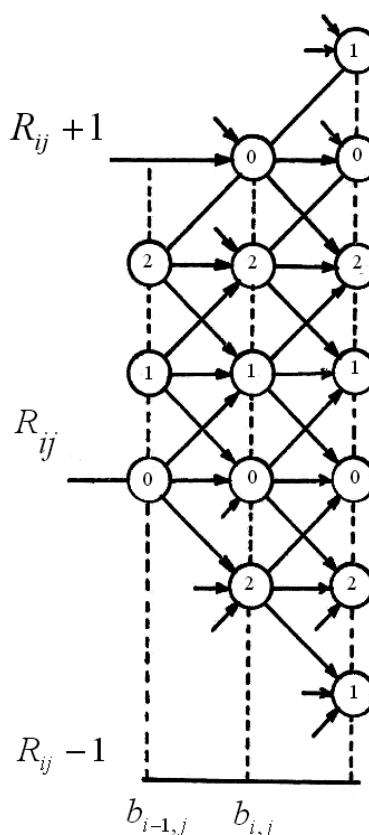


Рисунок 1 – Приклад дерева ймовірних переходів послідовності $b_{i-1,j} \rightarrow b_{i,j} \rightarrow b_{i+1,j}$

y_1	y_2	...	x_1	x_2	...	A
-------	-------	-----	-------	-------	-----	---

Рисунок 2 – Структура мікрокоманди

Одним зі способів задання адреси може бути вихід "1" або "0" логічної умови x_q поля X мікрокоманди. Так, для випадку $1 \rightarrow 1 \rightarrow 2$ дерева ймовірних переходів послідовності залишків $b_{i-ij}, b_{ij}, b_{i+ij}$ (рис. 3), вихід "1" логічної умови x_1 (рис. 4), вкаже на збільшення b_{ij} на одиницю.

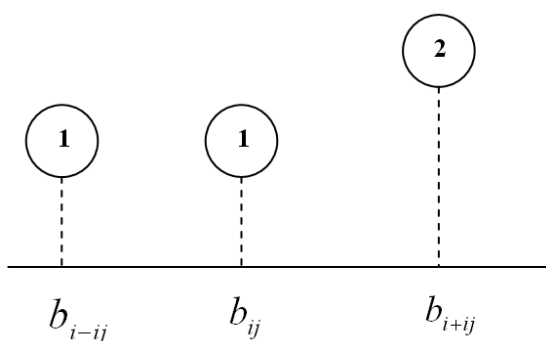


Рисунок 3 – Фрагмент дерева ймовірних переходів

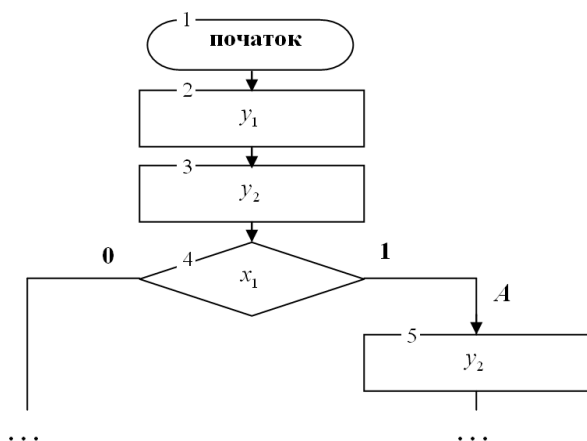


Рисунок 4 – Граф алгоритму ймовірних переходів

Як бачимо з рис. 4 граф алгоритму ймовірних переходів послідовністю операторних вершин y_1 і y_2 підтверджує збереження обчисленнями b_{i-ij} і b_{ij} значень одиниці. Операторна вершина y_3 дасть змогу залишкові b_{i+ij} прийняти значення 2, тобто привести до відповідності значення адреси A і поля Y наступної мікрокоманди.

Позначивши через $H = 1, 2, \dots, i$ і $G = 1, 2, \dots$ кількість полів Y і X і надалі будемо характеризувати структуру мікрокоманди парою значень (H, G) .

Кількість мікрокоманд залежить від структури мікрокоманди, зокрема, від кількості H і G полів Y і X та від складності алгоритму, обумовленої сумарним числом Y - і X -операторів, що складають алгоритм.

Запропонований підхід, заснований на визначенні довжини мікропрограми без використання процедури мікропрограмування, є менш затратним. Тоді кількість операторів n , що кодуються однією мікрокомандою, яка містить H полів Y і G полів X , лежить у межах $1 \leq n \leq (H + G)$ і залежить від числа операторних і умовних вершин у графі, а також від конфігурації графа. Досить просте рішення розглянутої задачі можна одержати, якщо складність алгоритму характеризувати тільки кількістю його складових операторів $n = n_1 + n_2$, де n_1 –

кількість Y -операторів; n_2 – кількість X -операторів; а конфігурацію графа (дуги між операторами) враховувати тільки в першому наближенні. Характеризуючи складність алгоритму тільки значеннями n_1 і n_2 , можна визначити лише кількість мікрокоманд, яка необхідна для кодування сукупності Y - і X -операторів, що складають сам алгоритм, і важко виявити кількість мікрокоманд, яка необхідна для кодування специфічних переходів (безумовних переходів у разі природної адресації, переходів між сторінками у разі сторінкової адресації і тощо).

Для визначення залежності середньої кількості операторів, що покриваються однією мікрокомандою, від характеристик алгоритму доцільно використати такий спосіб.

Нехай алгоритм, який складається з n операторів, містить n_1 операторних вершин і n_2 умовних вершин. Відношення

$$p_1 = \frac{n_1}{n}, p_2 = \frac{n_2}{n}, p_1 + p_2 = 1$$

будемо розглядати як ймовірність появи Y - і X -операторів в алгоритмі, тобто будемо вважати, що оператор з номером $s = 1, 2, \dots, n$ з ймовірністю c_1 є Y -оператором і з ймовірністю c_2 – X -оператором.

На основі ймовірнісної моделі можна визначити кількість операторів, які покриваються однією мікрокомандою. Мікрокомандою, що містить H полів Y і G полів X можна закодувати від 1 до $(H + G)$ операторів.

Розглянемо модель процесу породження мікрокоманд на прикладі структури мікрокоманди, яка має по одному полю для кодування Y - і X -операторів, тобто для мікрокоманд типу $(H, G) = (1, 1)$.

У випадку використання мікрокоманд типу $(H, G) = (1, 1)$ різні фрагменти графа алгоритму можуть бути закодовані мікрокомандами з різним вмістом полів Y і X . Фрагменти графа алгоритму рис. 5, а, б, в, кодуються відповідно такими наборами мікрокоманд:

- а) $Y_c \cdot 0 \cdot A$; б) $0 \cdot X_q \cdot A$; в) $Y_c \cdot X_q \cdot A$.

Тут 0 позначає порожнє поле в мікрокоманді.

Оскільки дві сусідні вершини не можуть бути закодовані однією мікрокомандою, то для побудови моделі процесу породження мікрокоманд необхідно знати ймовірності цих подій. Ці ймовірності визначаються в такий спосіб.

З кожної операторної вершини виходить строго одна дуга, а з кожної умовної вершини – строго дві дуги. Звідси ймовірність того, що в оператор входить більше однієї дуги дорівнює

$$p_2 = \frac{n_2}{n}$$

Тепер визначимо ймовірності того, що деяка дуга виходить з Y - або X -оператора. Загальна кількість дуг у графі алгоритму (не включаючи початкової вершини) дорівнює

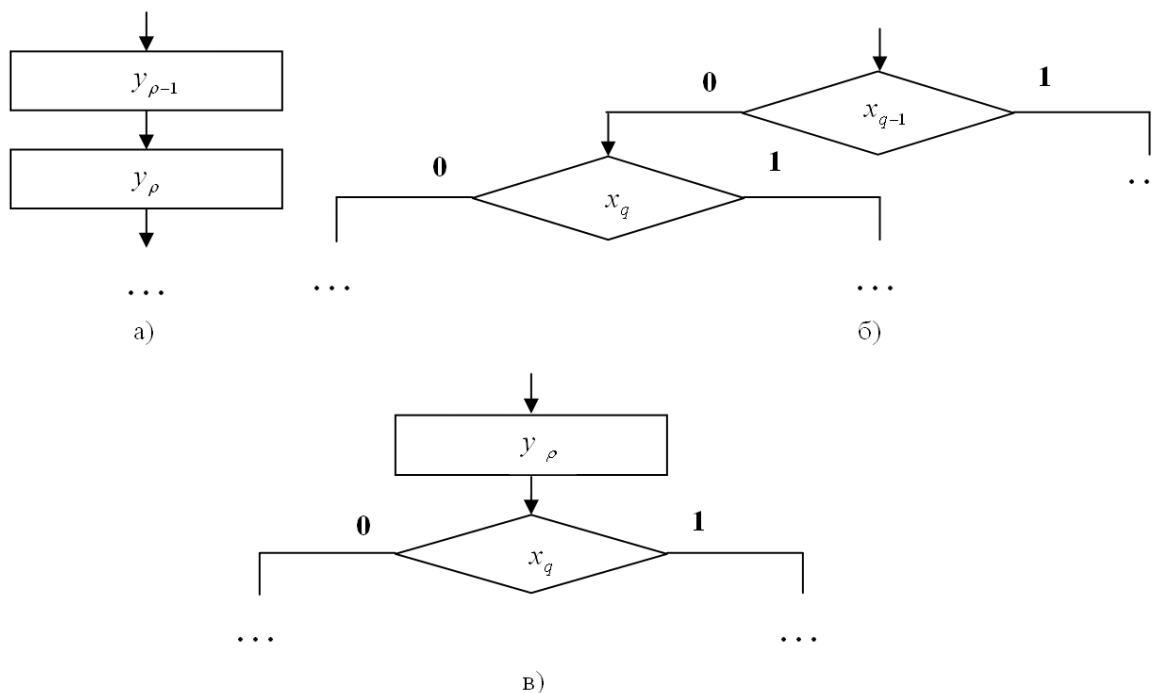


Рисунок 5 – Фрагменти графу кодування полів Y і X

$n_1 + 2n_2$. Тоді ймовірності того, що дуга виходить із умовної й операторної вершин рівні відповідно

$$p_3 = \frac{2n_2}{n_1 + 2n_2}, p_4 = \frac{n_1}{n_1 + 2n_2}, p_3 + p_4 = 1.$$

Виходячи з ймовірностей p_1, \dots, p_4 , можна побудувати модель процесу породжень мікрокоманд кожного типу. При цьому з ймовірністю p_1 з'являється Y-оператор і з ймовірністю p_2 – X-оператор. Перевіряється наявність більше однієї дуги, що веде в оператор, наступний за вже існуючим. Ця подія відбувається з ймовірністю p_2 . Потім визначається з якого оператора виходить ця додаткова дуга. Так з ймовірністю p_3 оператор буде умовним і з ймовірністю p_4 – Y-оператором.

Якщо додаткової дуги немає або ж вона є й виходить із Y-оператора, то здійснюється перехід до наступної вершини графу, що з ймовірністю p_1 буде Y-оператором і з ймовірністю p_2 – X-оператором. Дія повторюється доти, поки не з'являються ситуації, за яких необхідно породити мікрокоманду, а саме:

- кількість Y-операторів дорівнює кількості H полів Y в мікрокоманді й знову з'явився Y-оператор;
- кількість X-операторів дорівнює кількості G полів X у мікрокоманді;
- сумарне число Y- і X-операторів дорівнює сумі (H + G) визначальне число Y і X полів у мікрокоманді;
- з'явилася дуга, що веде з умовного оператора;

– кількість X-операторів менша числа G-полів X в мікрокоманді і з'явився Y-оператор.

Розглянемо процес породження мікрокоманд для випадку, коли мікрокоманда має по одному полю Y і X (рис. 6). Процес породження мікрокоманд починається від початку графу алгоритму (точка 0 на рис. 6). Y-оператор з'являється з ймовірністю p_1 і X-оператор – з ймовірністю p_2 . X-оператор кодується мікрокомандою 0.X.A і дія знову повертається в початкову точку 0. Якщо з'явилася операторна вершина, то здійснюється перевірка на наявність більше однієї дуги, що входить в оператор, наступних за Y-оператором. Наявність додаткової дуги позначимо у вигляді $Y \downarrow$. Ця дуга з ймовірністю p_3 виходить із X-оператора й з ймовірністю p_4 – з Y-оператора. Будемо позначати тип оператора, з якого виходить додаткова дуга, з стрілкою, а саме: $\downarrow(X)$ і $\downarrow(Y)$. Якщо додаткова дуга виходить із X-оператора, то необхідно з Y-оператора сформувати мікрокоманду Y.0.A і повернутися в початкову точку 0. Якщо дуга має вигляд $\downarrow(Y)$, то породжується оператор, наступний за Y-оператором, причому із ймовірністю p_1 – це буде Y-оператор і з ймовірністю p_2 – X-оператор. Аналогічна дія здійснюється й у тому випадку, якщо додаткової дуги не існує. Групи операторів виду YX і $Y \downarrow(Y)X$ породжують мікрокоманду одного виду Y.X.A і дія повертається в точку 0. Поява двох операторних вершин ви-

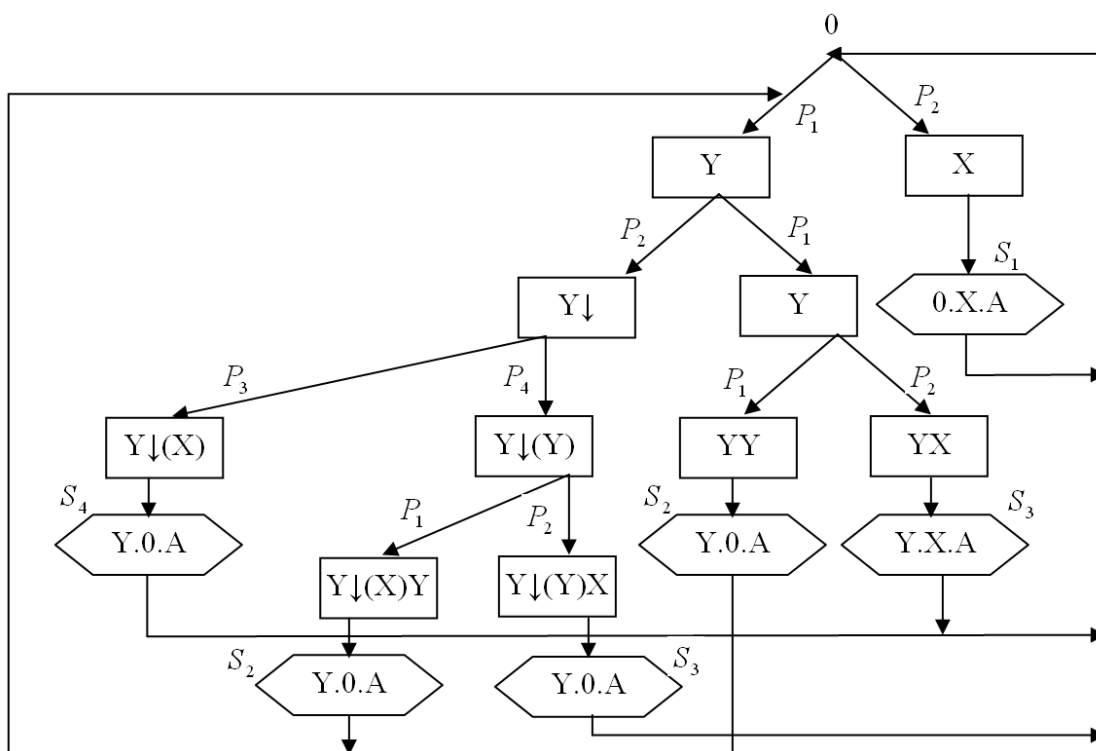


Рисунок 6 – Процес породження мікрокоманд $(H, G) = (1, 1)$

ду YY чи $Y \downarrow(Y)Y$ породжує мікрокоманду виду $Y.O.A$ і повертає процес до аналізу вершини, наступної за операторною. Породження мікрокоманди кожного типу будемо трактувати як стан S_1, S_2, \dots процесу кодування алгоритма. У розглянутих ймовірнісних моделях кожний перехід у графі рис. 6 не залежить від передісторії, тому процес, який розвивається на множині станів $\{S_1, S_2, \dots\}$, є марківським ланцюгом. Зазначимо, що стани марківського ланцюга S_1, S_2, \dots є ергодичними станами.

Позначимо ймовірності перебування процесу в станах S_1, S_2, \dots через V_1, V_2, \dots відповідно. Для ергодичних ланцюгів Маркова вірне таке співвідношення:

$$V_c = \sum_{q=1}^Q V_q P_{qc}, \quad (1)$$

де P_{qc} – ймовірність переходу із стану S_q у стан S_c , зумовлена за правилом множення ймовірностей незалежних подій. Застосування цього правила для процесу (рис. 6) зводиться до знаходження шляху на графі, що веде зі стану S_q в стан S_c . Ймовірність

$$P_{qc} = \sum (C_6, C_b, \dots, C_{ш}),$$

де $C_6, C_b, \dots, C_{ш}$ – ймовірності, якими відзначені дуги, які належать шляхам з S_q і S_c . Так для моделі (рис. 6)

$$P_{13} = p_1 p_1 p_3 + p_1 p_2 p_4 p_2 = p_1 p_2 (p_1 + p_2 p_4).$$

Ймовірності переходу P_{qc} можуть бути розкладені в матрицю ймовірностей переходів:

$$P_{qc} = \begin{pmatrix} P_{11} & P_{12} & \dots & P_{1Q} \\ P_{21} & P_{22} & \dots & P_{2Q} \\ \dots & \dots & \dots & \dots \\ P_{Q1} & P_{Q2} & \dots & P_{QQ} \end{pmatrix}. \quad (2)$$

Ця матриця визначає в статистичному змісті процес породження мікрокоманд різного типу, що відповідають станам S_1, \dots, S_Q .

Важливо провести оцінку ступеня відповідності запропонованої ймовірнісної моделі алгоритму будь-яким реальним структурам алгоритмів. Реальну структуру, що складається з n_1 Y -операторів і n_2 X -операторів, можна розглядати як одну реалізацію дискретного випадкового процесу, який полягає в тому, що вибирається $n = n_1 + n_2$ операторів, при цьому з ймовірністю p_1 обирається Y -оператор і з ймовірністю $p_2 = 1 - p_1$ – X -оператор. Реалізація цього процесу породить алгоритм, що містить n_1^* Y -операторів і n_2^* X -операторів. Через випадковість процесу вибору конкретна реалізація не буде збігатися з алгоритмом-оригіналом, тобто загалом $n_1^* \neq n_1$ і $n_2^* \neq n_2$. Таким чином, відповідність моделі і алгоритму зводиться до визначення ймовірності того, що породжуваний моделлю алгоритм збігається з наперед заданою точністю з алгоритмом-оригіналом, на основі якого були визначені ймовірності p_1 і p_2 появи операторів.

Ця ймовірність може бути визначена в такий спосіб. Нехай n – кількість операторів в алгоритмі і p_1 – частка Y -операторів. Ймовірність того, що в результаті n -кратної вибірки з поверненням буде отримано точно q операторів визначається таким виразом:

$$P_r(q) = \binom{n}{q} p_1^q (1 - p_1)^{n-q}.$$

Ймовірність того, що значення q відхилиться від n , не більше ніж на k операторів:

$$P_r(n_1 - k \leq q \leq n_1 + k) = \sum_{q=n_1-k}^{n+k} P_r(q),$$

де $n_1 = p_1 n$

Нехай d відсоток відхилення числа Y -операторів у вибірці. Тоді $k = \frac{d n_1}{100} - u$,

$$P_r = \sum_{p_1 n (1 - \frac{d}{100})}^{p_1 n (1 + \frac{d}{100})} \binom{n}{q} p_1^q (1 - p_1)^{n-q}. \quad (3)$$

Визначимо ймовірність відхилень для різних значень $d = 0; 2; 5$ і 10% . Результати табулювання функцій (2) свідчать, що ступінь збігу моделі з алгоритмом збільшується зі збільшенням довжини алгоритму. Мінімальні довжини алгоритмів, при яких з ймовірністю 90% породжувані моделлю реалізації відрізняються від алгоритму-оригіналу не більше ніж на $d\%$. Таким чином, застосування ймовірнісної моделі справедливо у тих випадках, коли довжина алгоритму $n \geq n_{\min}(p_1, d)$.

Припустимо, що процес породження мікрокоманд заданий моделлю виду рис. 6, на основі якої визначені:

- стани S_1, S_2, \dots, S_Q
- матриця P_{qc} ймовірностей переходів (2).

Ймовірності перебування V_1, \dots, V_Q процесу в станах S_1, S_2, \dots, S_Q зв'язані системою рівнянь (1) і відповідають нормуючому рівнянню:

$$\sum_{q=1}^Q V_q = 1. \quad (4)$$

Розв'язуючи $(Q-1)$ рівняння системи (1) разом з рівнянням (4) одержуємо аналітичні вирази для ймовірностей V_1, \dots, V_Q . Виходячи із цих ймовірностей можна визначити середню кількість операторів алгоритму, які кодуються однією мікрокомандою типу (H, G) .

Обчислення для різних значень n відбувалося із метою виявлення ступеня стабільності статистичної оцінки $g(H, G)$ залежно від розміру n вибірки – кількості операторів у модельованому алгоритмі. Значення $g(H, G)$ визначається для мікрокоманд із заданим числом H і G полів Y і X від частки p_1 Y -операторів в алгоритмі. Конкретні оцінки кількості операторів дали змогу значно скоротити об'єм інформаційних повідомлень спільного файлу техніко-економічної інформації та технологічних параметрів процесу буріння, прив'язаних до поточного часового виміру, що наближається до реального часу.

Література

- 1 Николайчук Я.Н., Процюк В.Р., Ширмовський Г.Я. Эффективное кодирование суточного рапорта бурового мастера для формализованного ввода в ЭВМ. – М.: ВНИИОЭНГ, 1986. – 256 с.
- 2 Баранов С.И., Скляр В.А. Цифровые устройства на программируемых БИС с матричной структурой. – М.: Радио и связь, 1986. – 272 с.: ил.

15-а Міжнародна конференція з автоматичного управління

АВТОМАТИКА-2008 AUTOMATICS-2008

м. Одеса
(23–26 вересня 2008 р.)

Оргкомітет конференції

Одеська національна морська академія,
65029, м. Одеса, вул. Дідріхсона, 8

fa@ma.odessa.ua
leovish@rambler.ru
тел. (048) 728 25 95

Тематика конференції:

- Математичні проблеми управління, оптимізації і теорії ігор
- Управління та ідентифікація в умовах невизначеності
- Автоматичне управління в технічних системах
- Управління аерокосмічними, морськими та іншими рухомими об'єктами
- Управління в соціальних та економічних системах
- Прогресивні інформаційні технології та інтелектуальне управління
- Підготовка кадрів в галузі управління та автоматизації
- Автоматизація навчання і тренажерна підготовка