БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Abramenko T, Gorish A., Bogush M., Mitko V.* Main characteristics analysis of the piezoelectric sensors under finite-element method. – Proceeding of the Tenth International Congress on Sound and Vibration, Stockholm, Sweden, 2003, v.3, p 951-959.
2. Пьезоэлектрическое приборостроение: Сборник в 3 томах. Т. 3. Богуш М.В. Пьезоэлектрические датчики для экстремальных условий эксплуатации. Ростов-на-Дону. Изд-во СКНЦ ВШ, 2006. 346 с: ил.
3. *Туричин А.М.* Электрические измерения неэлектрических величин. – М. –Л.: Энергия, 1966. – С.52-61
4. *Левшина К.С., Новицкий И.В.* Электрические измерения физических величин. – М.: Энергоатомиздат, 1973. – С. 107-130.
5. Проектирование датчиков для измерения механических величин/ Под общ. ред. Е.П. Осадчего. – М.: Машиностроение, 1979. – 480 с.
6. Датчики теплофизических и механических параметров: Справочник в трех томах / Под общ. ред. Ю.Н. Коптева; Под ред. Е.Е. Богдатьева, А.В. Гориша, Я.В. Малкова. – М.: ИПРЖР. Т.1 (кн.2). 1998. – 512 с., Т.2. 1999. – 688 с.
7. Пьезоэлектрические приборы для измерения давлений, усилий, ускорений. Проспект фирмы Kistler Insnrument AG., 1999. – 12 с.
8. Физические величины: Справочник / А.П. Бабичев, Н.А. Бабушкина, А.М. Братковский и др.; Под ред. И.С. Григорьева, Е.З. Мейлихова. – М.: Энергоатомиздат, 1991. – 1232 с.

**Н.С. Анишин, И.Н. Булатникова, Н.Н. Гершунина**

## РАЗНОСТНО-ИТЕРАЦИОННЫЕ АЛГОРИТМЫ ДЛЯ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

Широкое внедрение микропроцессоров (МП) и микроконтроллеров (МК) в системы автоматизации и управления технологическими процессами потребовало интенсивного развития их алгоритмического обеспечения.

Технико-экономические требования к МП и МК (дешевизна, быстродействие, простота конструкции вычислителя) привели к востребованности целочисленных алгоритмов.

Их достоинства – ограниченный набор простых операций (сложения, вычитания, сдвиги, условные переходы) и быстродействие – были успешно продемонстрированы при их аппаратной реализации (цифровые линейные, круговые интерполяторы [1], спецвычислители, например, для навигации [2]).

Исторически разностно-итерационные алгоритмы своё применение ведут от всемирно известных алгоритмов Волдера и Меджита [3,4].

Впоследствии они были развиты в трудах многих зарубежных и отечественных учёных. Однако в основном – на применении аппаратурно, в том числе и в микроэлектронном исполнении [2,5].

Причём абсолютное большинство этих алгоритмов создавалось эвристическим путём, т.е. они имели сугубо специфическое применение на реализацию конкретной функции (экспонента, тригонометрические функции).

Каждый такой алгоритм был уникальным (для конкретной функции) и обычно заявлялся как изобретение на функциональный преобразователь. В этом плане характерен такой [6] (для $x > 0, y > 0$):

$$q_{j-1} = sign(X_{j-1} - Y_{j-1});$$

$$\begin{matrix} X_0 = x & X_j = X_{j-1} - q_{j-1} \cdot y \cdot 2^{-j}, & X_n \to \dfrac{x^2 + y^2}{x + y}; \\ Y_o = y & Y_j = Y_{j-1} + q_{j-1} \cdot y \cdot 2^{-j}, & Y_n \to X_n, \end{matrix} \qquad (1)$$

где $j = 1,2,\ldots$ – номер итерации,

$(n+2)$ – разрядность аргументов, включая знак.

Алгоритм (1), как и ему подобные, является неаналитическим и служит для вычисления одной единственной функции – $f(x,y) = (x^2 + x^2)/(x + y)$.

Обращаем внимание на то, что РИА не содержат умножения и деления, хотя и реализуют аналитические выражения с присутствием последних.

В этом существенное преимущество при их реализации на простейших микропроцессорах (без команд умножения и деления), на скоростных микропроцессорах с сокращённым набором команд (RISC – архитектура).

Кроме простоты требуемого вычислителя, РИА обладают преимуществами по точности вычисления (при данной разрядности вычислителя).

Действительно, при умножении результат получается двойной длины, который в обычных алгоритмах усекается до одной длины. А при делении аналогичным образом теряется остаток.

И то, и другое существенно снижает точность всей цепочки вычисления. Чтобы этого не допустить, программисты переходят на удвоенную разрядную сетку либо на алгоритмы с двойной точностью представления величин.

В первом случае это более сложный микропроцессор (вычислитель), во втором случае более медленно работающий алгоритм.

Всё это сдерживало широкое и эффективное применение микропроцессорных устройств автоматизации (т.е. локальная автоматики). Выход из этого положения мы видим в использовании РИА как наиболее подходящих для микропроцессорной реализации.

Однако у разработчиков алгоритмического обеспечения микропроцессорных систем возникали трудности с анализом известных и синтезом новых разностно-итерационных алгоритмов.

Дело значительно упрощается, если манипулировать не с самими РИА, а с их математическими моделями. В этом случае алгоритмист имеет дело не с рекуррентными выражениями, а с обычными алгебраическими. Обычными исключениями и подстановками внутри системы таких выражений легко определяется:

- условия сходимости РИА;
- пределы, к которым стремятся интегрируемые величины;

- возможные модификации известных РИА для получения новых, требуемых по условиям автоматизации.

Основные положения анализа и синтеза РИА приведены в [7-9].

Приведём краткую классификацию РИА. Они бывают двух типов:

1) с аддитивным разложением частного двух величин, например $w$ и $u$:

$$
\begin{aligned}
&q_{j-1} = sign(W_{j-1}),\\
&W_0 = w \quad W_j = W_{j-1} - q_{j-1} \cdot u \cdot 2^{-j}, \qquad W_n \to 0;\\
&Z_o = z \quad Z_j = Z_{j-1} + q_{j-1} \cdot v \cdot 2^{-j}, \quad Z_n \to z + \frac{w}{u} \cdot v.
\end{aligned} \tag{2}
$$

На первом этапе частное от деления $w/u$ представляется в виде суммы

$$Q = \sum_{j=1}^{n} q_{j-1} \cdot 2^{-j}, \tag{3}$$

где $q_j \in \{-1;+1\}$ – индикаторы $j$-й итерации.

На втором этапе происходит функциональное преобразование (например, умножение) $Q$ на другие константы или итерируемые величины. Однако, это, допустим, умножение выполняется на каждый из членов суммы (3) и что самое главное, – последовательно во времени, по мере поступления очередных членов $q_{j-1}$ на первом этапе. Из-за особого характера такого умножения эта процедура носит название «квазиумножение»;

2) второй структурой РИА является мультипликативное разложение $w/u$

$$
\begin{aligned}
&q_{j-1} = signY_{j-1}\\
&Y_0 = |w| - |u|, \quad Y_j = Y_{j-1} - q_{j-1} \cdot 2^{-j+1} \cdot X_{j-1}, \quad Y_n \to 0;\\
&X_o = |u|, \quad X_j = X_{j-1} + q_{j-1} \cdot 2^{-j} \cdot X_{j-1}, \quad X_n \to |w|,
\end{aligned} \tag{4}
$$

где $j = 1,1,2,2,\ldots,n,n$ – номер двойных итераций.

Представление $Q$ стремится к величине $w/u$

$$Q = \prod_{j=1}^{n} (1 + q^*_{j-1} \cdot 2^{-j+1}) * (1 + q_{j-1} \cdot 2^{-j+1}), \tag{5}$$

где $q^*_{j-1}$ – индикатор после первой половины $(j-1)$-й итерации.

РИА второго типа (4) используются для некоторых преобразований (возведение дроби в натуральную степень, вычисление логарифма от дроби, вычисление обратной величины и др.).

Большие потенциальные возможности расширения типов функциональных зависимостей предоставляют предложенные нами математические модели РИА [7].

Покажем это на примере РИА типа (1). С использованием его матмодели нами он был модифицирован так (для $x,y,u,w$ – любого знака):

$$q_{j-1} = sign(X_{j-1} - Y_{j-1}) \cdot sign(u+w);$$
$$Y_0 = y, \quad Y_j = Y_{j-1} + q_{j-1} \cdot 2^{-j+1} \cdot u, \qquad Y_n \to X_n; \tag{6}$$
$$X_o = x, \quad X_j = X_{j-1} - q_{j-1} \cdot 2^{-j+1} \cdot w, \quad X_n = \frac{x \cdot u + y \cdot w}{u+w}.$$

Таким образом, получена возможность вычисления $z(x,y,w,u) = \dfrac{(x \cdot u + y \cdot w)}{(u+w)}$, т.е средневзвешенного двух величин.

Дальнейшее расширение области применимости РИА (6) состоит в представлении $x,y,u,w$ как функций нового аргумента $t$:

$$x = k_x \cdot t + m_x, \quad w = k_w \cdot t + m_w,$$
$$y = k_y \cdot t + m_y, \quad u = k_u \cdot t + m_u. \tag{7}$$

Тогда $Y_n = X_n = \dfrac{At^2 + Bt + C}{Dt + E}$,

где

$$A = k_x k_u + k_y k_w,$$
$$B = k_x m_u + k_u m_x + k_w m_y + k_y m_w,$$
$$C = m_x m_u + m_y m_w, \tag{8}$$
$$D = k_u + k_w,$$
$$E = m_u + m_w .$$

Для того, чтобы упростить вычисление (7) для заданного $t$ выберем $k_i$ ($i \in \{x,y,w,u\}$) равными $\pm 2^{-z}$ или 0 (где $z$ – целое неотрицательное число).

Если же требуется решить обратную задачу: назначить такие $k_i$ и $m_j$, чтобы $Y_n = X_n = F(t)$ на заданном промежутке $t \in [\alpha;\beta]$, то находим аппроксимацию $F(t)$ в виде (8), т.е. набор коэффициентов $A,B,C,D,E$.

По ним, решая систему из 5 уравнений с 8 неизвестными, находим набор $k_i$ и $m_j$, наиболее точно удовлетворяющих системе уравнений

$$k_x k_u + k_y k_w = A,$$
$$k_x m_u + k_u m_x + k_w m_y + k_y m_w = B,$$
$$m_x m_u + m_y m_w = C, \tag{9}$$
$$k_u + k_w = D,$$
$$m_u + m_w = E .$$

Способы поиска рационального представления (8) и методика решения системы (9) нами разработаны. Все это сведено в диалоговую систему – автоматизированное рабочее место (АРМ) «Булат - 2» на базе персонального компьютера [8]. Система выдаёт блок-схему алгоритма и его программу.

Преимущество этого АРМа состоит в том, что он позволяет все функциональные зависимости реализовать по одной и той же подпрограмме, реализующей РИА (6). Сама исходная функция может быть задана как аналитически, так и таблично.

Ввиду краткости и целочисленности РИА (6) его быстродействие велико и не зависит от сложности аналитического или табличного задания функции.

Предложенная система автоматизированного проектирования разностно-итерационных алгоритмов функционального преобразования и обработки информации рекомендуется для микропроцессорных устройств локальной автоматики в различных областях науки и техники.

Работа выполнена в Академии маркетинга и социально-информационных технологий (ИНФИТ), г. Краснодар.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК


1. *Ратмиров В.А.* Основы программного управления станками – М.: Машиностроение, 1978. –С. 121-125.
2. *Байков В.Д., Смолов В.Б.* Специализированные процессоры: итерационные алгоритмы и структуры. – М.: Радиосвязь, 1985. –288 с.
3. *Volder J.E.* The CORDIC trigonometric computing technique// The Trans. Electronic Comp. 1959., Vol. 8. N3. p. 330-334.
4. *Meggitt J.E.* Pseudovision and Pseudomultiplication // IBM J. Res. And Develop., 1965, Vol.6, N2, p. 210 – 226.
5. *Оранский А.М.* Аппаратные методы в цифровой вычислительной технике. – Минск: Изд-во БГУ, 1977. –208с.
6. А.С. 744 595 СССР, МКИ$^2$ G06F/34. Цифровой функциональный преобразователь/ А.М. Оранский, Л.А. Рейнхерберг (СССР). – 2379674/ 12-24; Заяв. 7.7.76 Опубл. 30.6.80; Бюл. 24. –2с.
7. *Анишин Н. С. и др.* Математические модели разностно-итерационных алгоритмов //Новые информационные технологии: Сборник трудов 7-й Всероссийской н/т конференции. – М.: МГАПИ 2004. –С.3-5.
8. *Анишин Н.С. и др.* АРМ проектировщика алгоритма для микропроцессорных систем. Материалы Всероссийской н/т конференции «Компьютерные технологии в инженерной и управленческой деятельности». –Таганрог: Изд-во ТРТУ, 2000. –С. 442-445.


**Б.А. Державец**

## УСТАНОВКА БИНАРНОЙ ВЕРСИИ XEN 3.1 В СРЕДЕ DEBIAN ETCH 4.0 (x86_64)

Статья отвечает на замечание в известном online издании HowTo-Forge [3] издаваемым Falko Timme.

В руководстве “The Perfect Xen 3.1.0 Setup For Debian Etch (i386)” Falko утверждает (язык оригинала сохранен) :

A note on x86_64 systems: I tried to install Xen 3.1.0 on Debian Etch AMD64 as well,but regardless of the method (Xen source install vs. Xen x86_64 binary install),the Xen kernel didn’t boot (no error messages, but the boot process was incredibly slow and never finished...).

При тестировании Debian Etch 4.0 (amd64) был установлен на машину со следующими характеристиками:

1) процессор – Core 2 Duo E6600;