ISSN 1816-0301

ИНФОРМАТИКА

ОИПИ НАН Беларуси

50 лет

1(45)

ЯНВАРЬ–МАРТ
2015

Адрес редакции:
220012, Минск,
ул. Сурганова, 6, к. 305
тел. (017) 284-26-22
e-mail: rio@newman.bas-net.by
http://uiip.bas-net.by

ОБЪЕДИНЕННЫЙ ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ
НАЦИОНАЛЬНОЙ АКАДЕМИИ НАУК БЕЛАРУСИ

ИНФОРМАТИКА

ЕЖЕКВАРТАЛЬНЫЙ НАУЧНЫЙ ЖУРНАЛ

Издается с января 2004 г.

№ 1(45) • январь-март 2015

СОДЕРЖАНИЕ

ЗАЩИТА ИНФОРМАЦИИ

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Редактор Г.Б. Гончаренко
Корректор А.А. Михайлова
Компьютерная верстка Д.С. Гавинович

Сдано в набор 23.01.2015. Подписано в печать 02.03.2015.
Формат 60×84 1/8. Бумага офсетная. Гарнитура Таймс.
Усл. печ. л. 14,0. Уч.-изд. л. 13,7. Тираж 100 экз. Заказ 1.

Государственное научное учреждение «Объединенный институт проблем информатики Национальной академии наук Беларуси».
Свидетельство о государственной регистрации издателя, изготовителя, распространителя печатных изданий № 1/274 от 04.04.2014.
ЛП № 02330/444 от 18.12.13.
Ул. Сурганова, 6, 220012, Минск.

THE UNITED INSTITUTE OF INFORMATICS PROBLEMS
OF THE NATIONAL ACADEMY OF SCIENCES OF BELARUS

INFORMATICS

PUBLISHED QUATERLY

Issued since 2004

№ 1(45) • January-March 2015

CONTENTS

INFORMATION SECURITY

APPLIED INFORMATION TECHNOLOGIES

ОБРАБОТКА СИГНАЛОВ, ИЗОБРАЖЕНИЙ И РЕЧИ

УДК 004.942

В.М. Артемьев, А.О. Наумов, Л.Л. Кохан

ОПТИМАЛЬНАЯ ЛИНЕЙНАЯ СОВМЕЩЕННАЯ ФИЛЬТРАЦИЯ СЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ НА ОСНОВЕ РЕКУРРЕНТНОГО МЕТОДА НАИМЕНЬШИХ КВАДРАТОВ

Решается задача синтеза линейного совмещенного (комплексного) фильтра по критерию минимума текущих потерь на основе рекуррентного метода наименьших квадратов. При таком подходе не требуется знаний априорных статистических характеристик воздействий, что является преимуществом по сравнению с фильтром Калмана. Дается сравнительная оценка точности этих фильтров по величинам дисперсий ошибок фильтрации.

Введение

Одним из способов повышения точности измерений является совмещение результатов измерений одних и тех же параметров от совокупности датчиков, которые могут быть построены на различных физических принципах. Как правило, измерения осуществляются в условиях помех, что приводит к необходимости фильтрации результатов. Такая процедура носит название комплексной фильтрации. Для получения наиболее точных данных целесообразно решать задачу оптимальной комплексной фильтрации, важную для ряда технических систем, например навигации [1], управления подвижными объектами [2] и др. Комплексирование может осуществляться посредством двух схем обработки: централизованной и децентрализованной. В первом случае производится объединение измеренных данных, а затем осуществляется их фильтрация, во втором первоначально осуществляется фильтрация каждой из составляющих измерений, а затем результаты объединяются. Любая из этих схем имеет свои преимущества и недостатки [3]. Решению задач комплексирования посвящен ряд публикаций. Одной из первых считается монография [4], в которой решены задачи комплексирования в системах управления летательными аппаратами. В ней используются спектральные методы анализа и синтеза линейных фильтров, пригодные для комплексирования стационарных случайных процессов, и они требуют априорного знания спектральных характеристик воздействий. В дальнейшем были разработаны методы, основанные на использовании марковской теории оценивания [5], обладающие большей общностью, которые позволяют решать стационарные и нестационарные, линейные и нелинейные задачи комплексирования. В то же время для своей реализации они также требуют знания априорных статистических характеристик воздействий, что в ряде случаев недоступно исследователю. Поэтому сохраняется необходимость разработки методов синтеза комплексных фильтров в условиях априорной статистической неопределенности характеристик воздействий.

Одним из подходов к решению данной задачи может быть использование рекуррентного метода наименьших квадратов (РМНК) [3], требующего эмпирических представлений об отношениях сигнала к шумам в каналах измерений. В настоящей работе предлагается методика синтеза структуры и параметров комплексного фильтра на основе РМНК для воздействий в виде случайных последовательностей при их неизвестных статистических характеристиках.

Предположим, что имеются N датчиков, измеряющих один и тот же параметр x_k в дискретные моменты времени $k = 0, 1, 2, \ldots$. Скорость его изменения определяется величиной первой разности $\vartheta_{1k} = x_k - x_{k-1}$, ускорение – величиной второй разности $\vartheta_{2k} = \vartheta_{1k} - \vartheta_{1k-1}$, третья разность задается выражением $\vartheta_{3k} = \vartheta_{2k} - \vartheta_{2k-1}$. Совокупность входных сигналов датчиков можно представить в виде вектора $x_k \cdot \mathbf{1}$, где $\mathbf{1} = [1, 1, \ldots, 1]^T$ есть единичный вектор раз-

мерности N. Характеристики совокупности линейных безынерционных датчиков и связи между ними задаются матрицей $\boldsymbol{H}$ размерности $N \times N$, элементы которой обозначаются символом h_{ij} ($i = \overline{1,N}$, $j = \overline{1,N}$). Диагональные элементы матрицы $h_{ii} = h_i$ являются коэффициентами чувствительности датчиков, а остальные определяют связи между ними. Ошибки измерений полагаются аддитивным вектором шумов $\boldsymbol{v}_k = \left[v_{1k}, v_{2k}, \ldots, v_{Nk}\right]^T$, в результате чего вектор измерений $\boldsymbol{z}_k = \left[z_{1k}, z_{2k}, \ldots, z_{Nk}\right]^T$ имеет вид

$$\boldsymbol{z}_k = x_k \cdot \boldsymbol{H} \cdot \boldsymbol{1} + \boldsymbol{v}_k. \tag{1}$$

Результаты измерений обрабатываются комплексным фильтром, выходом которого будет оценка $\hat{x}_k$ значений входной последовательности. Рассмотрим методику нахождения уравнений оптимального комплексного фильтра наименьших квадратов (ФНК) на основе РМНК.

1. Уравнение оптимального комплексного фильтра

Синтез линейного ФНК основан на введении критерия оптимальности в виде квадратичного функционала текущих потерь и его минимизации. В состав функционала входят квадратичная невязка решения $\left(\boldsymbol{z}_k - \hat{x}_k \boldsymbol{H} \cdot \boldsymbol{1}\right)^T \left(\boldsymbol{z}_k - \hat{x}_k \boldsymbol{H} \cdot \boldsymbol{1}\right)$ и сглаживающая часть, содержащая квадратичные составляющие оценки и ее разностей, которые обеспечивают единственность, стабильность и сглаживание решения. Число учитываемых разностей определяет порядок фильтра. Вариант функционала для синтеза фильтра первого порядка имеет следующий вид:

$$\begin{gathered} Q_k\left(\hat{x}_k\right) = \left(1-\alpha\right)\left(\boldsymbol{z}_k - \hat{x}_k \boldsymbol{H} \cdot \boldsymbol{1}\right)^T \left(\boldsymbol{z}_k - \hat{x}_k \boldsymbol{H} \cdot \boldsymbol{1}\right) + \alpha\left[\hat{x}_k^2 + \hat{\vartheta}_{1k}^2\right]; \\ \hat{\vartheta}_{1k} = \hat{x}_k - \hat{x}_{k-1}; \;\; 0 \le \alpha \le 1. \end{gathered} \tag{2}$$

Коэффициент регуляризации α задает степень сглаживания и имеет эмпирический характер [6]. Для нахождения уравнения оптимальных оценок $\hat{x}_k$ можно использовать необходимое условие оптимальности $\dfrac{\partial Q_k\left(\hat{x}_k\right)}{\partial \hat{x}_k} = 0$. Дифференцируя выражение (2) по скаляру $\hat{x}_k$, получаем соотношение

$$\frac{\partial Q_k\left(\hat{x}_k\right)}{\partial \hat{x}_k} = \left(1-\alpha\right)\left(2 \cdot \hat{x}_k \boldsymbol{1}^T \boldsymbol{H}^T \boldsymbol{H} \cdot \boldsymbol{1} - 2 \cdot \boldsymbol{1}^T \boldsymbol{H}^T \boldsymbol{z}_k\right) + 2\alpha\left(2\hat{x}_k - \hat{x}_{k-1}\right) = 0.$$

Его решение приводит к равенству

$$\hat{x}_k = \frac{\alpha + \left(1-\alpha\right)\boldsymbol{1}^T \boldsymbol{H}^T \boldsymbol{H} \cdot \boldsymbol{1}}{2\alpha + \left(1-\alpha\right)\boldsymbol{1}^T \boldsymbol{H}^T \boldsymbol{H} \cdot \boldsymbol{1}} \hat{x}_{k-1} + \frac{\left(1-\alpha\right)\boldsymbol{1}^T \boldsymbol{H}^T \boldsymbol{H} \cdot \boldsymbol{1}}{2\alpha + \left(1-\alpha\right)\boldsymbol{1}^T \boldsymbol{H}^T \boldsymbol{H} \cdot \boldsymbol{1}} \left(\frac{\boldsymbol{1}^T \boldsymbol{H}^T}{\boldsymbol{1}^T \boldsymbol{H}^T \boldsymbol{H} \cdot \boldsymbol{1}} \boldsymbol{z}_k - \hat{x}_{k-1} \right). \tag{3}$$

Скалярная величина $\boldsymbol{1}^T \boldsymbol{H}^T \boldsymbol{H} \cdot \boldsymbol{1} = h_0^2$ выражается через параметры матрицы $\boldsymbol{H}$ следующим образом:

$$h_0^2 = \sum_{i=1}^{N} \left(\sum_{j=1}^{N} h_{ij} \right)^2. \tag{4}$$

Посредством этой скалярной величины учитываются структура и параметры матрицы датчиков.

Первое слагаемое в правой части равенства (3) является экстраполяцией оценки $\hat{x}_{k-1}$ на следующий период фильтрации, поэтому сомножитель при этой величине можно определить как коэффициент экстраполяции

$$K_1 = \frac{\alpha + (1-\alpha)h_0^2}{2\alpha + (1-\alpha)h_0^2}. \quad (5)$$

Второе слагаемое учитывает результаты текущих измерений, и множитель перед круглой скобкой определяется как коэффициент усиления фильтра K_2:

$$K_2 = \frac{(1-\alpha)h_0^2}{2\alpha + (1-\alpha)h_0^2}. \quad (6)$$

Первое слагаемое в круглых скобках $(\mathbf{1}^T \boldsymbol{H}^T / h_0^2)\boldsymbol{z}_k$ является скалярным входом z_{0k} комплексного фильтра. С учетом выражения (1)

$$z_{0k} = \frac{\mathbf{1}^T \boldsymbol{H}^T}{h_0^2}\boldsymbol{z}_k = x_k + v_{0k}, \quad (7)$$

где $v_{0k} = \frac{\mathbf{1}^T \boldsymbol{H}^T}{h_0^2}\boldsymbol{v}_k$. В итоге уравнение оптимального линейного комплексного ФНК принимает вид

$$\hat{x}_k = K_1 \hat{x}_{k-1} + K_2 (z_{0k} - \hat{x}_{k-1}). \quad (8)$$

Изображенная на рис. 1 структура соответствует схеме централизованного комплексного фильтра, поскольку первоначально происходит объединение вектора измерений $\boldsymbol{z}_k$ в скалярный входной сигнал z_{0k}, состоящий из суммы значений измеряемого параметра x_k и комплексного шума измерений v_{0k}. Символом $k-1$ обозначена операция задержки данных на один период измерений. Синтезированный фильтр имеет первый порядок, поскольку в сглаживающей части функционала (2) использована лишь первая разность ϑ_{1k}. Для синтеза фильтров более высокого порядка следует дополнительно использовать более высокие разности, однако при этом методика синтеза остается прежней. Повышение порядка фильтра приводит к уменьшению ошибок фильтрации измеряемого параметра, но увеличивает уровень ошибок за счет шумов измерений.

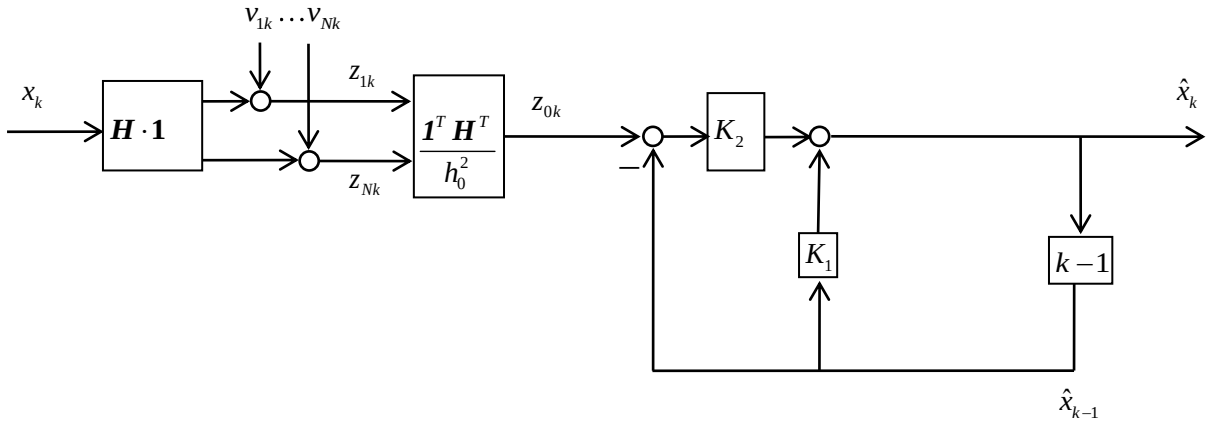


Рис. 1. Структурная схема комплексного фильтра

По своей структуре ФНК на рис. 1 близок фильтру Калмана (ФК) первого порядка с постоянными коэффициентами, однако их нахождение по формулам (4)–(6) не зависит от априорных статистических характеристик воздействий, а определяется лишь значением h_0^2 и коэффициентом α. Выбор α осуществляется из эмпирических представлений о величинах отношений сигнала к шумам в каналах фильтра. При большом отношении сигнала к шумам предпочтение

следует отдавать результатам измерений и выбирать величину α малой. При малом отношении сигнала к шумам роль сглаживания решения возрастает и значение α следует увеличивать. При полной априорной неопределенности целесообразно полагать $\alpha = 0{,}5$. Более точные значения α находятся по результатам моделирования.

2. Точность фильтрации

Отсутствие учета априорной статистической информации об измеряемом параметре и шумах измерений снижает точность фильтрации по сравнению с ФК, полностью учитывающим данную информацию. С этой точки зрения результаты оценок ФК будут для ФНК нижней границей ошибок фильтрации. Представляет интерес оценка степени ухудшения точности фильтрации за счет неучета априорной статистики. Естественно, что сравнительная оценка точности должна проводиться при одинаковых входных воздействиях.

Оптимальному ФК первого порядка с постоянными коэффициентами соответствует модель измеряемого параметра x_k в виде стохастического конечно-разностного уравнения первого порядка следующего вида [3]:

$$x_k = ax_{k-1} + \xi_k,\ k = 0, 1, 2, \ldots, \tag{9}$$

где коэффициент $0 < a < 1$ и ξ_k – дискретный белый шум с нулевым математическим ожиданием $\langle \xi_k \rangle = 0$ и дисперсией $\langle \xi_k^2 \rangle = \sigma_\xi^2$. Эта модель может быть использована для сравнительной оценки точности фильтров.

Входной процесс фильтра z_{0k} (см. рис. 1) является результатом измерений x_k с ошибками v_{0k}, зависящими от вектора шумов $\boldsymbol{v}_k$ (7). В отличие от общей формулировки задачи в рассматриваемом примере полагаем, что z_{0k} состоит из N стационарных статистически независимых белых шумов с нулевыми математическими ожиданиями и дисперсиями σ_{iv}^2, $i = \overline{1, N}$. При этом ковариационная матрица шумов $\boldsymbol{V} = \langle \boldsymbol{v}_k \boldsymbol{v}_k^T \rangle$ будет диагональной:

$$\boldsymbol{V} = \begin{pmatrix} \sigma_{1v}^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_{Nv}^2 \end{pmatrix}.$$

Скалярный процесс v_{0k} также будет белым шумом с нулевым математическим ожиданием и постоянной дисперсией $\sigma_{0v}^2 = \langle v_{0k}^2 \rangle$, которая находится из выражения

$$\sigma_{0v}^2 = \left\langle \frac{\mathbf{1}^T \boldsymbol{H}^T}{h_0^2} \boldsymbol{V} \frac{\boldsymbol{H}\mathbf{1}}{h_0^2} \right\rangle = \sum_{i=1}^{N} \delta_i \sigma_{iv}^2,\ \delta_i = \sum_{j=1}^{N} h_{ij} \Big/ h_0^4. \tag{10}$$

Шумы измерений $\boldsymbol{v}_k$ и дискретный белый шум ξ_k модели измеряемого параметра полагаются статистически независимыми между собой.

Определим дисперсию ошибок ФНК $e_k = x_k - \hat{x}_k$ при описанных выше моделях воздействий. Подставляя в формулу ошибок значения x_k и $\hat{x}_k$ из (8) и (9), получаем уравнение

$$e_k = ax_{k-1} + \xi_k - K_1 \hat{x}_{k-1} - K_2 \left(z_{0k} - \hat{x}_{k-1} \right).$$

Поскольку $\hat{x}_{k-1} = x_{k-1} - e_{k-1}$, $z_{0k} = x_k + v_{0k}$, а $x_k = ax_{k-1} + \xi_k$, то в итоге это выражение принимает следующую форму:

$$e_k = bx_{k-1} + ce_{k-1} + w_k, \tag{11}$$

где использованы обозначения

$$b = a(1-K_2) - c\,,\ c = K_1 - K_2\,. \tag{12}$$

Случайная последовательность $w_k = (1-K_2)\xi_k - K_2 v_{0k}$ является белым шумом с нулевым математическим ожиданием и дисперсией σ_w^2, задаваемой формулой

$$\sigma_w^2 = (1-K_2)^2\,\sigma_\xi^2 + K_2^2\sigma_{0v}^2\,. \tag{13}$$

Уравнения (9) и (11) образуют замкнутую систему стохастических конечно-разностных уравнений второго порядка. При нулевых начальных условиях $x_0 = 0$, $e_0 = 0$ математические ожидания $\langle x_k \rangle = 0$, $\langle e_k \rangle = 0$, а дисперсии $\sigma_{x_k}^2 = \langle x_k^2 \rangle$ и $\sigma_{e_k}^2 = \langle e_k^2 \rangle$. Чтобы найти их выражения, можно воспользоваться методом интегрирования вероятностных моментов процессов x_k и e_k, который описан, например, в [7].

Для получения трех уравнений вторых вероятностных моментов (двух дисперсий и ковариации) возведем в квадрат левые и правые части уравнений (9) и (11), а также перемножим эти уравнения по частям. Полученные результаты усредним и в итоге получим следующие уравнения:

$$\langle x_k^2 \rangle = a^2 \langle x_{k-1}^2 \rangle + 2a\langle x_{k-1}\xi_k \rangle + \langle \xi_k^2 \rangle;$$

$$\langle e_k^2 \rangle = b^2 \langle x_{k-1}^2 \rangle + 2bc\langle x_{k-1}e_{k-1} \rangle + c^2 \langle e_{k-1}^2 \rangle + 2c\langle e_{k-1}w_k \rangle + \langle w_k^2 \rangle;$$

$$\langle x_k e_k \rangle = ab\langle x_{k-1}^2 \rangle + ac\langle x_{k-1}e_{k-1} \rangle + a\langle x_{k-1}w_k \rangle + b\langle x_{k-1}\xi_k \rangle + c\langle e_{k-1}\xi_k \rangle + \langle \xi_k w_k \rangle.$$

В этих уравнениях $\langle x_k^2 \rangle = \sigma_{x_k}^2$, $\langle x_{k-1}^2 \rangle = \sigma_{x_{k-1}}^2$, $\langle e_k^2 \rangle = \sigma_{e_k}^2$, $\langle e_{k-1}^2 \rangle = \sigma_{e_{k-1}}^2$, $\langle \xi_k^2 \rangle = \sigma_\xi^2$, $\langle w_k^2 \rangle = \sigma_w^2$, $\langle x_k e_k \rangle = \sigma_{xe_k}^2$, $\langle x_{k-1}e_{k-1} \rangle = \sigma_{xe_{k-1}}^2$. Последние два выражения обозначают ковариацию процессов x_k и e_k в различные моменты времени. Среднее значение произведения белых шумов $\langle \xi_k w_k \rangle = (1-K_2)\langle \xi_k^2 \rangle - K_2\langle \xi_k v_{0k} \rangle$. Поскольку процессы ξ_k и v_{0k} статистически независимы, то $\langle \xi_k v_{0k} \rangle = 0$ и $\langle \xi_k w_k \rangle = (1-K_2)\sigma_\xi^2$. Ковариации $\langle x_{k-1}\xi_k \rangle = \langle x_{k-1}w_k \rangle = \langle e_{k-1}w_k \rangle = \langle e_{k-1}\xi_k \rangle = 0$, так как первые сомножители в момент $(k-1)$ статистически независимы от белых шумов в момент времени k. В итоге уравнения вторых вероятностных моментов образуют следующую систему регулярных линейных конечно-разностных уравнений:

$$\sigma_{x_k}^2 = a^2\sigma_{x_{k-1}}^2 + \sigma_\xi^2;$$

$$\sigma_{e_k}^2 = b^2\sigma_{x_{k-1}}^2 + 2bc\sigma_{xe_{k-1}}^2 + c^2\sigma_{e_{k-1}}^2 + \sigma_w^2;$$

$$\sigma_{xe_k}^2 = ab\sigma_{x_{k-1}}^2 + ac\sigma_{xe_{k-1}}^2 + (1-K_2)\sigma_\xi^2.$$

В установившемся режиме дисперсии остаются постоянными величинами, равными $\sigma_{x_k}^2 = \sigma_x^2$, $\sigma_{e_k}^2 = \sigma_e^2$, $\sigma_{xe_k}^2 = \sigma_{xe}^2$, и уравнения моментов образуют систему линейных алгебраических уравнений

$$0 = (a^2 - 1)\sigma_x^2 + \sigma_\xi^2; \tag{14}$$

$$0 = b^2\sigma_x^2 + 2bc\sigma_{xe}^2 + (c^2 - 1)\sigma_e^2 + \sigma_w^2; \tag{15}$$

$$0 = ab\sigma_x^2 + (ac - 1)\sigma_{xe}^2 + (1-K_2)\sigma_\xi^2. \tag{16}$$

Решение уравнения (14) приводит к формуле связи между дисперсиями σ_ξ^2 и σ_x^2 в виде

$$\sigma_\xi^2 = \sigma_x^2\left(1-a^2\right). \tag{17}$$

В [3] показано, что длительность корреляции τ_x процесса (9) связана с коэффициентом a следующим соотношением:

$$a = \frac{\tau_x}{1+\tau_x}. \tag{18}$$

Решение уравнений (15), (16) с учетом (13) и (17) позволяет найти выражение для дисперсии ошибок в установившемся режиме:

$$\sigma_e^2 = A_{\text{ФНК}}\sigma_x^2 + B_{\text{ФНК}}\sigma_{0v}^2, \tag{19}$$

где

$$A_{\text{ФНК}} = \frac{b^2\left(1+ac\right)+\left(1-K_2\right)\left(1-a^2\right)\left[2bc+\left(1-K_2\right)\left(1-ac\right)\right]}{\left(1-c^2\right)\left(1-ac\right)};\ B_{\text{ФНК}} = \frac{K_2^2}{\left(1-c^2\right)}. \tag{20}$$

Первое слагаемое (19) определяет дисперсию ошибок фильтрации параметра x_k, а второе – составляющую дисперсии за счет наличия шумов измерений $\boldsymbol{v}_k$. Коэффициенты a, b и c задаются формулами (12) и (18). В результате коэффициент $A_{\text{ФНК}}$ зависит от K_1, K_2 и τ_x, а коэффициент $B_{\text{ФНК}}$ – только от K_1 и K_2.

Введем относительную величину дисперсии ФНК $\varepsilon_{\text{ФНК}} = \sigma_e^2/\sigma_x^2$ и значения отношений сигнала к шумам в каналах датчиков $q_i = \sigma_x^2/\sigma_{iv}^2$. Тогда величина отношения сигнала к шумам q_0 на входе фильтра в соответствии с формулой (10)

$$q_0 = \sigma_x^2/\sigma_{0v}^2 = \left(\sum_{i=1}^{N}\delta_i q_i^{-1}\right)^{-1}. \tag{21}$$

В результате относительная величина дисперсии ошибок фильтрации $\varepsilon_{\text{ФНК}}$ определяется соотношением

$$\varepsilon_{\text{ФНК}} = A_{\text{ФНК}} + B_{\text{ФНК}}q_0^{-1}. \tag{22}$$

ФК первого порядка для моделей воздействия (9) и измерений (7) исследован и описан в литературе достаточно подробно (см., например, [3]). Выражение для относительной величины дисперсии ошибок фильтрации $\varepsilon_{\text{ФК}}$ в установившемся режиме выглядит аналогично (19):

$$\varepsilon_{\text{ФК}} = A_{\text{ФК}} + B_{\text{ФК}}q_0^{-1}, \tag{23}$$

где

$$A_{\text{ФК}} = \frac{\left(1-K_{\text{ФК}}\right)^2}{1-a^2\left(1-K_{\text{ФК}}\right)^2};\quad B_{\text{ФК}} = \frac{K_{\text{ФК}}^2}{1-a^2\left(1-K_{\text{ФК}}\right)^2}.$$

В этих выражениях оптимальный коэффициент ФК в установившемся режиме определяется по формуле

$$K_{\text{ФК}} = \frac{q_0}{2a}\left[\sqrt{\left(1-a^2\right)^2\left(1+q_0^{-1}\right)^2+4a^2\left(1-a^2\right)q_0^{-1}} - \left(1-a^2\right)\left(1+q_0^{-1}\right)\right]. \tag{24}$$

После подстановки (24) в (23) можно в явном виде получить выражение для относительной величины дисперсии ошибок ФК:

$$\varepsilon_{\text{ФК}} = \frac{\left(1-a^2\right)}{2a^2}\left[\sqrt{\left(1-q_0^{-1}\right)^2 + 4\frac{a^2}{\left(1-a^2\right)}q_0^{-1}} - \left(1-q_0^{-1}\right)\right]. \qquad (25)$$

Сопоставление значений $\varepsilon_{\text{ФНК}}$ и $\varepsilon_{\text{ФК}}$ при одинаковых характеристиках воздействий позволяет дать сравнительную оценку точности фильтрации.

3. Пример сравнительной оценки точности ФНК и ФК

Рассмотрим случай, когда комплексный фильтр содержит N датчиков с одинаковыми коэффициентами чувствительности $h_i = 1$, $i = \overline{1, N}$, и независимыми каналами измерений, т. е. $h_{ij} = 0$, $i \neq j$. Полагаем, что в каждом из каналов отношения сигналов к шумам одинаковые, $q_i = q$. Тогда в соответствии с формулами (4), (10) и (21) имеем $h_0^2 = N$, $\delta_i = N^{-2}$ и $q_0 = Nq$.

На рис. 2 для величины $q = 2{,}5$ приведены графики зависимостей $\varepsilon_{\text{ФНК}}$ и $\varepsilon_{\text{ФК}}$ от длительности корреляции τ_x измеряемого параметра при $N = 1,\ 2,\ 3,\ 4$. Для ФНК выбраны значения коэффициентов $\alpha = 0{,}25;\ 0{,}5;\ 0{,}75$. График $\varepsilon_{ФК}$ отображен пунктирной линией и определяет потенциальную точность фильтрации.

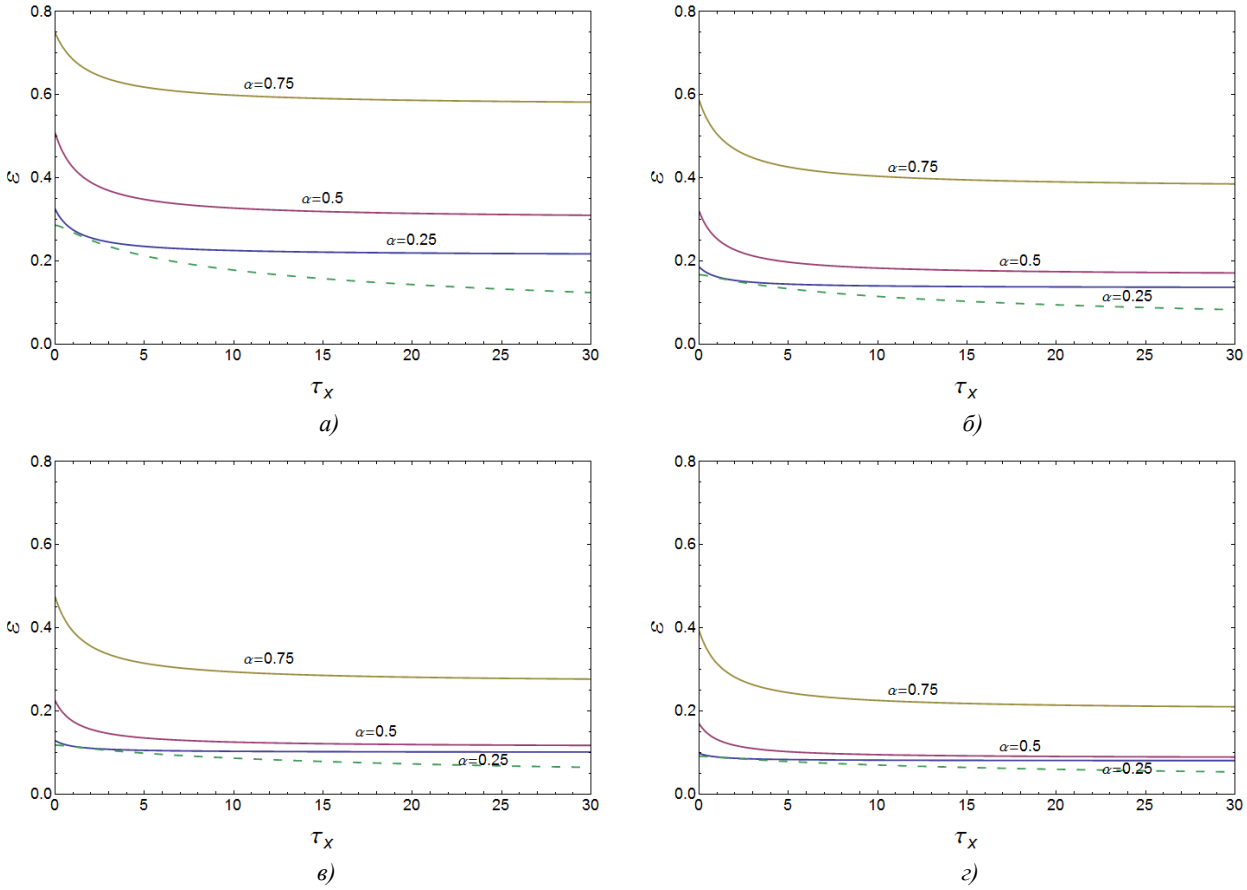


Рис. 2. Зависимость $\varepsilon_{ФНК}$ и $\varepsilon_{ФК}$ от длительности корреляции измеряемого параметра τ_x:
а) при числе каналов $N = 1$; *б)* $N = 2$; *в)* $N = 3$; *г)* $N = 4$

Из графиков следует, что с ростом числа каналов датчиков N точность фильтрации возрастает, поскольку увеличивается отношение сигнала к шумам на входе фильтра q_0. С ростом длительности корреляции τ_x измеряемого параметра дисперсии ошибок уменьшаются вследст-

вие лучшего сглаживания результатов измерений. Различия между дисперсиями ошибок ФНК и ФК показывают степень ухудшения точности за счет неучета априорной статистической информации о характеристиках воздействий и выбора величины коэффициента α.

Влияние отношений сигналов к шумам в каналах датчиков для различных величин α отображено на рис. 3 для величин $\tau_x = 3$ и $\tau_x = 10$ при числе каналов $N = 2$.

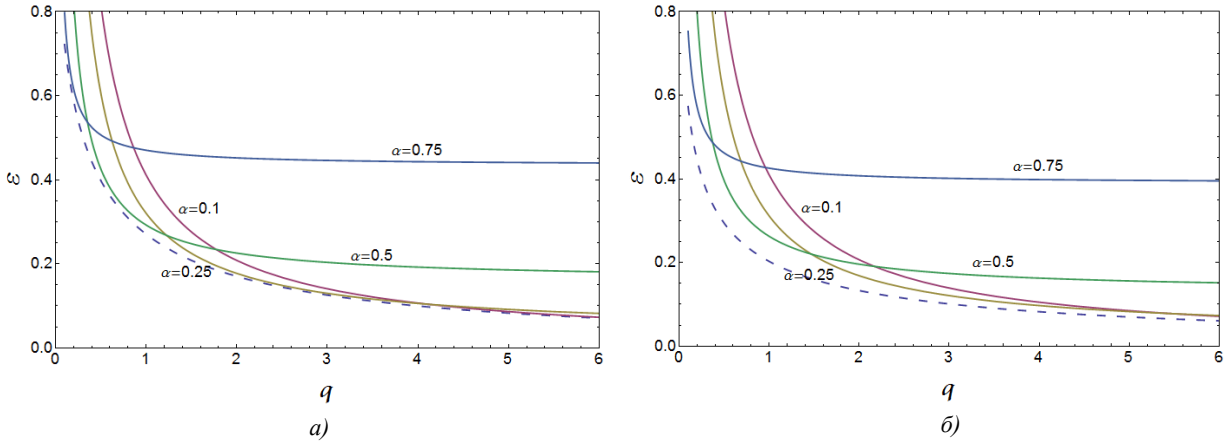


Рис. 3. Зависимость $\varepsilon_{ФНК}$ и $\varepsilon_{ФК}$ от величины отношения сигнала к шумам q в каналах: *а)* при $\tau_x = 3$; *б)* $\tau_x = 10$

При различных величинах наиболее близкими к результатам ФК оказываются данные ФНК с различными значениями α, приведенными в таблице.

Зависимость коэффициента α от q

α	$\tau_x = 3$	$\tau_x = 10$
0,75	$q < 0,28$	$q < 0,3$
0,5	$0,28 < q < 1,18$	$0,3 < q < 1,2$
0,25	$1,18 < q < 4$	$1,2 < q < 5$
0,1	$q > 4$	$q > 5$

Полученные зависимости подтверждают высказанное выше утверждение о том, что с ростом отношения сигнала к шумам следует выбирать меньшие значения α. Из рис. 3 видно, что величина $\alpha = 0,5$ может быть использована в случае неизвестных величин отношений сигнала к шумам.

Заключение

При отсутствии информации об априорных статистических характеристиках воздействий в задачах комплексной фильтрации возможно использование методики синтеза фильтров на основе РМНК. Для квадратичной функции текущих потерь это приводит к нахождению оптимального линейного комплексного фильтра в классе централизованных. На примере синтеза фильтра первого порядка показана методика решения задачи и дана оценка степени увеличения ошибок фильтрации по сравнению с фильтром Калмана. В дальнейшем путем видоизменения функции потерь целесообразно определить условия, при которых комплексный фильтр будет относиться к классу децентрализованных.

Список литературы

1. Бабич, О.А. Обработка информации в навигационных комплексах / О.А. Бабич. – М. : Машиностроение, 1991. – 512 с.

2. Алешин, Б.С. Ориентация и навигация подвижных объектов / Б.С. Алешин, К.К. Веремеенко, А.И. Черноморский. – М. : Физматлит, 2006. – 424 с.

3. Степанов, О.А. Основы теории оценивания с приложениями к задачам обработки навигационной информации. Ч.1. Введение в теорию оценивания / О.А. Степанов. – СПб. : ГНЦ РФ ЦНИИ «Электроприбор», 2009. – 496 с.

4. Бобнев, М.П. Комплексные системы радиоавтоматики : уч. пособие / М.П. Бобнев, Б.Х. Кривицкий, М.С. Ярлыков. – М. : Сов. радио, 1968. – 232 с.

5. Ярлыков, М.С. Статистическая теория радионавигации / М.С. Ярлыков. – М. : Сов. радио, 1985. – 344 с.

6. Сизиков, В.С. Математические методы обработки результатов измерений / В.С. Сизиков. – СПб. : Политехника, 2001. – 230 с.

7. Казаков, И.Е. Анализ стохастических систем в пространстве состояний / И.Е. Казаков, С.В. Мальчиков. – М. : Наука, 1983. – 384 с.

Поступила 18.11.2014

Институт прикладной физики
НАН Беларуси,
Минск, Академическая, 16
e-mail: naumov@iaph.bas-net.by

V.M. Artemiev, A.O. Naumov, L.L. Kokhan

OPTIMAL LINEAR COMBINED FILTERING OF RANDOM SEQUENCES BASED ON THE RECURSIVE LEAST SQUARES METHOD

The problem of the synthesis of linear combined filter for the criterion of minimizing current losses on the basis of the recursive least squares method is being solved. This approach does not require a priori knowledge of the statistical characteristics of impacts that is an advantage compared with the Kalman filter. A comparative evaluation of the filters' accuracy is provided using the values of variances of the filtering errors.

УДК 004.934.8'1

Я.Н. Имамвердиев, Л.В. Сухостат

МЕТОД ОБЪЕДИНЕНИЯ РЕШЕНИЙ КЛАССИФИКАТОРОВ ДЛЯ ЗАДАЧИ РАСПОЗНАВАНИЯ ДИКТОРА

Предлагается использование нечетких интегралов для объединения решений классификаторов систем распознавания диктора. В качестве набора признаков рассматриваются мгновенная частота и мгновенная амплитуда. Предлагаемый метод показывает значительно лучшие результаты по сравнению с применением единственного классификатора. Проводится сравнение предлагаемого метода с другими методами объединения решений классификаторов.

Введение

Верификация диктора – задача подтверждения личности заявленного пользователя на основе речевого образца. Некоторые из важных применений распознавания диктора включают верификацию клиентов для проведения банковских операций, доступ к банковским счетам через телефон, контроль использования кредитных карт, а также контроль безопасности в армии, военно-воздушных силах и на флоте.

Современные системы распознавания диктора по точности распознавания отстают от других биометрических систем, в том числе систем распознавания отпечатков пальцев. Между тем голос содержит богатую информацию о личности человека, и одним из путей повышения точности систем распознавания дикторов является объединение информации из разных источников в данных системах.

В биометрических системах существуют различные уровни объединения информации [1]:

– образца;

– признаков;

– значений соответствия;

– решений.

Согласно [2] для объединения значений соответствия наиболее часто применяется метод на основе преобразования данных. В этом методе для распознавания используются несколько классификаторов. Значения соответствия сначала нормализуются, и далее к ним применяются различные преобразования, например правило максимума, взвешенной суммы, логарифмического объединения и др.

Распознавание диктора включает два основных этапа: извлечение признаков и классификацию (рис. 1).

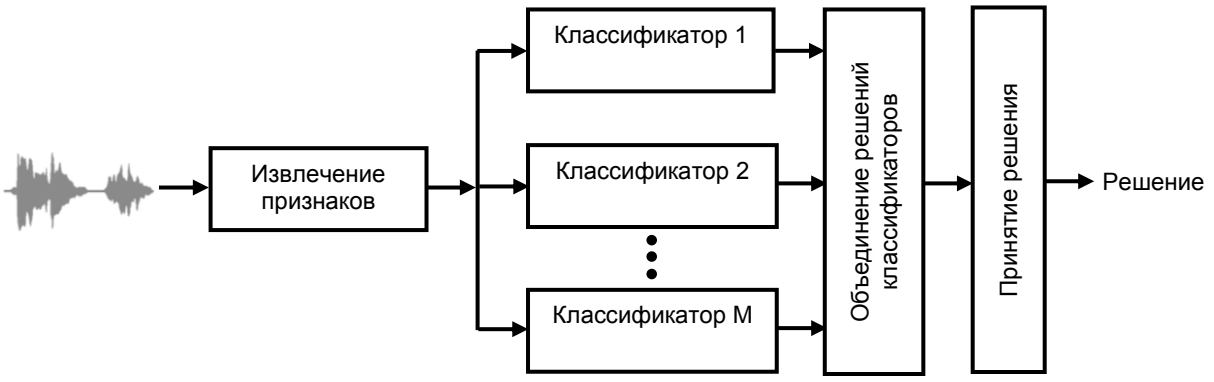


Рис. 1. Общая схема системы распознавания диктора

Цель выделения признаков заключается в преобразовании речевого сигнала к некоторому типу параметрического представления для дальнейшего анализа и обработки. Классификатор

использует эти характеристики, чтобы принять решение о принадлежности входных данных к одному из возможных классов, которые ассоциированы с отдельными дикторами из контрольной группы.

Классификатор состоит из $M+1$ моделей дикторов (M моделей дикторов из контрольной группы и одной дополнительной модели «все остальные»), которые необходимы для принятия решения и строятся на этапе обучения. На этапе тестирования тестовый вектор признаков будет связан с каждой моделью диктора с указанием степени соответствия модели.

Большинство работ по биометрическим системам сосредоточено на методах объединения информации на уровне значений соответствия из-за быстроты и эффективности.

В контексте распознавания диктора решения классификаторов, полученные на основе значений соответствия от различных моделей для каждого диктора, объединяются. Эти модели могут быть обучены с помощью различных речевых данных, признаков или методов моделирования. В конечном счете желательно, чтобы ошибки одной модели были исправлены другими и наоборот. Если все модели имеют согласованные ошибки, т. е. все они делают одну и ту же ошибку, то ни одна комбинация ее не исправит. Тем не менее, до тех пор пока существует некоторая степень несоответствия между ошибками, производительность может быть улучшена при правильном сочетании информационных данных.

Одной из отличительных особенностей нечеткого интеграла является то, что он способен представлять определенные взаимодействия между критериями. Теория нечетких интегралов была применена к распознаванию образов [3–5], обработке изображений [6–8] и объединению информации [9].

В данной работе предлагается метод объединения значений соответствия, полученных от классификаторов на основе нечетких интегралов для повышения точности распознавания диктора. В качестве речевых признаков рассматриваются мгновенные частота и амплитуда.

1. Извлечение речевых признаков для задачи распознавания диктора

Многие исследования были посвящены разработке различных схем извлечения характерных для диктора акустических признаков из речевых высказываний. Вульф в [10] среди наиболее существенных параметров выделяет частоту основного тона, спектральные признаки гласных, оценку голосового источника, продолжительность слова и др. Киннунен и соавторы [11] сделали обзор и обобщили основные особенности речи, которые были использованы для системы распознавания диктора. Наряду с классическими и ведущими признаками были приведены некоторые недавно полученные наборы параметров.

Идеальные признаки для систем распознавания диктора [10, 12] должны иметь большую вариабельность между дикторами и небольшую изменчивость внутри дикторов, быть достаточно устойчивыми к фоновому шуму и искажениям, часто использоваться в обычной речи, легко измеряться, быть стабильными во времени, не зависеть от здоровья (настроения) говорящего и быть трудно имитируемыми.

С точки зрения физической интерпретации [13] признаки, как правило, делятся на пять групп: спектральные, спектрально-временные, признаки голосового источника, просодические и признаки высокого уровня.

Среди спектральных признаков можно выделить кепстральные коэффициенты линейного предсказания (Linear Prediction Cepstral Coefficients, LPCC) [14], кепстральные коэффициенты по шкале мел (Mel-Frequency Cepstral Coefficients, MFCC) и др. [15].

MFCC-признаки являются наиболее известными и популярными спектральными признаками. Помимо кепстральных коэффициентов также рассматриваются их первые и вторые производные (дельта-признаки). В отличие от признаков высокого уровня, требующих более сложной предварительной обработки [16], их легче вычислить и получить хорошие результаты [17]. Помимо кепстральных особенностей, речь имеет и источник возбуждения, который содержит полезные свойства для распознавания диктора. Кроме того, в реальных ситуациях существуют большие различия между этапами разработки и практического применения системы распознавания диктора. Вследствие этого кепстральные признаки недостаточны, чтобы

обеспечить удовлетворительную и надежную точность распознавания. Они также не учитывают нестационарность и нелинейность человеческой речи.

В последние годы при описании и анализе свойств речи был использован подход на основе AM-FM (Amplitude-Modulation Frequency-Modulation)-моделирования [18]. Монокомпонент AM-FM-сигнала описывается уравнением

$$x(n) = A(n)\cos[\Theta_n], \tag{1}$$

где $A(n)$ обозначает мгновенную амплитуду монокомпонентного сигнала, а Θ_n – мгновенную фазу. При этом многокомпонентный сигнал сначала разлагается на монокомпоненты, каждый его компонент описывается мгновенной огибающей и мгновенной частотой. Данный подход показывает значительные улучшения показателей распознавания диктора.

2. Объединение решений классификаторов для задачи распознавания диктора

Сочетание различных источников информации было изучено в таких областях, как объединение данных [5], достижение консенсуса [19], теория принятия решений [20], комбинация решений нескольких экспертов [20] и т. д. Под комбинацией данных здесь понимается их объединение.

Выбор методов объединения данных может быть сделан в зависимости от типа информации, которая будет объединена. Например, если выходы модели есть вероятности, то могут быть применены такие методы, как линейное или логарифмическое объединение [19]. Если выходы модели на самом деле есть метки класса, то используются такие методы, как голосование [21] или ранжирование [20].

Правило максимума. Данный метод выбирает то значение соответствия, которое является наибольшим среди участвующих признаков. Математически он обозначается следующим образом:

$$P_{\max}(x) = \max(p_i(x)), \quad i = 1, \ldots, n,$$

где $p_i(x)$ – вероятностный выход i-й модели; n – число моделей.

Линейное объединение. Метод линейного объединения является широко используемым методом объединения данных, удобным благодаря своей простоте. Он оценивается как взвешенная сумма выходов для каждой модели:

$$P_{linear}(x) = \sum_{i=1}^{n} \alpha_i p_i(x),$$

где $P_{linear}(x)$ – линейное объединение; α_i – веса; $p_i(x)$ – вероятностный выход i-й модели; n – число моделей. Параметры α_i, как правило, выбираются между 0 и 1, их сумма принимается равной единице.

Линейное объединение на выходе дает распределение вероятностей, и веса α_i предоставляют грубую меру для распределения i-й модели. Тем не менее следует отметить, что распределение вероятностей на выходе сумматора, а именно $P_{linear}(x)$, может быть мультимодальным. Есть вероятность, что это усложнит процесс принятия стратегии. Метод линейного объединения был оценен по нескольким сценариям в рамках распознавания диктора. Они включают в себя сочетание VQ кодовых книг, обученных на кепстральных и дельтакепстральных признаках [22], а также другие подходы [23–26].

Логарифмическое объединение. Данный метод является альтернативой линейному объединению, но в отличие от линейного при логарифмическом объединении выходное распределение унимодально [19]. Логарифмическое объединение состоит из взвешенного произведения выходов моделей:

$$P_{linear}(x) = \prod_{i=1}^{n} p_i^{\alpha_i}(x).$$

Следует отметить, что, согласно этой формуле, если для любой модели вероятность равна нулю, то и сочетание вероятностей также будет равно нулю. Метод логарифмического объединения был оценен для верификации диктора [23, 25] и обеспечивает такую же производительность, как и линейный метод объединения.

Нечеткие интегралы. Исходя из концепции нечетких множеств, предложенных Л. Заде [27], М. Сугено ввел понятия нечеткой меры и нечеткого интеграла [28]. Нечеткая мера представляет собой набор функций с монотонностью, но не всегда аддитивностью, и нечеткий интеграл является функционалом с монотонностью, которая используется для агрегирования информации из различных источников по отношению к нечеткой мере.

Пусть Y – произвольное множество и $\tilde{\boldsymbol{B}}$ – поле Бореля из Y. Функция множества g, определенная на $\tilde{\boldsymbol{B}}$, является нечеткой мерой, если она удовлетворяет следующим трем аксиомам:

– граничные условия: $g(\varnothing)=0, \quad g(Y)=1$;

– монотонность: $g(A)\le g(B)$, если $A\subset B$ и $A, B\in\tilde{\boldsymbol{B}}$;

– непрерывность: $\lim_{i\to\infty} g(A_i)\le g(\lim_{i\to\infty} A_i)$, если $A_i\in\tilde{\boldsymbol{B}}$ и $\{A_i\}$ монотонно (возрастающая последовательность измеримых множеств).

Нечеткая мера g_λ, также предложенная М. Сугено, удовлетворяет еще одному условию, известному как правило λ ($\lambda>-1$):

$$g(A\cup B)=g(A)+g(B)+\lambda g(A)g(B),$$

где $A, B\subset Y$ и $A\cap B=\varnothing$.

Следует отметить, что, когда $\lambda=0$, нечеткая мера g_λ становится вероятностной мерой. В общем, значение константы λ может быть определено из свойств нечеткой меры g_λ следующим образом.

Пусть $Y=\{y_1, y_2, \ldots, y_m\}$. Если нечеткая плотность нечеткой меры g_λ определяется как функция g: $y_i\in Y\to[0,1]$, такая, что $g_i=g_\lambda(\{y_i\})$, $i=1, 2, \ldots, m$, то нечеткая мера g_λ конечного множества может быть получена в виде [29]

$$g_\lambda(Y)=\sum_{i=1}^{n} g_i+\lambda\sum_{i_1=1}^{m-1}\sum_{i_2=i_1+1}^{n} g_{i_1}g_{i_2}+\ldots+\lambda^{m-1}g_1g_2\ldots g_m. \qquad (2)$$

При условии, что $\lambda\ne 0$, (2) можно переписать в виде

$$g_\lambda(Y)=\frac{1}{\lambda}\left[\prod_{i=1}^{m}(1+\lambda g_i)\right].$$

Учитывая граничное условие $g(Y)=1$, постоянная λ может быть определена путем решения следующего уравнения:

$$\lambda+1=\prod_{i=1}^{n}(1+\lambda g_i). \qquad (3)$$

Пусть $(Y, \tilde{\boldsymbol{B}}, g)$ – пространство с нечеткой мерой и $f: Y\to[0,1]$ есть $\tilde{\boldsymbol{B}}$-мерная функция. Нечеткий интеграл из $A\subset Y$ от функции f по отношению к нечеткой мере g определяется как

$$\int_A f(y)\circ g(\cdot)=\sup_{\alpha\in[0,1]}[\min(\alpha, g(f_\alpha))], \qquad (4)$$

где $f_\alpha=\{y: f(y)\ge\alpha\}$.

Нечеткий интеграл в (4) называется интегралом Сугено. Когда $Y = \{y_1, y_2, \ldots, y_n\}$ есть конечное множество и $0 \le f(y_1) \le f(y_2) \le \ldots \le f(y_n) \le 1$ (если нет, то элементы из Y переупорядочивают, чтобы сохранить эту связь), интеграл Сугено можно вычислить согласно

$$\int_A f(y) \circ g(\cdot) = \max_{i=1}^{m} [\min(f(y_i), g(A_i))],$$

где $A_i = \{y_i, y_{i+1}, \ldots, y_m\}$ и $g(A_i)$ можно рекурсивно рассчитать с точки зрения нечеткой меры g_λ как

$$g(A_i) = g_i + g(A_{i-1}) + \lambda g_i g(A_{i-1}), \quad 1 \le i \le m. \tag{5}$$

Т. Мурофуши и М. Сугено в [29] предложили так называемый интеграл Шоке. Данный интеграл от f по отношению к нечеткой мере g определяется следующим образом:

$$\int_A f(y)\, dg(\cdot) = \sum_{i=1}^{m} [f(y_i) - f(y_{i-1})] g(A_i),$$

где $f(y_0) = 0$.

Интегралы Шоке и Сугено [29] являются идемпотентными, непрерывными, монотонно неубывающими операторами; это свойство подразумевает, что нечеткие интегралы всегда ограничены между минимумом и максимумом. Данные интегралы существенно различны по своей природе, так как первый основывается на линейных операторах, а второй – на нелинейных (минимума и максимума).

Интересным свойством нечеткого интеграла Шоке является то, что если g служит вероятностной мерой, интеграл Шоке эквивалентен классическому интегралу Лебега и вычисляет ожидание f относительно g по обычной вероятностной схеме.

Интеграл Шоке больше подходит для количественной агрегации (где числа имеют реальный смысл), в то время как интеграл Сугено – для порядковой (где только порядок имеет смысл).

3. Метод объединения значений соответствия на основе нечетких интегралов

Блок-схема предлагаемого метода включает выделение признаков для каждого диктора, классификацию и объединение полученных значений соответствия (рис. 2). Нечеткий интеграл используется для объединения результатов классификаторов с целью получения конечного решения. Значения, необходимые для объединения, получаются с учетом близости вектора признаков к соответствующим эталонным значениям.

В работе в качестве признаков рассматриваются мгновенная частота и мгновенная амплитуда [30]: x_1, x_2. Для каждого речевого признака определяются нечеткие меры $g(A_1)$ и $g(A_2)$. На основе формулы (3) вычисляется λ. Далее из формул (5) находятся нечеткие меры для всевозможных комбинаций биометрических характеристик. Полученные нечеткие меры обозначаются через $g(A_1)$ и $g(A_2)$. С помощью функции принадлежности фазифицируются значения соответствия, полученные при сравнении рассматриваемых речевых признаков, что позволяет вычислить нечеткий интеграл.

С помощью метода нечетких интегралов значения соответствия каждого классификатора объединяются в одно значение, а затем принимается решение о пользователе или самозванце, учитывая пороговое значение.

В приложениях реального времени дикторозависимые пороги должны быть оценены априори. Порог определяется как линейная комбинация средних, дисперсий или стандартное отклонение от соотношения пользователь/самозванец [15]. Другие подходы основаны на кривых FAR (False Acceptance Rate) и FRR (False Rejection Rate) [31].

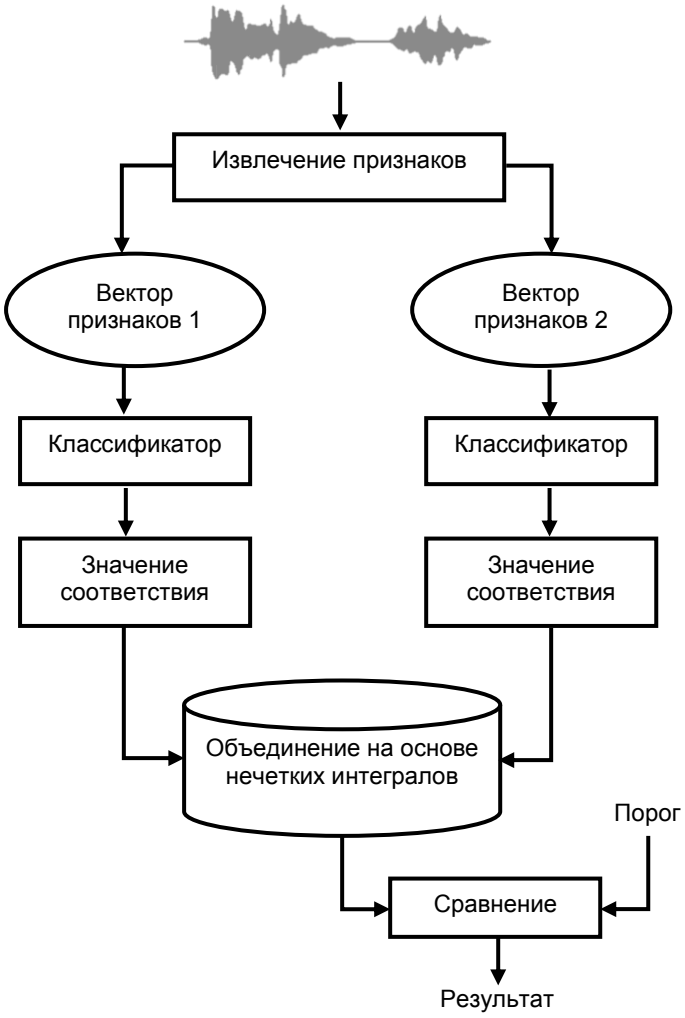


Рис. 2. Блок-схема предлагаемого метода объединения решений классификаторов

Метод на основе нечетких интегралов имеет высокую производительность и прост в применении. Основным преимуществом предлагаемого метода является то, что он может принять во внимание значимость каждого классификатора и взаимодействие между ними.

4. Результаты экспериментов

Для проведения экспериментов были взяты речевые образцы, полученные из пяти гласных английского языка (/a/, /e/, /i/, /o/ и /u/) от различных пользователей. Использовался набор из 50 образцов каждой из пяти гласных для 10 дикторов с целью идентификации на непересекающихся наборах гласных.

Гласные определяют значительную часть информации о речевом тракте человека и поэтому существенны при распознавании диктора. В качестве классификатора был рассмотрен метод SVM (Support Vector Machine) с радиальной базисной функцией ядра. С помощью процедуры кросс-валидации данные разделяются на k частей (блоков). В нашем случае $k=10$.

Эксперименты проводились с применением четырех методов объединения: правила максимума, взвешенной суммы, логарифмического объединения и предлагаемого подхода на основе нечетких интегралов. Веса α_1 и α_2 принимаются равными 0,4 и 0,6 соответственно [32].

Для каждой гласной в табл. 1 показаны полученные значения соответствия (нормализованные в диапазоне [0, 1]) в результате классификации для мгновенных частот и амплитуд ($g(x_1)$ и $g(x_2)$).

Таблица 1

Значения соответствия для гласных /a/, /e/, /i/, /o/ и /u/

Гласные	Мгновенная амплитуда	Мгновенная частота
/e/	0,4420	0,5835
/i/	0,3703	0,5758
/o/	0,5735	0,6796
/a/	0,4677	0,6816
/u/	0,3844	0,5896

На этапе классификации каждый набор речевых данных разделен на обучающий и тестовый (в отношении 90 к 10 соответственно).

В табл. 2 показана эффективность предлагаемого подхода в сравнении с правилом максимума, методами взвешенной суммы и логарифмического объединения для значений мгновенной частоты и мгновенной амплитуды из табл. 1. При распознавании гласной /e/ значения, полученные в результате логарифмического объединения и на основе нечеткого интеграла, совпадают и уступают правилу максимума.

Таблица 2

Результаты объединения значений классификаторов

Гласные	Правило максимума	Метод взвешенной суммы	Метод логарифмического объединения	Предлагаемый метод
/e/	0,5835	0,5269	0,5815	0,5815
/i/	0,5758	0,4936	0,5389	0,5699
/o/	0,6796	0,6372	0,6860	0,7256
/a/	0,6816	0,5960	0,6330	0,7285
/u/	0,5896	0,5075	0,5523	0,5908

Экспериментальное тестирование показало, что предложенный подход дает лучшие результаты на исследуемом наборе речевых данных по сравнению с другими методами.

Заключение

Результаты экспериментов показывают, что метод на основе нечетких интегралов для объединения значений соответствия, полученных в результате классификации мгновенной частоты и мгновенной амплитуды, эффективен для задачи распознавания личности по голосу.

Значимость метода состоит не только в объединении результатов классификаторов, но и в рассмотрении каждого речевого признака в отдельности. Проведенный анализ показывает, что применение нечетких интегралов эффективно для объединения значений соответствия и существенно улучшает проверку идентичности диктора.

Дальнейшие исследования будут направлены на совершенствование предлагаемого метода, а также разработку новых методов для повышения точности распознавания диктора.

Работа выполнена при финансовой поддержке Фонда развития науки при Президенте Азербайджанской Республики – грант № EİF-RİTN-MQM-2/İKT-2-2013-7(13)-29/18/1.

Список литературы

1. Ross, A.A. Handbook of Multibiometrics / A.A. Ross, K. Nandakumar, A.K. Jain. – London : Springer, 2006. – 198 p.

2. Solomonoff, A. Advances in channel compensation for SVM speaker recognition / A. Solomonoff, W. Campbell, I. Boardman // Proc. of ICASSP. – Philadelphia, PA, 2005. – P. 629–632.

3. Gader, P.D. Fusion of handwritten word classifiers / P.D. Gader, M.A. Mohamed, J.M. Keller // Pattern Recognition Letters. – 1996. – № 17. – P. 577–584.

4. Michel, G. The representation of importance and interaction of features by fuzzy measure / G. Michel // Pattern Recognition Letters. – 1996. – № 17. – P. 567–575.

5. Kuncheva, L.I. Decision templates for multiple classifier fusion: an experimental comparison / L.I. Kuncheva, J.C. Bezdek, R.P.W. Duin // Pattern Recognition. – 2001. – № 34. – P. 299–314.

6. Mirhosseini, A.R. Human face image recognition: an evidence aggregation approach / A.R. Mirhosseini, H. Yan // Computer Vision and Image Understanding. – 1998. – № 71. – P. 213–230.

7. Pham, T.D. Color image segmentation using fuzzy integral and mountain clustering / T.D. Pham, H. Yan // Fuzzy sets and systems. – 1999. – № 107. – P. 121–130.

8. Kwak, K.-C. Face recognition using fuzzy integral and wavelet decomposition method / K.-C. Kwak, W. Pedrycz // IEEE Transactions on Systems, Man, and Cybernetics. – 2004. – № 34. – P. 1666–1675.

9. Auephanwiriyakul, S. Generalized Choquet fuzzy integral fusion / S. Auephanwiriyakul, M.K. James, P.D. Gader // Information Fusion. – 2002. – № 3. – P. 69–85.

10. Wolf, J.J. Efficient acoustic parameters for speaker recognition / J.J. Wolf // J. Acoustical Society of America. – 1982. – Vol. 51, № 6. – P. 2044–2056.

11. Kinnunen, T. An overview of text-independent speaker recognition: from features to supervectors / T. Kinnunen, H. Li // Speech Communication. – 2010. – Vol. 52, № 1. – P. 12–40.

12. Rose, P. Forensic speaker identification. Taylor & Francis forensic science series / P. Rose. – N.Y. : Taylor & Francis, 2002. – 380 p.

13. Kinnunen, T. Spectral features for automatic text-independent speaker recognition. Licentiate thesis / T. Kinnunen. – Finland : University of Joensuu, 2003.

14. Маркел, Дж. Линейное предсказание речи / Дж. Маркел, А.Х. Грей. – М. : Связь, 1980. – 308 с.

15. Furui, S. Cepstral analysis techniques for automatic speaker verification / S. Furui // IEEE tran. acoust., speech, signal processing. – 1981. – Vol. 27. – P. 254–272.

16. Reynolds, D. Channel robust speaker verification via feature mapping / D. Reynolds // Proc. of ICASSP. – Hong Kong, 2003. – Vol. 2. – P. 53–56.

17. Doddington, G. Speaker recognition based on idiolectal differences between speakers / G. Doddington // Proc. of Eurospeech. – Aalborg, Denmark, 2001. – Vol. 4. – P. 2521–2524.

18. Hemant, A.P. Forensic Speaker Recognition / A.P. Hemant, Amy Neustein. – Heidelberg : Springer, 2012. – 540 p.

19. Benediktsson, J.A. Consensus theoretic classification methods / J.A. Benediktsson, P.H. Swain // IEEE Trans. Systems Man Cybernet. – 1992. – № 22. – P. 688–704.

20. Ho, T.K. Decision combination in multiple classifier systems / T.K. Ho, J.J. Hull, S.N. Srihari // IEEE Trans. Pattern Anal. Machine Intelligence. – 1994. – № 16. – P. 66–75.

21. Xu, L. Methods of combining multiple classifiers and their applications to hand-written character recognition / L. Xu, A. Krzyzak, C.Y. Suen // IEEE Trans. Systems Man Cybernet. – 1992. – № 23. – P. 418–435.

22. Soong, F.K. On the use of instantaneous and transitional spectral information in speaker recognition / F.K. Soong, A.E. Rosenberg // IEEE Trans. Acoust. Speech, Signal Process. – 1988. – ASSP-36. – P. 871–879.

23. Farrell, K.R. Text-dependent speaker verification using data fusion / K.R. Farrell // IEEE Intern. Conf. on Acoustic, Speech and Signal Processing. – Detroit, Michigan, USA, 1995. – P. 349–352.

24. Sub-word speaker verification using data fusion methods / K.R. Farrell [et al.] // IEEE Workshop on Neural Networks for Signal Processing. – Amelia Island, Florida, 1997. – P. 531–540.

25. Farrell, K.R. An analysis of data fusion methods for speaker verification / K.R. Farrell, R.P. Ramachandran, R.J. Mammone // IEEE Intern. Conf. on Acoustic, Speech and Signal Processing. – Washington, USA, 1998. – P. 1129–1132.

26. Schalkwyk, J. Speaker verification with low storage requirements / J. Schalkwyk, N. Jain, E. Barnard // IEEE Intern. Conf. on Acoustic, Speech and Signal Processing. – Georgia, USA, 1996. – P. 693–696.

27. Zadeh, L.A. Fuzzy sets / L.A. Zadeh // Information and Control. – 1965. – № 8. – P. 338–353.

28. Gupta, M.M. Fuzzy measures and fuzzy integrals / M.M. Gupta, G.N. Saridis, B.R. Gaines. – N.Y. : Elsevier, 1977. – 510 p.

29. Murofushi, T. A theory of fuzzy measures. Representation, the Choquet integral and null sets / T. Murofushi, M. Sugeno // J. Math. Anal. Appl. – 1991. – Vol. 159, № 2. – P. 532–549.

30. Maragos, P. On amplitude and frequency demodulation using energy operators / P. Maragos, J.F. Kaiser, T.F. Quatieri // IEEE Trans. on Signal Processing. – 1993. – Vol. 41, № 4. – P. 1532–1550.

31. Zhang, W.D. A priori threshold determination for phrase-prompted speaker verification / W.D. Zhang [et al.] // Proc. Eurospeech'99. – Budapest, Hungary, 1999. – P. 1203–1206.

32. Hamid, L.A. Quality based Speaker Verification Systems using Fuzzy Inference Fusion Scheme / L.A. Hamid, D.A. Ramli // Proc. of the Intern. Conf. on Communications, Signal Processing and Computers. – Interlaken, Switzerland, 2014. – P. 96–103.

Поступила 27.10.2014

Институт информационных технологий
Национальной академии наук Азербайджана,
Баку, ул. Б. Вахабзаде, 9
e-mail: yadigar@lan.ab.az,
lsuhostat@hotmail.com

Y.N. Imamverdiyev, L.V. Sukhostat

MERGING CLASSIFIER DECISIONS FOR SPEAKER RECOGNITION

The paper proposes using fuzzy integrals for merging classifier decisions in speaker recognition systems. Instantaneous frequency and instantaneous amplitude are considered as the set of features. The approach shows significantly better results than a single classifier. A comparison of the proposed approach with the other methods for merging classifier decisions is provided.

УДК 519.254

В.В. Старовойтов

БИОМЕТРИЧЕСКИЕ СИСТЕМЫ КОНТРОЛЯ ДОСТУПА ПО ОТПЕЧАТКАМ ПАЛЬЦЕВ

Рассматриваются особенности построения биометрических систем контроля доступа людей в определенные помещения. Детально описываются основные алгоритмы обработки и анализа отпечатков пальцев для этих целей. Представляются схемы построения трех вариантов системы контроля доступа.

Введение

Организация системы контроля и управления доступом (СКУД) – это совокупность программно-аппаратных технических средств, целью которых является регулирование входа людей на заданную территорию или доступа к определенным информационным ресурсам. В свою очередь, управление доступом – это разграничение прав доступа, т. е. определение кого, в какое время и на какую территорию (к каким ресурсам) допускать.

Средства СКУД по функциональному назначению подразделяют на следующие группы:

– устройства (преграждающие, управляемые, исполнительные, считывающие);

– идентификаторы;

– средства управления в составе аппаратных устройств и программных средств.

Устройства первой группы представляют собой механические препятствия типа дверей, ворот, турникетов и т. п. и далее в данной работе рассматриваться не будут.

По виду используемых признаков идентификаторы можно разделить на механические средства (карточки, жетоны), пароли и биометрические данные (отпечаток пальца, изображение лица и т. п.). В отличие от бумажных и пластиковых идентификаторов (паспорта, водительских прав) или пароля биометрические характеристики нельзя забыть или потерять, подделать их также достаточно трудно. По оценкам зарубежных специалистов, более 85 % установленных в США средств биометрического контроля доступа предназначались для защиты машинных залов ЭВМ, хранилищ ценной информации, исследовательских центров, военных установок и учреждений. Таким образом, в ближайшем будущем все СКУД будут использовать биометрические данные человека для определения прав его доступа в определенные помещения или к информационным ресурсам.

Главное преимущество биометрической идентификации заключается в том, что идентифицируется конкретный человек, а не отчуждаемый носитель (карта, жетон и т. п.) или пароль. Биометрический идентификатор нельзя забыть, украсть или передать. Современные средства биометрической идентификации обладают развитыми средствами определения муляжей. Таким образом, можно утверждать, что при использовании биометрии отпадает и проблема предъявления поддельных идентификаторов. Вместе с тем не следует противопоставлять различные методы и средства идентификации друг другу. Наиболее эффективно комплексное применение разных технологий, например биометрических средств и смарт-карт. В этом случае цифровую модель идентификатора (например, отпечатка пальца) можно хранить в защищенной памяти смарт-карты и при распознавании пользователя сравнивать модель отпечатка, хранимую в памяти смарт-карты, с моделью отпечатка, предъявляемого в данный момент. При таком подходе биометрическая идентификация дополняется «имущественной» (необходимостью предъявить материальный носитель).

1. Отпечатки пальцев как идентификатор личности

Идентификации человека по отпечаткам пальцев в настоящее время является лидером среди биометрических технологий. Это достаточно точная, дружественная к пользователю и экономичная технология для применения в области идентификации. Данной технологией

в США пользуются, например, ФБР, Секретная служба, Агентство национальной безопасности, министерства финансов и обороны и другие организации.

Преимущества доступа по отпечаткам пальцев – простота использования, удобство и надежность. Критический обзор статистических моделей отпечатков в различных системах идентификации приведен в статье [1]. Современные системы распознавания нельзя обмануть отрубленными пальцами (можно измерить физические параметры кожи, температуру, пульс) и муляжами [2].

Алгоритмы распознавания отпечатков пальцев делятся на два класса [3]: распознавание по отдельным деталям (характерным точкам) и по рельефу всей поверхности пальца. В первом случае устройство анализирует участки, уникальные для конкретного отпечатка, и определяет их взаимное расположение. Во втором случае обрабатывается изображение всего отпечатка. В современных системах часто используется комбинация этих двух способов, что позволяет повысить достоверность идентификации. Регистрация отпечатка пальца человека на оптическом сканере занимает немного времени. Крошечная CCD-камера делает снимок отпечатка пальца. Затем полученное изображение преобразуется в уникальный шаблон отпечатка. Этот шаблон шифруется и записывается в базу данных для аутентификации пользователей.

На сегодняшний день использование отпечатка пальца для идентификации личности – самый удобный для пользователя из всех биометрических методов. Качество распознавания отпечатка и возможность его правильной обработки алгоритмом существенно зависят от состояния поверхности пальца, его положения относительно сканирующего элемента, чистоты пальца и окна сканера, а также от ряда других условий.

Папиллярные узоры формируются совокупностью выступов и впадин на коже. Они различается даже у близнецов. На каждом отпечатке пальца можно определить два типа признаков: глобальные и локальные. Глобальные признаки – это те, которые можно увидеть невооруженным глазом:

– узор типа «петля» (левая, правая, центральная, двойная);

– узор типа «дельта», или «дуга» (простая и острая), – зона, где выступ разветвляется на три линии, которые затем сходятся в одной точке;

– узор типа «спираль» (центральная и смешанная).

Локальные признаки, или минуции, подробно описаны в ГОСТ Р ИСО/МЭК 19794-2:2005 «Информационные технологии. Форматы обмена биометрическими данными. Часть 2. Данные изображения отпечатка» [4]. Он определяет следующие понятия локальных признаков отпечатков пальцев:

1. Папиллярные гребни – это гребни кожи ладонной поверхности кистей и пальцев рук, непосредственно контактирующие с поверхностью при соприкосновении. Уникальный рельеф, образованный папиллярными гребнями на пальце, формирует отпечатки пальцев. На рис. 1 гребни показаны темными полосами, а впадины – светлыми.

а) *б)* *в)*

Рис. 1. Примеры основных типов глобальных признаков отпечатков пальцев: *а)* дуга; *б)* петля; *в)* завиток

2. Контрольные точки (минуции) – точки нарушения непрерывности гребней, которые могут иметь вид окончания, разделения гребней или иметь более сложную составную форму. ГОСТ [4] определяет два основных типа минуций (рис. 2):

бифуркация (раздвоение) гребня – точка, соответствующая области, в которой отпечаток гребня разделяется на два гребня;

окончание гребня – точка, соответствующая области, в которой отпечаток гребня заканчивается или начинается.

Идентификация по отпечаткам пальцев – наиболее распространенная и развитая биометрическая технология. Ее используют до 60 % биометрических систем.

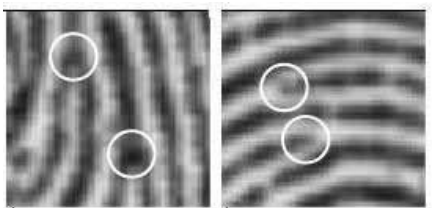


Рис. 2. Примеры окончания и бифуркации (раздвоения) гребня

2. Устройства сканирования отпечатков пальцев

Сканеры последних поколений надежны, компактны и доступны по цене. Для снятия отпечатка и дальнейшего распознавания образца используются три основные технологии: оптическая, полупроводниковая и ультразвуковая.

По соотношению «цена – качество» одними из лучших сканеров отпечатка пальца являются сканеры BioLink U-Match 3.5. Они относятся к устройствам первого типа. Эти сканеры применяются сотрудниками коммерческих компаний и государственных структур более чем в 50 странах мира.

3. Улучшение изображения отпечатка пальца

Отпечатки, полученные стационарно (например, в милиции) под контролем специалиста, обычно имеют хорошее качество и являются информационно-избыточными, т. е. содержат более чем достаточное количество индивидуальных признаков (рис. 3). Алгоритмы обработки и сравнения таких отпечатков достаточно хорошо проработаны (см., например, [3]).

Рис. 3. Отпечатки хорошего качества

Отпечатки, получаемые непосредственно пользователем на пунктах контроля доступа (в спешке, при частично испачканных пальцах или полях сканера), могут привести к получению недостаточно качественного изображения (рис. 4). В связи с этим становится актуальной задача повышения качества изображения и выделения зоны (т. е. сегментации) с четко прослеживаемыми гребнями для надежного выделения минуций и последующего распознавания.

Рис. 4. Пример недостаточно качественного изображения отпечатка

Один из современных подходов к построению модели отпечатков плохого качества описан в работе [5], однако до его применения необходимо выполнить сегментацию области отпечатка и повысить качество изображения.

Алгоритм сегментации и улучшения изображения отпечатка пальца состоит из следующих шагов:

Шаг 1. Нормализация изображения. Выполнить нормализацию исходного изображения отпечатка пальца, чтобы после преобразования оно имело заданные среднее значение и среднеквадратичное отклонение.

Шаг 2. Вычисление локальной ориентации. Вычислить ориентационное изображение из нормализованного изображения отпечатка пальца.

Шаг 3. Оценка локальной частоты хребтов. Вычислить матрицу частот на базе нормализованного и ориентационного изображений.

Шаг 4. Сегментация отпечатка. Построить маску отпечатка путем разбиения нормализованного изображения на блоки и выполнения классификации каждого блока, разделив их на содержащие хребты отпечатки и не содержащие. Затем маску сгладить с помощью морфологических фильтров.

Шаг 5. Фильтрация нормализованного изображения. Применить набор фильтров (Габора или подобных), настроенных на локальную ориентацию выступов и частоту выступов, к пикселам хребтов и впадин в нормализованном изображении для получения улучшенного изображения отпечатка пальца. Для построения шаблона отпечатка использовать часть изображения, которое получено после фильтрации изображения, попавшего в маску, построенную на шаге 4.

Опишем перечисленные шаги более детально.

Нормализация изображения. Пусть $I(i, j)$ обозначает полутоновое значение (уровень яркости) пиксела (i, j); M и VAR – среднее значение и среднеквадратическое отклонение изображения I соответственно; $G(i, j)$ – нормализованное полутоновое значение пиксела (i, j). Нормализованное изображение G (рис. 5) вычисляется по формуле

$$G(i, j) = \begin{cases} M_0 + \sqrt{\dfrac{VAR_0 (I(i, j) - M)^2}{VAR}}, & \text{если } I(i, j) > M; \\ M_0 - \sqrt{\dfrac{VAR_0 (I(i, j) - M)^2}{VAR}} & \text{в противном случае,} \end{cases}$$

где M_0 и VAR_0 – заданные значения среднего и среднеквадратического отклонения соответственно.

а) *б)*

Рис. 5. Результат нормализации изображения отпечатка: *а)* исходное изображение *I*; *б)* нормализованное изображение *G* ($M_0 = 100$, $VAR_0 = 100$)

Вычисление локальной ориентации. Ориентационное изображение передает важные свойства отпечатков пальцев и определяет постоянные координаты для выступов и впадин в локальном соседстве.

Пусть *G* – нормализованное изображение. Для вычисления локальной ориентации к нормальному изображению применяются следующие действия:

– разделить *G* на блоки размером $w \times w$ (например, 16 × 16);

– вычислить градиенты $dx(i, j)$ и $dy(i, j)$ в каждом пикселе (i, j);

– оценить локальную ориентацию в каждом блоке относительно центрального пиксела (i, j):

$$V_x(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2d_x(u,v)d_y(u,v);$$

$$V_y(i, j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} \left(d_x^2(u,v)d_y^2(u,v)\right);$$

$$\theta(i, j) = \frac{1}{2}\tan^{-1}\left(\frac{V_y(i, j)}{V_x(i, j)}\right),$$

где $\theta(i, j)$ – оценка методом наименьших квадратов локальной ориентации выступа в блоке, расположенном симметрично относительно пиксела (i, j). Она представляет направление, которое является ортогональным к доминантному направлению спектра Фурье в окне размерности w x w (рис. 6).

Рис. 6. Пример ориентационного поля: показано белыми стрелками для параметров $w = 16$ и $w_\phi = 5$

Вследствие шума, искажения структур выступов, впадин и других деталей на входном изображении рассчитанная локальная ориентация выступа $\theta(i, j)$ не всегда корректна. Так как локальная ориентация выступа изменяется медленно в соседних блоках, где нет особых точек, применяем низкочастотный фильтр для ее корректировки. Чтобы выполнить низкочастотную фильтрацию, ориентационное изображение необходимо преобразовать в непрерывное векторное поле:

$$\Phi_x(i, j) = \cos(2\theta(i, j));$$

$$\Phi_y(i, j) = \sin(2\theta(i, j)),$$

где $\hat{O}_x$ и $\hat{O}_y$ – компоненты *x* и *y* векторного поля соответственно.

Полученное поле подвергается низкочастотной фильтрации следующим образом:

$$\Phi'_x(i, j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u,v)\Phi_x(i - uw, j - vw);$$

$$\Phi'_y(i, j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u,v)\Phi_y(i - uw, j - vw),$$

где W – двухмерный низкочастотный фильтр и $w_\Phi \times w_\Phi$ определяет размер фильтра. Операция сглаживания выполняется в блоках, размер фильтра равен 5 x 5.

Локальная ориентация выступа в пикселе (i, j) вычисляется по формуле

$$O(i, j) = \frac{1}{2}\tan^{-1}\left(\frac{\hat{O}'_y(i, j)}{\hat{O}'_x(i, j)}\right).$$

Оценка локальной частоты хребтов. Если минуции не обнаружены локально, уровни яркости вдоль гребней могу быть смоделированы как синусоидальная волна вдоль нормали к ориентации хребта (рис. 7).

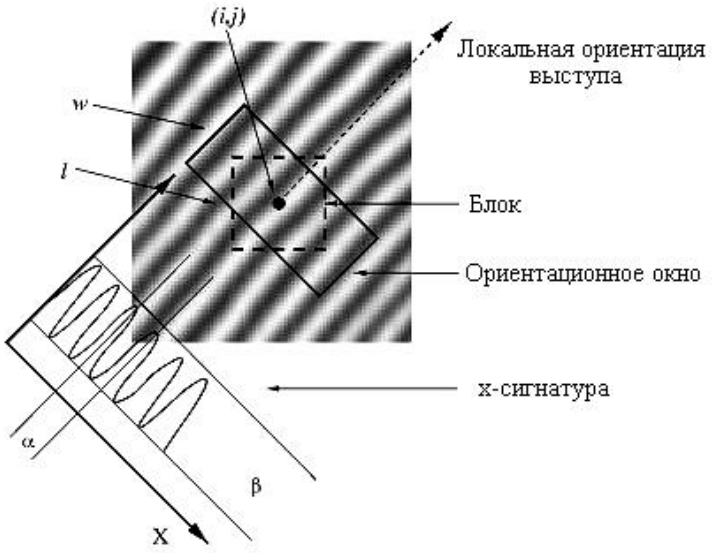


Рис. 7. Ориентационное окно и *x*-сигнатура

Локальная частота выступов – это важное свойство, присущее изображению отпечатка пальца. Пусть G – нормализованное изображение, а O – ориентационное. Тогда для оценки локальной частоты выступов необходимо:

– разделить G на блоки размером $w \times w$ (16 × 16);

– для каждого блока, центрированного в пикселе (i, j), вычислить ориентационное окно размером $l \times w$ (32 × 16), которое определяется в координатной системе выступа;

– для каждого блока, центрированного в пикселе (i, j), вычислить x-сигнатуру $X[0]$, $X[1]$, ..., $X[l-1]$ выступов и впадин в пределах ориентационного окна:

$$X\left[k\right] = \frac{1}{w}\sum_{d=0}^{w-1} G(u,v),\ k = 0,\ 1,\ ...,\ l-1;$$

$$u = i + \left(d - \frac{w}{2}\right)\cos O(i,j) + \left(k - \frac{l}{2}\right)\sin O(i,j);$$

$$v = j + \left(d - \frac{w}{2}\right)\sin O(i,j) + \left(\frac{l}{2} - k\right)\cos O(i,j).$$

Если не обнаружено особых точек в ориентационном окне, x-сигнатура формирует дискретную синусоидально очерченную волну, которая имеет такую же частоту, как выступы и впадины в ориентационном окне. Поэтому частота выступов и впадин может быть оценена из x-сигнатуры. Пусть $T(i, j)$ – среднее число пикселов между двумя следующими друг за другом вершинами в x-сигнатуре, тогда частота Ω вычисляется как $\Omega = 1 / T(i, j)$.

Если не обнаружено следующих друг за другом вершин из x-сигнатуры, частота устанавливается в значение –1, чтобы отличить ее от действительных частотных значений.

Для изображения отпечатка пальца, отсканированного с постоянным разрешением, значение частоты выступов и впадин в локальном соседстве лежит в определенном диапазоне. Для изображения, отсканированного с разрешением 500 dpi, этот диапазон равен $\left[\frac{1}{4} \div \frac{1}{21}\right]$.

Блоки, в которых детали или особые точки обнаружены, но выступы и впадины испорчены или искажены, не формируют четкую синусоидально очерченную волну. Частотные значения для этих блоков должны быть интерполированы по частотам соседних блоков, которые имеют хорошо определяемую частоту. Для каждого блока, расположенного симметрично относительно (i, j):

$$\Omega'(i,j) = \begin{cases} \Omega(i,j), \text{ если } \Omega(i,j) \neq -1; \\ \dfrac{\sum_{u=-w_\Omega/2}^{w_\Omega/2}\sum_{v=-w_\Omega/2}^{w_\Omega/2} W_g(u,v)\mu\left(\Omega\left(i-uw, j-vw\right)\right)}{\sum_{u=-w_\Omega/2}^{w_\Omega/2}\sum_{v=-w_\Omega/2}^{w_\Omega/2} W_g(u,v)\delta\left(\Omega\left(i-uw, j-vw\right)+1\right)} \text{ в противном случае,} \end{cases}$$

где $$\mu(x) = \begin{cases} 0, \text{ если } x \leq 0, \\ x \text{ в противном случае;} \end{cases}$$

$$\delta(x) = \begin{cases} 0, \text{ если } x \leq 0, \\ 1 \text{ в противном случае;} \end{cases}$$

Wg – дискретное ядро Гаусса со средней величиной 0 и изменением 9; $W_\Omega = 7$ – размер ядра.

Если существует по хотя бы один блок с частотным значением –1, необходимо поменять местами Ω и Ω' и повторить предыдущие вычисления еще раз.

Расстояние между выступами локально меняется медленно, поэтому низкочастотный фильтр может быть использован для сглаживания полученных значений:

$$F(i, j) = \sum_{u=-w_{\Omega/2}}^{w_{l/2}} \sum_{v=-w_{\Omega/2}}^{w_{l/2}} W_l(u,v)\Omega'(i - uw, j - vw),$$

где W_l – двухмерный низкочастотный фильтр с модульным интегралом; $W_l = 7$ – размер фильтра.

Сегментация отпечатка. Изображение разбивается на блоки размером $W_l \times W_l$, в каждом блоке вычисляется среднеквадратичное отклонение std_{ij}. Эмпирически выбирается порог T. Блоки, в которых $std_{ij} > T$, считаются содержащими хребты. Остальные блоки считаются неинформативными. Строится бинарная маска информативности блоков. Полученная маска подвергается фильтрации морфологической операцией замыкания для сглаживания краев.

Фильтрация нормализованного изображения. Очертания параллельных хребтов и впадин с хорошо определяемой частотой и ориентацией на изображении отпечатка пальца содержат полезную информацию, которая помогает устранить нежелательные шумы. Для этого используется полосовой фильтр, который настраивается на соответствующую частоту и ориентацию. Он может эффективно удалять нежелательные шумы и сохранять достоверные структуры хребтов и впадин. Фильтры Габора дают оптимальное решение этой задачи как в пространственной, так и в частотной областях, поэтому их целесообразно использовать как полосовые фильтры.

Фильтр Габора описывается формулой

$$h(x, y : \varphi, f) = \exp\left\{-\frac{1}{2}\left[\frac{x_\varphi^2}{\delta_x^2} + \frac{y_\varphi^2}{\delta_y^2}\right]\right\}\cos(2\pi f x_\varphi),$$

$$x_\varphi = x\cos\varphi + y\sin\varphi,$$

$$y_\varphi = x\sin\varphi + y\cos\varphi,$$

где φ – ориентация фильтра Габора; f – частота синусоидальной плоскостной волны; δ_x и δ_y – пространственные константы огибающей Гаусса вдоль осей x и y соответственно. Модуляционно-передаточная функция фильтра Габора определяется выражением

$$H(u, v : \varphi, f) = 2\pi\delta_x\delta_y \exp\left\{-\frac{1}{2}\left[\frac{(u_\varphi - u_0)^2}{\delta_u^2} + \frac{(v_\varphi - v_0)^2}{\delta_v^2}\right]\right\} + 2\pi\delta_x\delta_y \exp\left\{-\frac{1}{2}\left[\frac{(u_\varphi + u_0)^2}{\delta_u^2} + \frac{(v_\varphi + v_0)^2}{\delta_v^2}\right]\right\},$$

$$u_\varphi = u\cos\varphi + v\sin\varphi,$$

$$v_\varphi = -u\sin\varphi + v\cos\varphi,$$

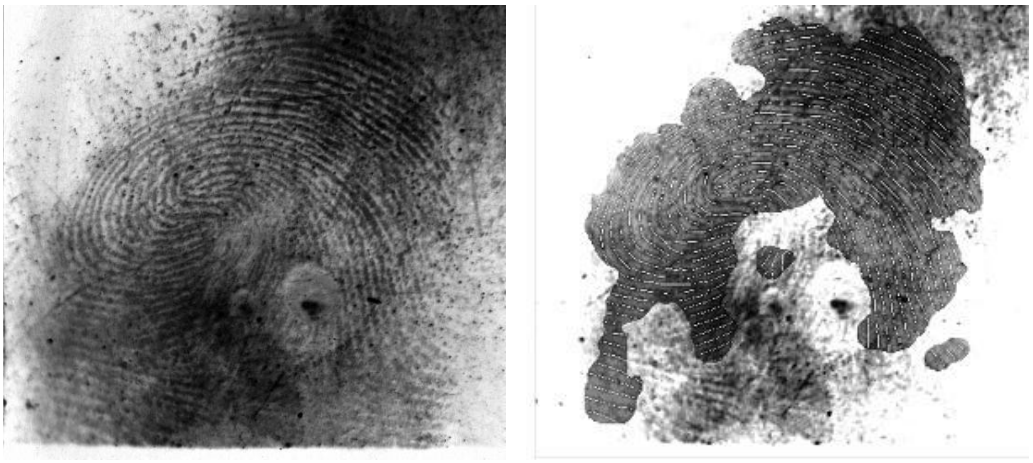
$$u_0 = \frac{2\pi\cos\varphi}{f},$$

$$v_0 = \frac{2\pi\sin\varphi}{f},$$

где $\delta_u = 1/2\pi\delta_x$ и $\delta_v = 1/2\pi\delta_y$.

Для применения фильтров Габора к изображению задаются три параметра: частота синусоидальной плоскостной волны f, направление фильтра, среднеквадратичные отклонения огибающей (оболочки) Гаусса δ_x и δ_y.

Частотная характеристика фильтра *f* определяется локальной частотой хребтов, а направление – локальной ориентацией выступа. Выбор значений δ_x и δ_y содержит компромисс. Чем больше эти значения, тем фильтры более устойчивы к шумам, но при этом возрастает вероятность того, что фильтры будут создавать ложные выступы и впадины. С другой стороны, чем меньше значения δ_x и δ_y, тем менее вероятно, что фильтры будут создавать ложные выступы и впадины; следовательно, они будут менее эффективны в устранении шумов. Значения δ_x и δ_y выбраны равными 4,0 на основе экспериментов.



а) *б)*

Рис. 8. Построение улучшенного изображения: *а)* исходное изображение низкого качества; *б)* улучшенное изображение с выделенной областью, содержащей хребты (показана темным, центральные линии хребтов показаны белым)

Пусть *G* – нормализованное изображение отпечатка пальца, *O* – ориентационное изображение, *F* – частотное изображение, а *R* – восстанавливающая маска. Тогда улучшенное изображение *E* (рис. 8) вычисляется по формуле

$$\mathrm{E}(i,j)=\begin{cases}255, \text{ если } R(i,j)=0;\\ \sum\limits_{u=-w_g/2}^{w_g/2}\sum\limits_{v=-w_g/2}^{w_g/2} h\big(u,v:O(i,j),F(i,j)\big)G(i-u,j-v) \text{ в противном случае,}\end{cases}$$

где w_g = 11 и определяет размер фильтров Габора.

Дополнительную информацию по этому вопросу можно найти в работах В.Ю. Гудкова [6].

4. Сравнение отпечатков по найденным локальным признакам

Алгоритм сравнения отпечатков по локальным признакам состоит из следующих шагов:

Шаг 1. Улучшение качества исходного изображения отпечатка. Повысить резкость папиллярных линий (хребтов) в найденной маске.

Шаг 2. Бинаризация изображения отпечатка. Преобразовать изображение к черно-белому представлению пороговой обработкой.

Шаг 3. Утончение линий изображения отпечатка. Выполнить утончение бинарного изображения до получения линий шириной 1 пиксел.

Шаг 4. Выделение минуций. Изображение разбить на блоки (например, 9x9 пикселов). Анализируя окрестности каждого пиксела, выделить окончания и раздвоения хребтов (рис. 9).

Рис. 9. Пример выделения минуций

Координаты обнаруженных минуций и их углы ориентации записать в вектор $W(p) = [(x_1, y_1, t_1), (x_2, y_2, t_2), \ldots, (x_p, y_p, t_p)]$, где p – число минуций. При регистрации пользователей этот вектор считается эталоном и записывается в базу данных. При распознавании вектор определяет текущий отпечаток.

Шаг 5. Сопоставление минуций. Два отпечатка одного пальца будут отличаться друг от друга поворотом, смещением, изменением масштаба и/или площадью соприкосновения в зависимости от того, как пользователь прикладывает палец к сканеру. Поэтому процесс сопоставления выполняется для разных пар минуций (рис. 10). При поиске для каждой минуции перебирают до 30 значений поворота (от −15 до +15°), 500 значений сдвига (от −250 до +250 пикселов) и 10 значений масштаба (от 0,5 до 1,5 с шагом 0,1), т. е. до 150 000 вариантов для каждой из 70 возможных минуций.

Шаг 6. Принятие решения о совпадении отпечатков. Оценка совпадения отпечатков выполняется по формуле $K = \dfrac{D^2}{pq} \cdot 100\,\%$, где D – количество совпавших минуций; p – количество минуций шаблона, хранящегося в базе; q – количество минуций предъявленного отпечатка. Если K превышает 65 %, отпечатки считаются идентичными. Для более высокого уровня защиты от незарегистрированного пользователя порог может быть повышен.

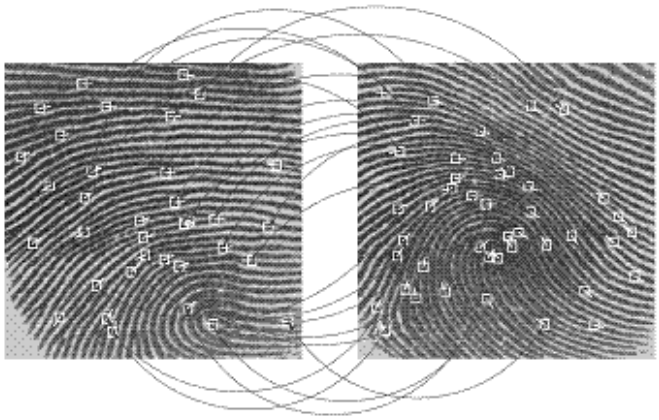


Рис. 10. Пример сравнения одинаковых точек на двух отпечатках

Более подробно о точности результатов сравнения отпечатков пальцев 13 разными способами можно прочитать в статье [7].

5. Организация системы биометрического контроля доступа

Ниже описаны три схемы реализации биометрических систем под конкретные применения, в каждой из них на одного пользователя можно зарегистрировать до 10 отпечатков пальцев, причем в базу данных записываются только шаблоны отпечатков. Далее пользователю присваиваются пра-

ва доступа на конкретные точки прохода, при этом шаблоны отпечатков в кодированном виде передаются по линии связи в контроллер биометрического терминала и хранятся в нем независимо от компьютера. Кратко опишем схемы построения систем контроля доступа [8].

5.1. Схема автономного биометрического терминала для контроля доступа в помещение

Автономный биометрический терминал (биометрический замок) управляет точкой прохода (дверью) без подключения к компьютеру (рис. 11). Устройство представляет собой контроллер и считывающий сенсор отпечатков пальцев в едином корпусе. Для открытия двери пользователь прикладывает палец к сканеру отпечатков пальцев. После положительного сравнения отпечатка пальца с хранящимися в базе шаблонами устройство открывает замок. Для выхода используется кнопка выхода (как в домофонной системе).

Дополнительно может быть подключена сирена для оповещения о взломе устройства. Это самый простой вариант применения биометрических технологий по отпечатку пальца для доступа в помещение. Применяется для реализации СКУД на одной точке прохода или в нескольких точках с независимым программированием оборудования. Преимуществом биометрического замка является автономное питание от встроенных пальчиковых батареек.

Рис. 11. Пример автономного биометрического терминала для контроля доступа в помещение

5.2. Схема сетевого биометрического терминала для контроля доступа в помещение

Сетевой терминал доступа может подключаться к компьютеру по сетевым интерфейсам RS485, Ethernet (рис. 12). Сравнение отпечатков пальцев может производиться как в самом терминале, так и на сервере с помощью режима серверной идентификации. При сравнении отпечатков пальцев на сервере могут создаваться практически не ограниченные по размеру базы данных шаблонов отпечатков. Сетевые терминалы используются для создания распределенных сетевых систем контроля доступа. На компьютер устанавливается программное обеспечение, которое управляет одновременно большим количеством терминалов. К нему может подключаться usb-сканер отпечатков пальцев для регистрации отпечатка пальца пользователя в системе. Шаблоны отпечатков разрешенных пользователей можно загрузить во все терминалы в сети.

Рис. 12. Пример сетевого биометрического терминала для контроля доступа в помещение

5.3. Схема сетевого биометрического терминала для контроля доступа к информационным ресурсам

На рис. 13 представлен один из вариантов сетевой распределенной системы с разграничением прав доступа пользователей. Она открыта для интеграции с устройствами других производителей и при необходимости может наращиваться. Организация сети строится с использованием интерфейса RS485 и выделенных линий связи Ethernet либо сотовых сетей формата GSM. Биометрические терминалы объединяются в магистраль RS485 (до 255 шт.).

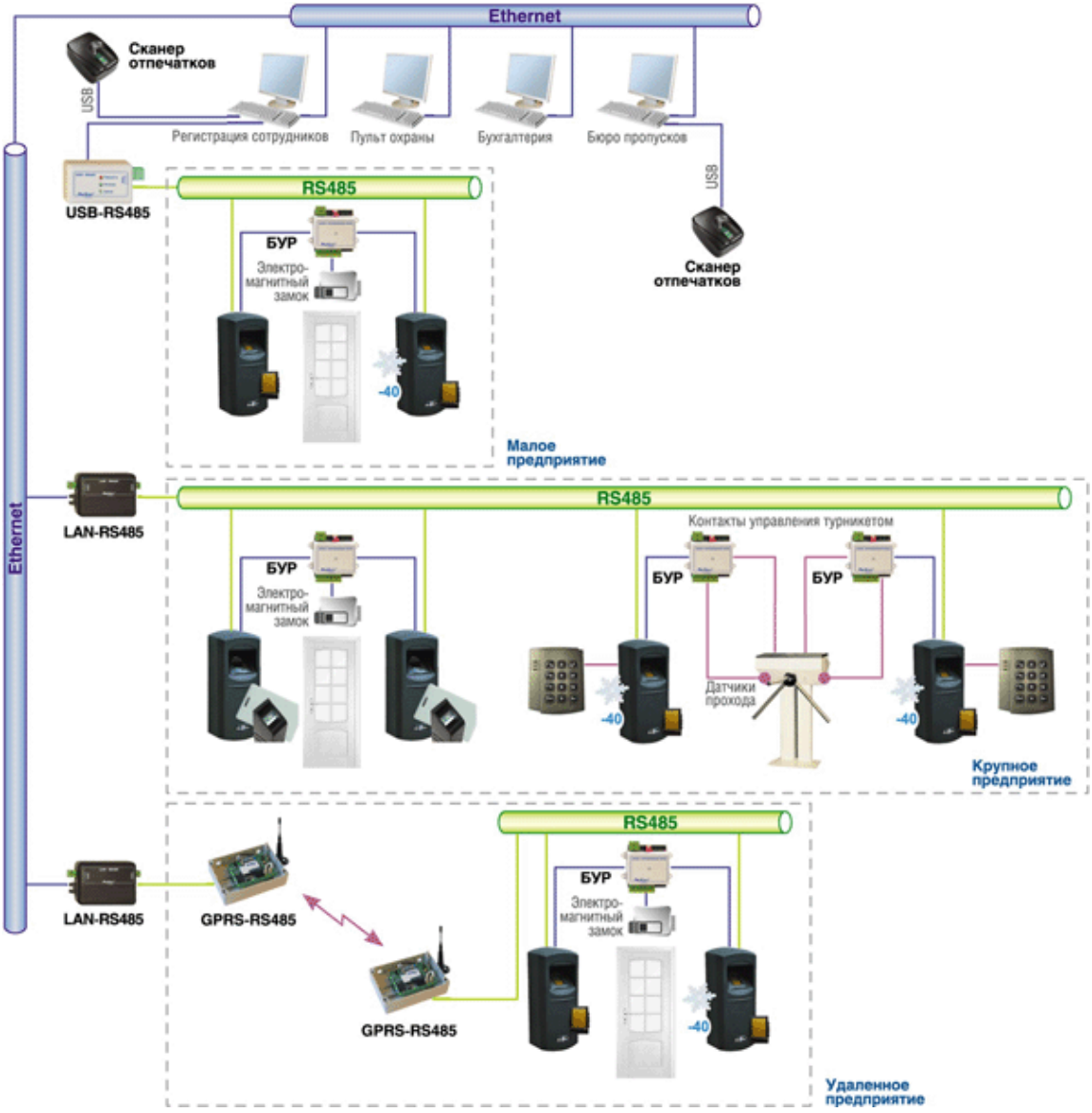


Рис. 13. Пример сетевого биометрического терминала для контроля доступа к информационным ресурсам

Заключение

Сегодня использование отпечатков пальцев при идентификации личности – наиболее простой и комфортный для пользователя биометрический метод доступа. Поэтому для организации системы контроля и управления доступом людей на заданную территорию или к определенным информационным ресурсам предлагается использовать биометрическую технологию на основе признаков, извлеченных из отпечатков пальцев. Биометрические системы подобного типа все чаще используются для различных практических приложений, однако детальные описания алгоритмов и особенности построения подобных систем в литературе отсутствуют.

В статье описаны ключевые алгоритмы обработки и анализа изображений отпечатка пальца различного качества, представлены схемы трех вариантов организации биометрической системы контроля доступа в помещение и к информационным ресурсам. Приведены ссылки на ключевые работы по биометрической идентификации с использованием отпечатков пальцев.

Список литературы

1. Modern statistical models for forensic fingerprint examinations: A critical review / J. Abraham [et al.] // Forensic Science International. – 2013. – Vol. 232, no. 1–3. – P. 131–150.

2. Ларин, П.З. Обведем вокруг пальца? Обман биометрических систем доступа, использующих дактилоскопическую идентификацию личности / П.З. Ларин, Е.И. Ревер // Информационная безопасность. – 2004. – № 3. – С. 24–27.

3. Handbook of fingerprint recognition / D. Maltoni [et al.]. – N.Y. : Springer-Verlag, 2009. – 494 p.

4. Информационные технологии. Форматы обмена биометрическими данными. Часть 2. Данные изображения отпечатка : ГОСТ Р ИСО/МЭК 19794-2:2005. – Введ. 29.12.05. – М. : Федеральное агентство по техническому регулированию и метрологии, 2005. – 42 с.

5. Muñoz-Briseño, A. Fingerprint indexing with bad quality areas / A. Muñoz-Briseño, A. Gago-Alonso, J. Hernández-Palancar // Expert Systems with Applications. – 2013. – Vol. 40. – P. 1839–1846.

6. Гудков, В.Ю. Математические модели и методы обработки цифровых дактилоскопических изображений : автореф. дис. … д-ра физ.-мат. наук : 05.13.18 / В.Ю. Гудков ; Челябинский гос. ун-т. – Челябинск, 2010. – 40 с.

7. Haber, R.N. Experimental results of fingerprint comparison validity and reliability : A review and critical analysis / R.N. Haber, L. Haber // Science and Justice, 2014. – Vol. 54. – P. 375–389.

8. Схемы монтажа биометрических терминалов контроля доступа [Электронный ресурс]. – Режим доступа : http://fingerprint.com.ua/article/schaccessterminal.html. – Дата доступа : 15.01.2015.

Поступила 14.01.2015

Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: valerys@newman.bas-net.by

V.V. Starovoitov

BIOMETRIC ACCESS CONTROL SYSTEMS BASED ON FINGERPRINTS

Features of biometric access control system design for control people's access to certain facilities are described. Basic algorithms for fingerprint processing and analysis are given in details. Construction schemes of three variants of an access control system are presented.

УДК 004.272.43+004.272.32

М.М. Татур

ОСОБЕННОСТИ ПОСТРОЕНИЯ ВЫЧИСЛИТЕЛЕЙ ИНТЕЛЛЕКТУАЛЬНОЙ ОБРАБОТКИ ДАННЫХ

Излагается подход к построению параллельных машин, проблемно-ориентированных на широкий круг задач интеллектуальной обработки данных. Представляется архитектура верхнего уровня, где в качестве функциональных единиц выступают алгоритмы Data Mining, а в качестве ускорителя – универсальный либо специализированный сопроцессор. Делается акцент на необходимость обеспечения совместимости библиотечных алгоритмов и архитектуры параллельного сопроцессора. Приводится методика корректной сравнительной оценки альтернативных аппаратных платформ в рамках предложенного подхода.

Введение

Современная теория интеллектуального анализа данных представлена широким перечнем математических методов и алгоритмов, позволяющих решать задачи кластеризации, классификации, ассоциативного поиска и др., получать количественные оценки эффективности решения таких задач. В ряде работ вскрыты принципиальные ограничения, присущие известным методам, и предложены возможные пути их преодоления [1, 2]. Однако практические достижения в области построения интеллектуальных систем остаются достаточно скромными: созданные системы назвать интеллектуальными можно лишь с большой натяжкой, а время, трудоемкость и стоимость разработки интеллектуальных систем продолжают оставаться неприемлемо большими. В настоящей работе предлагается следующая версия объяснения такого отставания в практике построения интеллектуальных систем.

Большинство актуальных прикладных задач выходят за рамки отдельно взятого формального алгоритма классификации, регрессии и т. п. Существует определенная проблема представления решаемой прикладной задачи как композиции формальных алгоритмов интеллектуального (и не только) анализа данных. Поэтому решения прикладных задач зачастую выглядят либо попыткой «лобового» применения одного из методов, известных инженеру-разработчику, либо «гремучей смеси» различных алгоритмов и эвристик. Крайне редко встречаются действительно научно и технически обоснованные композиции методов и алгоритмов, позволяющих получить гармоничное, эффективное решение прикладной задачи.

Еще с большей очевидностью дисбаланс в достижениях теории и практики проявляется в разработке высокопроизводительных аппаратных средств поддержки интеллектуальных вычислений. Это связано с тем, что создание специализированных аппаратных средств идет вслед за алгоритмическим проектированием и поэтому на порядок сложнее и затратнее во всех отношениях. Ярким примером этому является судьба нейрокомпьютеров и нейрочипов, которым пророчили большое будущее. Между тем оказалось, что помимо чисто технических ограничений (число связей, число одновременно моделируемых нейронов) модель нейронных сетей далеко не всегда обеспечивает эффективное решение реальных интеллектуальных задач. В итоге нейрокомпьютеры вроде бы есть, а покупателей нейрокомпьютеров, готовых применить их в создаваемых системах, – нет [3, 4]. Под данное утверждение не подпадают те случаи, когда отдельные производители электронных компонентов свои, по сути, сигнальные процессоры (DSP) называли нейрочипами, поскольку и в первом и во втором случаях в качестве массовой операции выступает взвешенное суммирование [5].

Между тем новый технологический виток, связанный с развитием многоядерных CPU, GPU, суперкомпьютеров, кластеров, распределенных и облачных вычислений, по сути унифицированных параллельных вычислительных архитектур, составил дополнительную конкуренцию специализированным аппаратным средствам поддержки интеллектуальных вычислений. Возникает резонный вопрос, что же может явиться альтернативой «жестким» нейронным, ассоциативным, семантическим и тому подобным процессорам, с одной стороны, и универсальным

параллельным компьютерам – с другой. Может быть, это будут интеллектуальные, или когнитивные, компьютеры, проблемно-ориентированные на широкий круг задач интеллектуального анализа данных?

1. Спецификации функций и архитектура верхнего уровня

Ограничим исследование вопроса уровнем электронных и компьютерных технологий, доступных в настоящее время в Беларуси (а также в России), в разумной ценовой категории. Это значит, что не будут рассматриваться области «живой» элементной базы, самоорганизующихся синергетических архитектур, трехмерных чипов, нано-, квантовых, фотонных и тому подобных компьютеров. Целью настоящей работы является определение прагматичного подхода, который откроет путь к созданию эффективных программно-аппаратных комплексов как основы для построения интеллектуальных систем. Предлагаемые идеи и пути решения являются дальнейшим развитием авторских результатов (математических моделей нечеткой классификации, семантической обработки знаний и их программно-аппаратных реализаций) [6–8] в направлении унификации.

В настоящей работе предлагается оригинальная архитектура проблемно-ориентированной машины, которая представляет собой надстройку над архитектурой обычного параллельного либо специализированного компьютера. Архитектура включает следующие составные части: центральный процессор или супервизор, параллельный сопроцессор в качестве ускорителя, а также библиотеку программ интеллектуальной обработки данных (рис. 1). Связка центральный процессор – сопроцессор является общеизвестной и широко применяемой, а отличительные особенности заключаются в появлении библиотеки программ алгоритмов. (Конструктивные варианты реализации вычислительной системы могут быть различными и в настоящей работе не рассматриваются.)

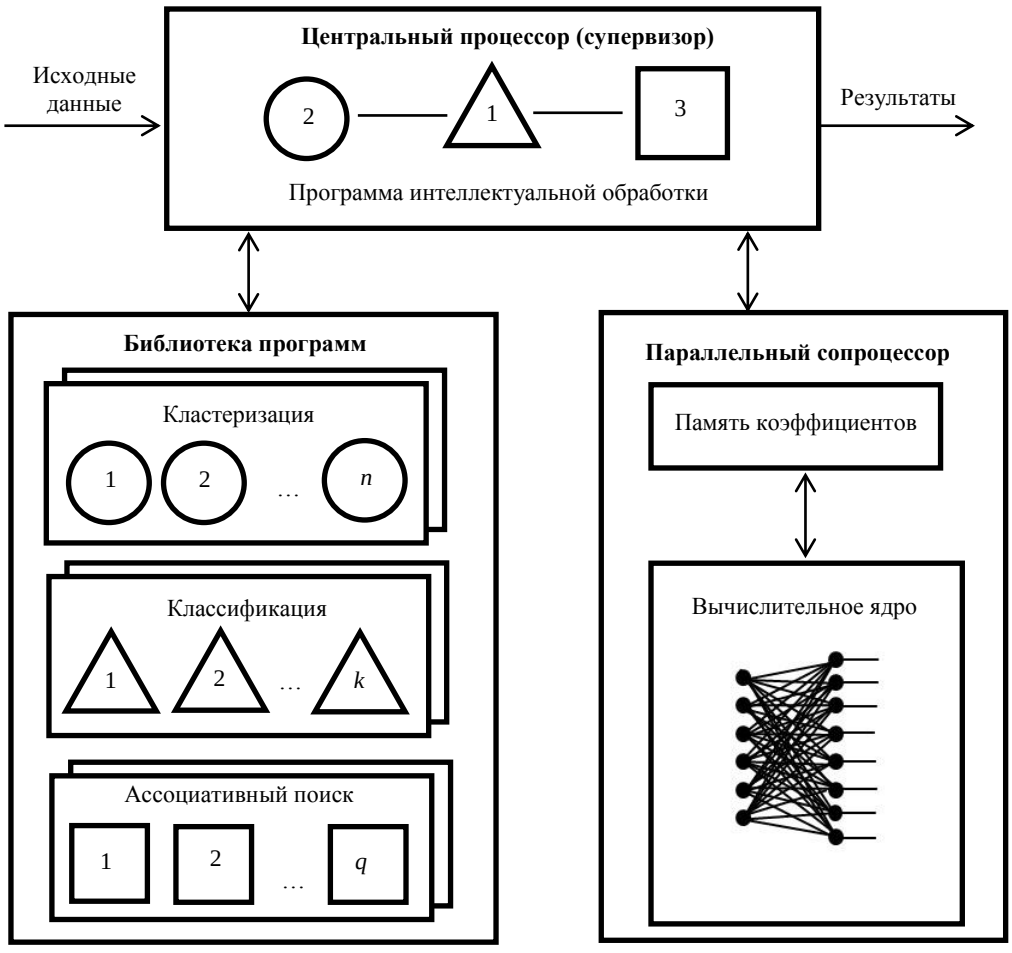


Рис. 1. Обобщенная схема интеллектуального, или когнитивного, суперкомпьютера

Концептуально предлагаемый подход реализует принцип программного управления высокого уровня. Полагается, что прикладная задача интеллектуальной обработки данных может быть представлена как композиция формальных задач кластеризации, классификации, регрессии, ассоциативного поиска, семантической обработки и некоторых других, включая задачи математической статистики. Для каждой из задач известен перечень алгоритмов реализации, причем большинство из них являются переборными, а значит, вычислительная сложность возрастает нелинейно в зависимости от объема входных данных. Каждый из алгоритмов реализован в виде программы, оптимизированной под конкретную аппаратную реализацию параллельного сопроцессора. Также в библиотеке содержатся программы алгоритмов оценки эффективности решения задач кластеризации, классификации и др.

Головная программа интеллектуальной обработки запускается в центральном процессоре и состоит из программ отдельных алгоритмов из библиотеки, а также вспомогательных команд, не включенных в библиотеку. На рис. 1 показан упрощенный пример такой программы, состоящей из алгоритма кластеризации 2, алгоритма классификации 1 и алгоритма ассоциативного поиска 3. Эта же программа осуществляет прием исходных данных, организацию ускорения вычислений на параллельном сопроцессоре и выдачу результатов обработки.

С функциональной точки зрения анонсируемая вычислительная система напоминает программные системы Wolfram Mathematica и Weka [9, 10]. Отличие состоит в применении параллельного универсального сопроцессора, например кластера GPU, и адаптации алгоритмов под его архитектуру. Для каждого алгоритма библиотеки будет определен расчетный показатель эффективности параллельной реализации на сопроцессоре, а это значит, что в головной программе, как правило, будут применяться алгоритмы с высокими показателями эффективности.

В случае оригинальной программно-аппаратной реализации сопроцессора следует ожидать более высокую степень согласованности алгоритмов библиотеки и архитектуры параллельного спецвычислителя, а в результате и эффективность вычислительной системы в целом.

2. Сравнение эффективности аппаратных платформ при построении интеллектуальных компьютеров

Общепринято под эффективностью понимать отношение достигаемого качества решения задачи к стоимости или другим затраченным ресурсам. Применительно к компьютерам это традиционно отношение производительности к стоимости или объему оборудования. Из опыта оценки эффективности нейрокомпьютеров известно, что оценивать производительность в флопсах оказалось не совсем корректным и поэтому были предложены другие меры, например такие, как число моделируемых синапсов (умножений со сложением), нейронов (то же плюс пороговая функция) в единицу времени. В ряде работ по нейронным сетям заявляется, что они по определению предполагают эффективную параллельную реализацию. Однако при этом обычно опускается немаловажный вопрос, сколько нейронов могут моделироваться одновременно. Простейшие эксперименты по моделированию нейронных сетей на параллельных компьютерах показывают, что закономерности Амдаля – Густавсона распространяются и на параллельную реализацию нейросетей, т. е. увеличение числа процессорных элементов не приводит к линейному росту производительности вычислений в целом. Это и понятно, так как проблема распараллеливания при моделировании имеет отношение не только к параллельной компьютерной архитектуре, но и к реализуемой модели.

Какую эффективность можно ожидать от проблемно-ориентированного когнитивного компьютера? Чтобы ответить на этот вопрос, сначала укажем и прокомментируем параметры, которые в той или иной мере определяют эффективность: время вычисления тестовой задачи, объем входных данных тестовой задачи, точность вычисления тестовой задачи и число процессорных элементов. Стоимость, естественно, является интегральным параметром, характеризующим эффективность вычислительных машин. Однако в настоящей работе стоимость как параметр рассматриваться не будет, а будет полагаться, что число процессорных элементов характеризует общие затраты ресурсов на достижение определенной производительности.

Объем входных (обрабатываемых) данных – это величина, которая определяется числом образов, числом классов и (или) кластеров, числом информативных признаков (размерностью

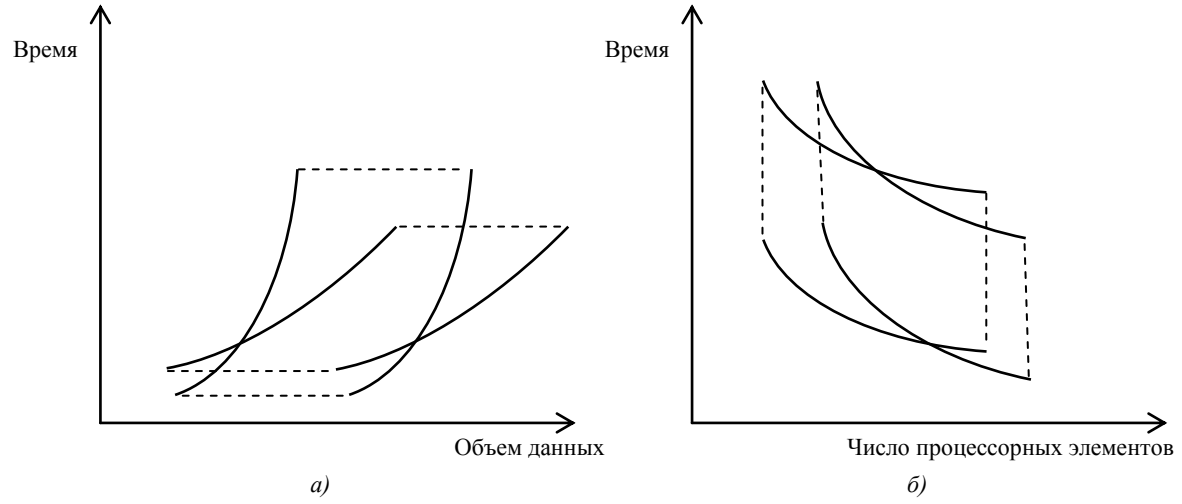
пространства признаков). Для упрощения рассуждений абстрагируемся от влияния разрядности и форм представления данных в вычислительной машине на объем вычислений.

Точность вычислений (в нашем случае качество, достоверность принятия решений) – это величина, которая определяется математическим методом и алгоритмом вычислений, репрезентативностью обучающей выборки, критерием принятия решений и т. п. Для упрощения рассуждений абстрагируемся от влияния разрядности и форм представления данных в вычислительной машине на точность вычислений.

Число процессорных элементов характеризует напрямую потенциальную возможность распараллеливания вычислительного процесса, а косвенно, как было отмечено выше, – стоимость вычислительной системы.

Время вычисления тестовой задачи (имеется в виду время кластеризации, классификации, ассоциации и т. п. или их композиции) определяется с момента подачи входных данных до получения результата. Этот параметр может измеряться как в абсолютных величинах, так и в условных – модельных тактах. В первом случае на оценку будут влиять технические характеристики аппаратной платформы (тактовая частота процессора, время доступа к данным в оперативной памяти и др.), во втором случае – способ определения модельного времени. Заметим, что производительность будет представлена не в гига-, терафлопсах и не в кило-, меганейронах, а в задачах кластеризации и (или) классификации и т. д. в единицу времени. Время будет складываться как из времени непосредственных вычислений на параллельном сопроцессоре, так и времени «накладных расходов», связанных с загрузкой либо выгрузкой данных сопроцессора, а также реализации участков программ, не поддающихся распараллеливанию, подготовительных операций, сервисных функций и т. п.

Влияние указанных параметров друг на друга будет характеризовать эффективность интеллектуальных вычислений. Графики качественных зависимостей такого влияния для различных аппаратных платформ показаны на рис. 2.



а) *б)*

Рис. 2. Качественные зависимости времени вычислений тестовой задачи: *а)* от объема обрабатываемых данных; *б)* от числа процессорных элементов

На рис. 2, *а* изображен вполне ожидаемый вид экспоненциальной зависимости, поскольку задачи интеллектуального анализа данных носят переборный характер. Однако характер наклона может существенно изменяться в зависимости от того, насколько качественно выполнено распараллеливание алгоритмов применительно к конкретной аппаратной платформе или насколько удачно спроектирована архитектура проблемно-ориентированной машины. На рис. 2, *б* показано проявление закона Амдаля – Густафсона, согласно которому производительность параллельной машины не может возрастать линейно с увеличением числа процессорных элементов. Между тем за счет предлагаемого подхода к построению когнитивных компьютеров наклон этой кривой можно значительно изменить (снизить в первом случае и повысить

во втором). Для обоих графиков полагается, что другие существенные параметры (точность, объем данных или число процессорных элементов) принимаются постоянными.

На рис. 2 одноименные кривые образуют некоторый диапазон пространства, отмеченный пунктиром, в котором может проходить конкретная кривая, а ее реальное положение будет определено техническими характеристиками аппаратной платформы. Важно отметить, что графики, характеризующие эффективность различных альтернативных платформ, могут пересекаться и это будет крайне полезной информацией при системном проектировании вычислительной машины в целом.

Заключение

В работе в качестве формальных задач Data Mining рассматриваются кластеризация, классификация, регрессия и ассоциативный поиск. В качестве исходных данных выступают образы, представленные в виде многомерных численных векторов информативных признаков. Для каждой из задач известен широкий перечень алгоритмов решения, которые имеют принципиальные отличия. Так, например, одни связаны с вычислением расстояний между образами в определенной метрике, другие – с аппроксимацией дискриминантной или регрессионной функций, третьи – с анализом и обработкой графов. Эти различия обусловливают очень узкую специфику применения известных нейрокомпьютеров и ассоциативных процессоров.

Предложен подход, который позволяет унифицировать программно-аппаратную реализацию задач регрессии, классификации, кластеризации и ассоциативного поиска посредством приведения различных алгоритмов к единой аппаратной платформе. В качестве аппаратной платформы может выступать как универсальная параллельная машина (например, многоядерная CPU, GPU, суперЭВМ), так и специализированная. В любом случае необходимо обеспечить эффективную реализацию библиотечных алгоритмов интеллектуальной обработки на параллельном сопроцессоре. Это открывает путь к созданию аппаратного вычислительного ядра, общего для широкого круга задач Data Mining и Knowledge Discovery, а следовательно, к созданию нового типа компьютеров. Предлагаемая модель вычислительной машины является более универсальной по сравнению с нейронными, ассоциативными, графовыми, семантическими и некоторыми другими, так как ориентирована на более высокий уровень представления формальных математических задач. Поэтому для ее определения использован термин «когнитивная», или «интеллектуальная», ЭВМ в связи с тем, что она охватывает широкий круг задач из области интеллектуального анализа данных.

Предложенный подход позволяет осуществить корректную сравнительную оценку альтернативных аппаратных средств, применяемых для построения интеллектуальных систем. Для этого в качестве функциональной единицы необходимо использовать алгоритм реализации одной из задач – кластеризации, классификации, ассоциативного поиска или их комбинации.

Список литературы

1. Data Mining: A Knowledge Discovery Approach / K.J. Cios [et al.]. – Springer, 2007.

2. Том, И.Э. Методы интеллектуального анализа многомерных данных для решения задач классификации / И.Э. Том, Н.А. Новоселова, О.В. Красько. – Минск : ОИПИ НАН Беларуси, 2011. – 233 с.

3. Кирсанов, Э.Ю. Нейрокомпьютеры с параллельной архитектурой / Э.Ю. Кирсанов. – М. : Радиотехника, 2004. – 221 с.

4. Аляутдинов, М.А. Нейрокомпьютеры: от программной к аппаратной реализации / М.А. Аляутдинов, А.И. Галушкин, П.А. Казанцев. – М. : Горячая линия – Телеком, 2008. – 152 с.

5. Реализация искусственных нейронных сетей в НТЦ «Модуль» // Компоненты и технологии. – 2005. – № 4. – С. 98–102.

6. Synthesis and Analysis of Classifiers Based on Generalized Model of Identification / D. Adzinets [et al.] //Advances in intelligent and soft computing. – Springer, 2010. – Vol. 71. – P. 529–536.

7. Tatur, M. Problem-Oriented Processors for the Solving of Classification Tasks / M. Tatur // J. of Information, Control and Management Systems. – 2013. – Vol. 11, no. 2. – P. 155–164.

8. Cognitive information processing based on a parallel processor / N. Verenik [et al.] // Proc. of 10th Intern. Conf. on Digital Technologies. – Žilina, 2014. – P. 367–371.

9. Wolfram Mathematica [Electronic resource]. – Mode of access : http://www.wolfram.com/mathematica. – Date of access : 12.01.2015.

10. Weka 3: Data Mining Software in Java [Electronic resource]. – Mode of access : http://www.cs.waikato.ac.nz/ml/weka. – Date of access : 12.01.2015.

Поступила 29.01.2015

Белорусский государственный университет
информатики и радиоэлектроники,
Минск, ул. П. Бровки, 6
e-mail: tatur@bsuir.by

M.M. Tatur

CONSTRUCTION PRINCIPLES OF COMPUTING UNITS FOR INTELLECTUAL DATA PROCESSING

An approach to the development of problem-oriented parallel computers for a wide range of tasks of intelligent data processing is described. A high level architecture is shown, where Data Mining algorithms are considered as functional elements and a general or specialized co-processor as an accelerator for computer performance. An accent has been made on the necessity to provide compatibility of the librarian algorithms and the architecture of the parallel co-processor. The technique for correct comparative evaluation of the alternative hardware platforms in the framework of the proposed approach is presented.

ИНФОРМАТИКА

2015 **январь-март** **№ 1**

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

УДК 519.872

С.А. Дудин, О.С. Дудина

МНОГОКАНАЛЬНАЯ СИСТЕМА ОБСЛУЖИВАНИЯ С МАРКОВСКИМ ВХОДНЫМ ПОТОКОМ НЕТЕРПЕЛИВЫХ ЗАПРОСОВ, ФУНКЦИОНИРУЮЩАЯ В СЛУЧАЙНОЙ СРЕДЕ

Исследуется многолинейная система массового обслуживания с бесконечным буфером и нетерпеливыми запросами, функционирующая в случайной среде. В систему поступает марковский входной поток запросов. Время обслуживания запроса имеет распределение фазового типа. В течение времени ожидания в буфере запросы могут проявлять нетерпеливость и покидать систему без обслуживания. Параметры системы зависят от состояния случайной среды. Находится условие существования стационарного режима. Приводятся формулы для вычисления основных характеристик производительности системы. Находится преобразование Лапласа – Стилтьеса распределения времен ожидания и пребывания запроса в системе.

Введение

Современные телекоммуникационные системы и сети связи являются сложным комплексом технических средств, обеспечивающих передачу разнообразных данных на различные расстояния с определенными параметрами качества. Телекоммуникационные системы и сети широко представлены во всех сферах жизнедеятельности человека, науки и техники. Основу телекоммуникационных систем составляют системы передачи данных по электрическим, волоконно-оптическим и радиоканалам. Область телекоммуникационных систем и сетей связи в настоящее время испытывает значительные преобразования, связанные с развитием новых технологий беспроводной передачи данных (в том числе в сенсорных сетях), технологий хранения и обработки информации (облачные вычисления), внедрением волоконно-оптической техники и т. д. Для развития современных телекоммуникационных систем требуется создание адекватных математических моделей их функционирования, которые учитывают особенности, присущие таким системам.

Известно, что на функционирование телекоммуникационных систем могут оказывать существенное воздействие случайные факторы, которые влияют на пропускную способность систем, характеристики входного потока требований, качество передачи данных и т. д. Примерами случайных факторов могут являться искусственные и естественные помехи, различные уровни шумов в канале передачи, изменение расстояния между мобильными передатчиками, параллельная передача информации высокого приоритета, различные сбои и поломки оборудования, влияние погодных условий. Особенно сильное влияние случайные факторы оказывают на беспроводные сети связи, которые интенсивно развиваются в последнее время. Учет влияния случайных факторов на процесс функционирования является крайне важным при построении адекватных математических моделей и расчете характеристик производительности телекоммуникационных систем.

Математические методы теории систем массового обслуживания, функционирующих в случайной среде, позволяют создавать адекватные стохастические модели современных телекоммуникационных систем, учитывающие влияние случайных факторов. Под случайной средой понимается случайный процесс с конечным пространством состояний, независимый от системы. При фиксированном состоянии случайной среды система функционирует как классическая система массового обслуживания. Однако параметры системы (входной поток, распределение времени обслуживания и др.) мгновенно изменяются с изменением состояния случайной среды. Обзор литературы по системам, функционирующим в случайной среде, приведен в статье [1].

В рассматриваемой в данной статье модели предполагается, что обслуживание запросов осуществляется конечным числом приборов. Входной поток запросов определяется с помощью марковского входного потока (MAP, от англ. Markovian arrival process), который позволяет учитывать коррелированный и взрывной характер входного трафика в современных телекоммуникационных сетях. Время обслуживания запроса каждым прибором имеет распределение фазового типа. Как известно, класс распределений фазового типа включает в себя многие традиционно используемые в теории массового обслуживания распределения и всюду плотен в классе всех распределений на неотрицательной полуоси. Вследствие этого произвольное распределение теоретически может быть с любой точностью аппроксимировано распределением фазового типа. Если в момент поступления запроса все приборы заняты, запрос направляется в буфер. Во время ожидания в буфере запросы могут проявлять нетерпеливость и покидать его.

1. Математическая модель

Рассматривается система массового обслуживания, которая состоит из N приборов и буфера бесконечного размера. Поведение системы зависит от состояния случайной среды. Случайная среда определяется стохастическим процессом $r_t, t \ge 0,$ который является неприводимой цепью Маркова с непрерывным временем, с пространством состояний $\{1,2,\ldots,R\}$ и инфинитезимальным генератором H.

Запросы поступают в соответствии с MAP-потоком, т. е. поступление запросов управляется стохастическим процессом $\nu_t, t \ge 0,$ с пространством состояний $\{0,1,\ldots,W\}$. При фиксированном состоянии случайной среды r этот процесс является неприводимой цепью Маркова с непрерывным временем. Время пребывания цепи в состоянии ν имеет экспоненциальное распределение с параметром $\lambda_\nu^{(r)}$. Когда время пребывания в состоянии ν истекло, процесс ν_t переходит в состояние ν' с вероятностью $p_0^{(r)}(\nu,\nu')$ без генерации запроса, а с вероятностью $p_1^{(r)}(\nu,\nu')$ с генерацией запроса, $\nu,\nu' = \overline{0,W},\ r = \overline{1,R}.$

Поведение MAP-потока полностью характеризуется квадратными матрицами $D_0^{(r)}$ и $D_1^{(r)}$ размера $\bar{W} = W+1,$ которые задаются следующим образом:

$$(D_0^{(r)})_{\nu,\nu} = -\lambda_\nu^{(r)},\ \nu = \overline{0,W},\ (D_0^{(r)})_{\nu,\nu'} = \lambda_\nu^{(r)} p_0^{(r)}(\nu,\nu'),\ \nu \ne \nu',\ \nu,\nu' = \overline{0,W},$$

и

$$(D_1^{(r)})_{\nu,\nu'} = \lambda_\nu^{(r)} p_1^{(r)}(\nu,\nu'),\ \nu,\nu' = \overline{0,W}.$$

Матрица $D^{(r)}(1) = D_0^{(r)} + D_1^{(r)}$ представляет собой генератор процесса $\nu_t,\ t \ge 0,$ при фиксированном состоянии среды $r,\ r = \overline{1,R}.$

Средняя интенсивность $\lambda^{(r)}$ поступления запросов при фиксированном состоянии случайной среды r определяется формулой

$$\lambda^{(r)} = \boldsymbol{\theta}^{(r)} D_1^{(r)} \boldsymbol{e},$$

где $\boldsymbol{\theta}^{(r)}$ – вектор стационарного распределения цепи Маркова $\nu_t,\ t \ge 0,$ при фиксированном состоянии среды $r,\ r = \overline{1,R}.$ Вектор $\boldsymbol{\theta}^{(r)}$ является единственным решением следующей системы линейных алгебраических уравнений:

$$\boldsymbol{\theta}^{(r)} D^{(r)}(1) = \mathbf{0},\ \boldsymbol{\theta}^{(r)} \boldsymbol{e} = 1.$$

Здесь и далее $\boldsymbol{e}$ – вектор-столбец, состоящий из единиц, а $\mathbf{0}$ – вектор-строка, состоящая из нулей.

Коэффициент вариации $c_{\text{var}}^{(r)}$ длин интервалов между моментами поступления запросов при фиксированном состоянии случайной среды r определяется формулой

$$c_{\text{var}}^{(r)} = 2\lambda^{(r)}\boldsymbol{\theta}^{(r)}(-D_0^{(r)})^{-1}\boldsymbol{e} - 1,\ r = \overline{1, R}.$$

Коэффициент корреляции $c_{\text{cor}}^{(r)}$ длин двух соседних интервалов при фиксированном состоянии случайной среды r вычисляется по формуле

$$c_{\text{cor}}^{(r)} = (\lambda^{(r)}\boldsymbol{\theta}^{(r)}(-D_0^{(r)})^{-1}D_1^{(r)}(-D_0^{(r)})^{-1}\boldsymbol{e} - 1) / c_{\text{var}}^{(r)},\ r = \overline{1, R}.$$

Более подробную информацию о MAP-потоке и его свойствах можно найти, например, в [2].

Запросы могут проявлять нетерпеливость, т. е. при фиксированном состоянии случайной среды r, $r = \overline{1, R}$, запрос уходит из системы через экспоненциально распределенное с параметром $\alpha^{(r)}$, $0 < \alpha^{(r)} < \infty$, время с момента попадания в буфер при условии, что он не попал на обслуживание.

Время обслуживания запроса прибором имеет распределение фазового типа (PH, от англ. phase type). Время обслуживания, имеющее распределение фазового типа, можно интерпретировать как время, в течение которого управляющий марковский процесс η_t, $t \geq 0$, с конечным пространством состояний $\{1, 2, \ldots, M, M+1\}$ достигнет единственное поглощающее состояние $M+1$ при условии, что при фиксированном состоянии случайной среды r начальное состояние этого процесса выбирается в множестве $\{1, 2, \ldots, M\}$ согласно стохастическому вектору-строке $\boldsymbol{\beta}^{(r)} = (\beta_1^{(r)}, \ldots, \beta_M^{(r)})$, $r = \overline{1, R}$. При фиксированном состоянии случайной среды интенсивности переходов процесса η_t в множестве состояний $\{1, 2, \ldots, M\}$ определяются субгенератором $S^{(r)}$, а интенсивности переходов в поглощающее состояние – элементами вектора-столбца $\boldsymbol{S}_0^{(r)} = -S^{(r)}\boldsymbol{e}$, $r = \overline{1, R}$. Среднее время обслуживания при фиксированном состоянии случайной среды r вычисляется по формуле $b_1^{(r)} = \boldsymbol{\beta}^{(r)}(-S^{(r)})^{-1}\boldsymbol{e}$, $r = \overline{1, R}$. Более подробное описание распределения фазового типа можно найти в [3].

2. Процесс изменения состояний системы и условие эргодичности

Примем следующие обозначения:

i_t – число запросов в системе, $i_t \geq 0$;

r_t – состояние случайной среды, $r_t = \overline{1, R}$;

ν_t – состояние управляющего процесса MAP-потока, $\nu_t = \overline{0, W}$;

$\eta_t^{(m)}$ – число приборов, в которых процесс обслуживания находится на фазе m, $m = \overline{1, M}$, $\eta_t^{(m)} = \overline{0, \min\{i_t, N\}}$, $\sum_{m=1}^{M}\eta_t^{(m)} = \min\{i_t, N\}$, в момент времени t, $t \geq 0$. Тогда поведение системы описывается неприводимой регулярной цепью Маркова с непрерывным временем:

$$\xi_t = \{i_t, r_t, \nu_t, \eta_t^{(1)}, \ldots, \eta_t^{(M)}\},\ t \geq 0.$$

Отметим, что значение компонентов $\eta_t^{(m)}$, $m = \overline{1, M}$, выбрано в соответствии с подходом Рамасвами и Лукантони [4].

Лемма 1. Инфинитезимальный генератор Q цепи Маркова ξ_t, $t \geq 0$, имеет блочно-трехдиагональную структуру:

$$Q = \begin{pmatrix} Q_{0,0} & Q_{0,1} & O & O & \dots \\ Q_{1,0} & Q_{1,1} & Q_{1,2} & O & \dots \\ O & Q_{2,1} & Q_{2,2} & Q_{2,3} & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Ненулевые блоки $Q_{i,j},\ i,\ j \ge 0,$ имеют вид

$$Q_{i,i} = H \otimes I_{\bar{W}K_i} + \tilde{D}_0^{(i)} + A_i + \Delta_i,\ i = \overline{0, N};$$

$$Q_{i,i} = H \otimes I_{\bar{W}K_N} + \tilde{D}_0^{(N)} + A_N + \Delta_N - E_{i-N},\ i > N;$$

$$Q_{i,i-1} = L_{N-i},\ i = \overline{1, N};$$

$$Q_{i,i-1} = \mathrm{diag}\{I_{\bar{W}} \otimes L_0(N, \tilde{S}^{(r)}) P_{N-1}(\boldsymbol{\beta}^{(r)}),\ r = \overline{1, R}\} + E_{i-N},\ i > N;$$

$$Q_{i,i+1} = \mathrm{diag}\{D_1^{(r)} \otimes P_i(\boldsymbol{\beta}^{(r)}),\ r = \overline{1, R}\},\ i = \overline{0, N-1};$$

$$Q_{i,i+1} = \tilde{D}_1^{(N)},\ i \ge N,$$

где I – единичная матрица, O – нулевая матрица соответствующего размера;
$\oplus$ и $\otimes$ – символы кронекеровых суммы и произведения матриц;

$K_i = C_{i+M-1}^{M-1},\ i = \overline{0, N};$

$\tilde{D}_l^{(i)} = \mathrm{diag}\{D_l^{(r)},\ r = \overline{1, R}\} \otimes I_{K_i},\ l = 0, 1,\ i = \overline{0, N};$

$A_i = \mathrm{diag}\{I_{\bar{W}} \otimes A_i(N, S^{(r)}),\ r = \overline{1, R}\},\ i = \overline{0, N};$

$\Delta_i = -\mathrm{diag}\{I_{\bar{W}} \otimes \Delta_i^{(r)},\ r = \overline{1, R}\},\ i = \overline{0, N},\ \Delta_0 = O_{\bar{W}(R+1)};$

$\Delta_i^{(r)} = \mathrm{diag}\{A_i(N, S^{(r)})\boldsymbol{e} + L_{N-i}(N, \tilde{S}^{(r)})\boldsymbol{e},\ r = \overline{1, R}\},\ i = \overline{0, N};$

$\tilde{S}^{(r)} = \begin{pmatrix} 0 & \mathbf{0} \\ \boldsymbol{S}_0^{(r)} & S^{(r)} \end{pmatrix},\ r = \overline{1, R};$

$E_k = \mathrm{diag}\{k\alpha^{(r)}, r = \overline{1, R}\} \otimes I_{\bar{W}K_N},\ k = \overline{1, K};$

$L_{N-i} = \mathrm{diag}\{I_{\bar{W}} \otimes L_{N-i}(N, \tilde{S}^{(r)}),\ r = \overline{1, R}\},\ i = \overline{1, N}.$

Детальное описание и алгоритмы для вычисления матриц $P_i(\boldsymbol{\beta}^{(r)}),\ i = \overline{0, N-1},\ A_i(N, S^{(r)})$ и $L_{N-i}(N, \tilde{S}^{(r)}),\ i = \overline{0, N},\ r = \overline{1, R},$ приведены в работе [5].

Доказательство леммы опирается на анализ переходов цепи Маркова $\xi_t,\ t \ge 0,$ за бесконечно малый интервал времени с последующей группировкой интенсивностей соответствующих переходов в блоки матрицы Q.

Замечание. Цепь Маркова $\xi_t,\ t \ge 0,$ принадлежит классу асимптотически квазителицевых цепей Маркова с непрерывным временем [6].

Как следует из [6], необходимым условием существования стационарного распределения асимптотически квазителицевой цепи Маркова $\xi_t,\ t \ge 0,$ является выполнение неравенства

$$\boldsymbol{y}Y_0\boldsymbol{e} > \boldsymbol{y}Y_2\boldsymbol{e}, \qquad (1)$$

где вектор-строка $\boldsymbol{y}$ – решение системы линейных алгебраических уравнений

$$\boldsymbol{y}(Y_0 + Y_1 + Y_2) = \boldsymbol{y},\ \ \boldsymbol{y}\boldsymbol{e} = 1,$$

матрицы Y_0, Y_1 и Y_2 определяются как

$$Y_0 = \lim_{i\to\infty} R_i^{-1} Q_{i,i-1},\ Y_1 = \lim_{i\to\infty} R_i^{-1} Q_{i,i} + I,\ Y_2 = \lim_{i\to\infty} R_i^{-1} Q_{i,i+1},$$

а матрица R_i является диагональной матрицей, диагональные элементы которой определяются как модули соответствующих диагональных элементов матрицы $Q_{i,i}$, $i \ge 0$.

Легко убедиться, что в рассматриваемом случае матрицы Y_0, Y_1 и Y_2 имеют следующий вид:

$$Y_1 = (\Phi_{r,r'}),\ r, r' = \overline{1, R},\ Y_2 = \mathrm{diag}\{Z_1, \ldots, Z_R\},\ Y_0 = \mathrm{diag}\{\Omega_1, \ldots, \Omega_R\},$$

где

$$\Phi_{r,r'} = \begin{cases} R_r[(H)_{r,r} I_{\bar{W}K_N} + D_0^{(r)} \otimes I_{K_N} + I_{\bar{W}} \otimes (A_N(N, S^{(r)}) - \Delta_N^{(r)})] + I_{\bar{W}K_N}, \ \alpha = 0,\ r = r', \\ (H)_{r,r'} R_r,\ \alpha = 0,\ r \ne r',\ \ r, r' = \overline{1, R}, \\ O,\ \alpha > 0; \end{cases}$$

$$Z_r = \begin{cases} R_r D_1^{(r)},\ \alpha = 0, \\ O,\ \alpha > 0, \end{cases} \quad r = \overline{1, R};$$

$$\Omega_r = \begin{cases} R_r (I_{\bar{W}} \otimes L_0(N, \tilde{S}^{(r)}) P_{N-1}(\boldsymbol{\beta}^{(r)})),\ \alpha = 0, \\ I,\ \alpha > 0, \end{cases} \quad r = \overline{1, R};$$

$$R_r = (-(H)_{r,r} I_{\bar{W}K_N} + \Sigma^{(r)} \otimes I_{K_N} + I_{\bar{W}} \otimes \mathrm{diag}\{L_0(N, \tilde{S}^{(r)})\boldsymbol{e},\ r = \overline{1, R}\})^{-1}.$$

Здесь $\Sigma^{(r)}$ – диагональная матрица, диагональные элементы которой являются соответствующими диагональными элементами матрицы $-D_0^{(r)}$.

Теорема 1. *Если запросы являются нетерпеливыми (т. е.* $\alpha^{(r)} > 0$*) хотя бы для одного состояния случайной среды* r, $r = \overline{1, R}$, *цепь Маркова* ξ_t *является эргодической при любых параметрах системы.*

Доказательство. Пусть $L = \{l_1, l_2, \ldots, l_m\}$ – множество состояний среды, для которых $\alpha^{(l)} > 0$, $l \in L$. Можно убедиться, что в данном случае матрица $Y = Y_0 + Y_1 + Y_2$ приводимая и путем согласованной перестановки строк и столбцов может быть представлена в виде

$$Y = \begin{pmatrix} Y_{1,1} & Y_{1,2} \\ O & I \end{pmatrix},$$

где $Y_{1,1}$ – матрица, полученная из матрицы Y путем отбрасывания блочных строк и столбцов с номерами l, $l \in L$; $Y_{1,2}$ – матрица, полученная из матрицы Y путем отбрасывания блочных строк с номерами l, $l \in L$, и столбцов с номерами r, $r = \overline{1, R}$, $r \notin L$.

Как следует из [6], условие эргодичности (1) может быть переписано в виде

$$\boldsymbol{x}\bar{Y}_0\boldsymbol{e} > \boldsymbol{x}\bar{Y}_2\boldsymbol{e}, \tag{2}$$

где вектор $\boldsymbol{x}$ является решением системы линейных алгебраических уравнений

$$\boldsymbol{x}I = \boldsymbol{x},\ \boldsymbol{x}\boldsymbol{e} = 1,$$

а матрицы $\bar{Y}_0$ и $\bar{Y}_2$ получены из матриц Y_0 и Y_2 путем отбрасывания блочных строк и столбцов с номерами r, $r \in R / L$. Видно, что матрица $\bar{Y}_2$ является нулевой, а $\bar{Y}_0$ – единичной. Очевидно, что условие (2) выполняется всегда:

$$\boldsymbol{x}\bar{Y}_0\boldsymbol{e} = \boldsymbol{x}I\boldsymbol{e} = \boldsymbol{x}\boldsymbol{e} = 1 > 0 = \boldsymbol{x}\bar{Y}_2\boldsymbol{e}.$$

Рассмотрим случай, когда запросы являются абсолютно терпеливыми при всех состояниях среды, т. е. $\alpha^{(r)} = 0$, $r = \overline{1, R}$. В этом случае блоки генератора имеют следующий вид:

$$Q_{i,i} = Q_1 = H \otimes I_{\bar{W}K_N} + \tilde{D}_0^{(N)} + A_N + \Delta_N,\ i > N;$$

$$Q_{i,i+1} = Q_2 = \tilde{D}_1^{(N)},\ i \geq N;$$

$$Q_{i,i-1} = Q_0 = \mathrm{diag}\{I_{\bar{W}} \otimes L_0(N, \tilde{S}^{(r)})P_{N-1}(\boldsymbol{\beta}^{(r)}),\ r = \overline{1, R}\},\ i > N.$$

Таким образом, блоки генератора не зависят от переменной i при $i > N$ и цепь Маркова ξ_t, $t \geq 0$, принадлежит классу квазитеплицевых цепей Маркова с непрерывным временем или цепей Маркова типа $M / G / 1$ [3].

Как следует из [3], необходимым и достаточным условием эргодичности квазитеплицевых цепей Маркова является выполнение неравенства

$$\boldsymbol{\varphi}Q_0\boldsymbol{e} > \boldsymbol{\varphi}Q_2\boldsymbol{e},$$

где вектор $\boldsymbol{\varphi}$ – единственное решение системы линейных алгебраических уравнений

$$\boldsymbol{\varphi}(Q_0 + Q_1 + Q_2) = \boldsymbol{0},\ \boldsymbol{\varphi}\boldsymbol{e} = 1.$$

Далее считаем, что условие эргодичности рассматриваемой системы выполняется. Тогда существуют следующие пределы (стационарные вероятности):

$$\pi(i, r, \nu, \eta^{(1)}, \ldots, \eta^{(M)}) = \lim_{t \to \infty} P\{i_t = i,\ r_t = r,\ \nu_t = \nu,\ \eta_t^{(1)} = \eta^{(1)}, \ldots, \eta_t^{(M)} = \eta^{(M)}\},$$

$$i \geq 0,\ r = \overline{1, R},\ \nu = \overline{0, W},\ \eta^{(m)} = \overline{0, \min\{i, N\}},\ m = \overline{1, M},\ \sum_{m=1}^{M} \eta^{(m)} = \min\{i, N\}.$$

Сформируем векторы $\boldsymbol{\pi}(i, r, \nu)$ из вероятностей $\pi(i, r, \nu, \eta^{(1)}, \ldots, \eta^{(M)})$, перенумерованных в обратном лексикографическом порядке компонентов $\eta^{(1)}, \ldots, \eta^{(M)}$. Далее сформируем векторы

$$\boldsymbol{\pi}(i, r) = (\boldsymbol{\pi}(i, r, 0), \boldsymbol{\pi}(i, r, 1), \ldots, \boldsymbol{\pi}(i, r, W)),\ r = \overline{1, R};$$

$$\boldsymbol{\pi}_i = (\boldsymbol{\pi}(i, 1), \boldsymbol{\pi}(i, 2), \ldots, \boldsymbol{\pi}(i, R)),\ i \geq 0.$$

Известно, что векторы $\boldsymbol{\pi}_i$, $i \geq 0$, являются единственным решением следующей системы линейных алгебраических уравнений:

$$(\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \ldots)Q = \boldsymbol{0},\ (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \ldots)\boldsymbol{e} = 1,$$

где Q – генератор цепи Маркова ξ_t, $t \geq 0$.

Для решения данной системы может быть использован специальный устойчивый алгоритм, приведенный в работе [7].

3. Характеристики производительности

Найдя векторы стационарных вероятностей $\boldsymbol{\pi}_i$, $i \geq 0$, можно вычислить различные характеристики производительности системы:

– среднее число запросов в системе

$$\bar{L} = \sum_{i=1}^{\infty} i\boldsymbol{\pi}_i \boldsymbol{e};$$

– среднее число запросов в буфере

$$N^{\text{buffer}} = \sum_{i=N+1}^{\infty} (i-N)\boldsymbol{\pi}_i \boldsymbol{e};$$

– среднее число занятых приборов

$$N^{\text{server}} = \sum_{i=1}^{\infty} \min\{i, N\}\boldsymbol{\pi}_i \boldsymbol{e};$$

– интенсивность выходного потока обслуженных запросов

$$\lambda^{\text{out}} = \sum_{i=1}^{\infty} \boldsymbol{\pi}_i \operatorname{diag}\{I_{\bar{W}} \otimes L_{\max\{N-i,0\}}(N, \tilde{S}^{(r)}),\ r = \overline{1, R}\}\boldsymbol{e};$$

– вероятность потери произвольного запроса

$$P^{\text{loss}} = 1 - \frac{\lambda^{\text{out}}}{\lambda},$$

где средняя интенсивность входного потока λ вычисляется по формуле

$$\lambda = \boldsymbol{\theta} \operatorname{diag}\{D_1^{(r)},\ r = \overline{1, R}\}\boldsymbol{e},$$

а вектор $\boldsymbol{\theta}$ является единственным решением следующей системы линейных алгебраических уравнений:

$$\boldsymbol{\theta}(H \otimes I_{\bar{W}} + \operatorname{diag}\{D_0^{(r)} + D_1^{(r)},\ r = \overline{1, R}\}) = \mathbf{0},\ \boldsymbol{\theta}\boldsymbol{e} = 1.$$

4. Распределение времени пребывания произвольного запроса в системе

Пусть $V(x)$ – функция распределения времени пребывания произвольного запроса в системе и $v(s) = \int_0^{\infty} e^{-sx} dV(x)$, $\operatorname{Re} s > 0$, – ее преобразование Лапласа – Стилтьеса.

Пометим произвольный запрос и будем отслеживать его пребывание в системе. Выведем выражение для преобразования Лапласа – Стилтьеса $v(s)$ с помощью метода коллективных отметок (метода введения дополнительного события, метода катастроф) [8, 9]. С этой целью интерпретируем переменную s как интенсивность воображаемого стационарного пуассоновского потока катастроф. Таким образом, $v(s)$ означает вероятность того, что катастрофа не наступит в течение времени пребывания помеченного запроса.

Пусть $y(s, r, m)$ – вероятность того, что катастрофа не наступит в течение оставшейся части времени обслуживания помеченного запроса в системе при условии, что в данный момент состояние случайной среды есть r, $r = \overline{1, R}$, а фаза обслуживания – m, $m = \overline{1, M}$.

Сформируем векторы

$$\boldsymbol{y}(s, r) = (y(s, r, 1), \ldots, y(s, r, M))^T, \ r = \overline{1, R};$$

$$\boldsymbol{y}(s) = ((\boldsymbol{y}(s, 1))^T, \ldots, (\boldsymbol{y}(s, R))^T)^T$$

и введем следующие обозначения:

$$\tilde{S} = \mathrm{diag}\{S^{(r)}, \ r = \overline{1, R}\};$$

$$\boldsymbol{S}_0 = ((\boldsymbol{S}_0^{(1)})^T, \ldots, (\boldsymbol{S}_0^{(R)})^T)^T.$$

Лемма 2. Вектор $\boldsymbol{y}(s)$ вычисляется по формуле

$$\boldsymbol{y}(s) = (-\tilde{S} - H \otimes I_M + sI)^{-1} \boldsymbol{S}_0.$$

Доказательство. Основываясь на вероятностном смысле преобразования Лапласа – Стилтьеса и формуле полной вероятности, можно показать, что вероятности $y(s, r, m)$ удовлетворяют следующей системе линейных алгебраических уравнений:

$$y(s, r, m) = (-(S^{(r)})_{m, m} - (H)_{r, r} + s)^{-1}[(\boldsymbol{S}_0^{(r)})_m + \sum_{r'=1, \ r' \neq r}^{R} (H)_{r, r'} y(s, r', m) + \\ + \sum_{m'=1, m' \neq m}^{M} (S^{(r)})_{m, m'} y(s, r, m')], \ r = \overline{1, R}, \ m = \overline{1, M}. \tag{3}$$

Используя обозначения, введенные выше, перепишем систему (3) в матричной форме:

$$(\tilde{S} + H \otimes I_M - sI)\boldsymbol{y}(s) = -\boldsymbol{S}_0.$$

Отсюда непосредственно следует утверждение леммы, так как матрица $\tilde{S} + H \otimes I_M$ является субгенератором, т. е. у матрицы $\tilde{S} + H \otimes I_M - sI$ существует обратная.

Пусть $w(s, l, r, \eta^{(1)}, \ldots, \eta^{(M)})$ – вероятность того, что катастрофа не наступит в течение оставшейся части времени пребывания помеченного запроса в системе при условии, что в данный момент помеченный запрос имеет положение l, $l > 0$, в буфере, состояние случайной среды есть r, $r = \overline{1, R}$, и состояние процессов обслуживания есть $\eta^{(1)}, \ldots, \eta^{(M)}$.

Перенумеруем вероятности $w(s, l, r, \eta^{(1)}, \ldots, \eta^{(M)})$ в обратном лексикографическом порядке компонентов $\eta^{(1)}, \ldots, \eta^{(M)}$ и сформируем из этих вероятностей векторы-столбцы $\boldsymbol{w}(s, l, r)$.

Векторы $\boldsymbol{w}(s, l, r)$, $l > 0$, $r = \overline{1, R}$, могут быть найдены из следующей системы линейных алгебраических уравнений:

$$\boldsymbol{w}(s, l, r) = ((s + l\alpha^{(r)} - (H)_{r, r})I_{T_N} - A^{(r)})^{-1}(\delta_{l, 1} L_0(N, \tilde{S}^{(r)})\boldsymbol{e}\boldsymbol{\beta}^{(r)} \boldsymbol{y}(s, r) + \\ + (1 - \delta_{l, 1})(L^{(r)} + (l-1)\alpha^{(r)} I_{T_N})\boldsymbol{w}(s, l-1, r) + \sum_{r'=1, \ r' \neq r}^{R} (H)_{r, r'} \boldsymbol{w}(s, l, r') + \alpha^{(r)} \boldsymbol{e}_{T_N}), \tag{4}$$

где $\delta_{i, j} = 0$ при $i = j$ и $\delta_{i, j} = 1$ в противном случае,

$$L^{(r)} = L_0(N, \tilde{S}^{(r)})P_{N-1}(\boldsymbol{\beta}^{(r)});$$

$$A^{(r)} = A_N(N, S^{(r)}) - \mathrm{diag}\{A_N(N, S^{(r)})\boldsymbol{e} + L_0(N, \tilde{S}^{(r)})\boldsymbol{e},\ r = \overline{1, R}\},\ r = \overline{1, R}.$$

Чтобы найти решение системы (4), введем векторы-столбцы

$$\boldsymbol{w}(s, l) = ((\boldsymbol{w}(s, l, 1))^T, \ldots, (\boldsymbol{w}(s, l, R))^T)^T$$

и перепишем систему (3) в матричном виде

$$(-sI + A - lE + H \otimes I_{T_N})\boldsymbol{w}(s, l) +$$

$$+\delta_{l,1}L\boldsymbol{e}\boldsymbol{\beta}\boldsymbol{y}(s) + (1-\delta_{l,1})(L + (l-1)E)\boldsymbol{w}(s, l-1) + E\boldsymbol{e} = \boldsymbol{0}^T,\ l > 0,$$

где

$$A = \mathrm{diag}\{A^{(r)},\ r = \overline{1, R}\};$$

$$E = \mathrm{diag}\{\alpha^{(r)}I_{T_N},\ r = \overline{1, R}\};$$

$$L = \mathrm{diag}\{L^{(r)},\ r = \overline{1, R}\};$$

$$\boldsymbol{\beta} = \mathrm{diag}\{\boldsymbol{\beta}^{(r)},\ r = \overline{1, R}\}.$$

Векторы $\boldsymbol{w}(s, l),\ l > 0,$ можно вычислить по рекуррентным формулам

$$\boldsymbol{w}(s, 1) = (-A + E - H \otimes I_{T_N} + sI)^{-1}(L\boldsymbol{e}\boldsymbol{\beta}\boldsymbol{y}(s) + E\boldsymbol{e})^T;$$

$$\boldsymbol{w}(s, l+1) = (-A + (l+1)E - H \otimes I_{T_N} + sI)^{-1}[E\boldsymbol{e} - (L + lE)\boldsymbol{w}(s, l)]^T,\ l > 1.$$

После нахождения векторов $\boldsymbol{w}(s, l),\ l > 1,$ получим следующий результат.

Теорема 2. *Преобразование Лапласа – Стилтьеса $v(s)$ распределения времени пребывания произвольного запроса в системе вычисляется как*

$$v(s) = P^{\mathrm{loss}} + \lambda^{-1}[\sum_{i=0}^{N-1}\sum_{r=1}^{R}\boldsymbol{\pi}(i, r)(D_1^{(r)} \otimes I_{T_i})\boldsymbol{e}\boldsymbol{\beta}^{(r)}\boldsymbol{y}(s, r) + \sum_{i=N}^{\infty}\sum_{r=1}^{R}\boldsymbol{\pi}(i, r)(D_1^{(r)}\boldsymbol{e} \otimes I_{T_N})\boldsymbol{w}(s, i-N+1, r)].$$

Доказательство основывается на вероятностном смысле преобразования Лапласа – Стилтьеса и формуле полной вероятности.

Следствие 1. *Среднее время пребывания V_{soj} произвольного запроса рассчитывается как*

$$V_{\mathrm{soj}} = -v'(s)|_{s=0} = -\lambda^{-1}[\sum_{i=0}^{N-1}\sum_{r=1}^{R}\boldsymbol{\pi}(i, r)(D_1^{(r)} \otimes I_{T_i})\boldsymbol{e}\boldsymbol{\beta}^{(r)}\frac{\partial \boldsymbol{y}(s, r)}{\partial s}|_{s=0} +$$

$$+\sum_{i=N}^{\infty}\sum_{r=1}^{R}\boldsymbol{\pi}(i, r)(D_1^{(r)}\boldsymbol{e} \otimes I_{T_N})\frac{\partial \boldsymbol{w}(s, i-N+1, r)}{\partial s}|_{s=0}].$$

Теорема 3. *Преобразование Лапласа – Стилтьеса $z(s)$ распределения времени ожидания произвольного запроса в системе вычисляется как*

$$z(s) = P^{\mathrm{loss}} + \lambda^{-1}[\sum_{i=0}^{N-1}\sum_{r=1}^{R}\boldsymbol{\pi}(i, r)(D_1^{(r)} \otimes I_{T_i})\boldsymbol{e} + \sum_{i=N}^{\infty}\sum_{r=1}^{R}\boldsymbol{\pi}(i, r)(D_1^{(r)}\boldsymbol{e} \otimes I_{T_N})\boldsymbol{z}(s, i-N+1, r)],$$

где векторы $\boldsymbol{z}(s,l,r),\ l>0,$ *являются подвекторами векторов* $\boldsymbol{z}(s,l),\ l>0,$ *которые вычисляются по рекуррентным формулам*

$$\boldsymbol{z}(s,1)=(-A+E-H\otimes I_{T_N}+sI)^{-1}(L\boldsymbol{e}+E\boldsymbol{e})^T;$$
$$\boldsymbol{z}(s,l+1)=(-A+(l+1)E-H\otimes I_{T_N}+sI)^{-1}[E\boldsymbol{e}-(L+lE)\boldsymbol{z}(s,l)]^T,\ l>1.$$

Следствие 2. *Среднее время ожидания* V_{wait} *произвольного запроса рассчитывается как*

$$V_{\text{wait}}=-z'(s)|_{s=0}=-\lambda^{-1}\sum_{i=N}^{\infty}\sum_{r=1}^{R}\boldsymbol{\pi}(i,r)(D_1^{(r)}\boldsymbol{e}\otimes I_{T_N})\frac{\partial\boldsymbol{z}(s,i-N+1,r)}{\partial s}|_{s=0}.$$

Заключение

В статье изучена многолинейная система обслуживания с марковским входным потоком нетерпеливых запросов и фазовым распределением времени обслуживания, функционирующая в случайной среде. Рассмотрен процесс изменения состояний системы, найдено условие эргодичности, приведены формулы для нахождения основных характеристик производительности, включая средние времена пребывания и ожидания произвольного запроса в системе. Результаты исследования могут применяться для оценивания производительности и оптимизации функционирования телекоммуникационных систем и сетей связи, на функционирование которых оказывают влияние случайные факторы различной природы.

Работа выполнена при частичной поддержке Белорусского республиканского фонда фундаментальных исследований, проект № Ф14МВ-001.

Список литературы

1. Erlang loss queueing system with batch arrivals operating in a random environment / C.S. Kim [et al.] // Computers & Operations Research. – 2009. – Vol. 36, № 3. – P. 674–697.

2. He, Q.M. Queues with marked customers / Q.M. He // Advances in Applied Probability. – 1996. – Vol. 28. – P. 567–587.

3. Neuts, M. Matrix-geometric solutions in stochastic models – an algorithmic approach / M. Neuts. – Johns Hopkins University Press, 1981. – 332 p.

4. Ramaswami, V. Algorithms for the multi-server queue with phase-type service / V. Ramaswami, D.M. Lucantoni // Comm. Statist.-Stochastic Models. – 1985. – Vol. 1. – P. 393–417.

5. Queueing system MMAP/PH/N/N+R with impatient heterogeneous customers as a model of call center / C.S. Kim [et al.] // Applied Mathematical Modelling. – 2013. – Vol. 37, № 3. – P. 958–976.

6. Klimenok, V.I. Multi-dimensional asymptotically quasi-Toeplitz Markov chains and their application in queueing theory / V.I. Klimenok, A.N. Dudin // Queueing Systems. – 2006. – Vol. 54. – P. 245–259.

7. Dudina O. Retrial Queueing System with Markovian Arrival Flow and Phase Type Service Time Distribution / O. Dudina, Ch. Kim, S. Dudin // Computers and Industrial Engineering. – 2013. – Vol. 66. – P. 360–373.

8. Kesten, H. Priority in waiting line problems / H. Kesten, J.Th. Runnenburg. – Amsterdam : Mathematisch Centrum, 1956. – 234 p.

9. Danzig, van D. Chaines de Markof dans les ensembles abstraits et applications aux processus avec regions absorbantes et au probleme des boucles / D. van Danzig // Ann. de l'Inst. H. Pioncare. – 1955. – Vol. 14. – P. 145–199.

Поступила 09.12.2014

Белорусский государственный университет,
Минск, пр. Независимости, 4
e-mail: dudin85@mail.ru,
dudina_olga@email.com

S.A. Dudin, O.S. Dudina

MULTISERVER QUEUEING SYSTEM WITH MARKOVIAN ARRIVAL FLOW OF IMPATIENT CUSTOMERS OPERATING IN A RANDOM ENVIRONMENT

Multiserver queuing system with an infinite buffer and impatient customers, operating in a random environment is investigated. Customers arrive to the system according to the Markovian arrival flow. Service time of a customer has a phase type distribution. During the waiting time in the buffer customers can be impatient and leave the system forever. The system parameters depend on the state of the random environment. The ergodicity condition is derived. The formulas for calculating the main performance measures of the system are obtained. The Laplace-Stieltjes transforms of waiting and sojourn times of a customer in the system are calculated.

УДК 699.86:519.85

Е.В. Кресова[1], С.П. Кундас[2]

ТЕПЛОВАЯ МОДЕЛЬ ИНДИВИДУАЛЬНОГО ЖИЛОГО ДОМА

Предлагается трехмерная тепловая модель дома для сельской местности, которая включает конструктивную, конечно-элементную и расчетно-аналитическую модели, созданные в программных комплексах SolidWorks (конструктивная модель) и COMSOL Multiphysics (конечно-элементная и расчетно-аналитическая модели). Для моделирования процесса переноса тепла во всем объеме здания создается дополнительный объект – воздушная среда помещения. Показывается работоспособность модели и возможность ее применения для решения задач оптимизации ограждающих конструкций зданий.

Введение

Отапливаемый объект представляет собой сложную архитектурно-конструктивную систему с многообразием составляющих ее энергетически взаимосвязанных элементов: ограждений, окон, нагревательных приборов, оборудования помещений, воздушной среды (наружной и внутренней), бытовых и производственных теплопоступлений. В рассматриваемой системе протекают различные по физической сущности процессы поглощения, превращения и переноса теплоты.

При анализе теплового режима учитываются наиболее существенные характеристики конструкции и протекающие физические процессы, т. е. создается в определенной степени идеализированная тепловая модель. Основное требование к тепловой модели может быть сформулировано следующим образом: тепловая модель должна быть адекватна изучаемым явлениям и реализуема математически [1].

Одним из важных требований к математической модели является обеспечение быстродействия при ее компьютерной реализации, что может быть достигнуто упрощением модели или применением высокопроизводительных вычислительных средств. Последнее в реальных производственных условиях не всегда доступно. Поэтому целью настоящей работы было создание прикладной тепловой модели здания с использованием возможностей доступных коммерческих программных средств, которая обеспечивает проведение вычислений с достаточной для практического применения точностью [2], и ее применение для анализа и оптимизации теплоизоляционных материалов ограждающих конструкций зданий на этапе их проектирования.

1. Методика создания тепловой модели здания

Процесс переноса тепла в теле с учетом конвекции можно описать уравнением теплопроводности [3, 4]

$$\rho c_p \frac{\partial T}{\partial \tau} Q + \rho c_p \vec{u} \nabla T = \nabla(\lambda \nabla T) + Q, \quad (1)$$

где c_p – теплоемкость материала при постоянном давлении, Дж/(кг · °С);

ρ – плотность материала, кг/м^3;

T – температура, °С;

u – поле скоростей, м/с;

Q – мощность внутренних источников теплоты, Вт/м^3;

λ – коэффициент теплопроводности материала, Вт/м · °С;

τ – время, с.

Выражение (1) устанавливает связь между временным и пространственным изменениями температуры в любой точке тела, в котором происходит процесс теплопроводности. Это уравнение описывает процесс, протекающий в телах, которые обладают следующими свойствами [3]:

– тело однородно и изотропно;

– деформация рассматриваемого объема, связанная с изменением температуры, очень мала по сравнению с самим объемом;

– внутренние источники теплоты в теле, которые в общем случае могут быть заданы как $q_{\text{п}} = f(x, y, z, \tau)$, распределены равномерно [5].

Уравнение (1) является основой тепловой математической модели, реализованной в программном комплексе COMSOL Multiphysics. Для практического использования указанной модели были дополнительно разработаны конструктивная, конечно-элементная и расчетная модели анализируемого дома, сформулированы начальные и граничные условия.

Конструктивная модель. В качестве объекта исследований выбран строящийся энергоэффективный дом учебно-научного комплекса (УНК) «Волма» Международного государственного экологического университета им. А.Д. Сахарова [6].

Для заполнения стен в качестве внутреннего утеплителя в типовом проекте дома использовалась специально приготовленная смесь из глины, щепы и воды. Эта смесь производится непосредственно на строительной площадке. По своим теплофизическим характеристикам она приближается к арболиту с плотностью около 500 кг/м^3. Арболит представляет собой разновидность легкого бетона, изготавливаемого из подобранной смеси цемента, органических заполнителей, химических добавок и воды. Физико-технические показатели стеновых блоков из арболита согласно СТБ 1105–98 [7]:

средняя плотность в сухом состоянии – не более 550 кг/м^3;

класс по прочности на сжатие – В 0,35;

теплопроводность – 0,11 Вт/(м·К).

Конструктивная модель типового дома (рис. 1) разработана в программном комплексе SolidWorks и отображает пространственные геометрические соотношения между элементами конструкции объекта.

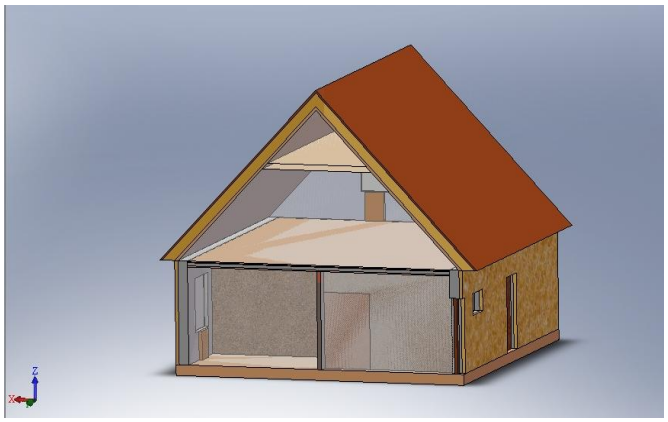


Рис. 1. Конструктивная модель дома

В предложенной модели был введен дополнительный объект – воздушная среда внутри здания, что позволяет избежать задания фиксированной температуры на внутренней границе ограждающей конструкции как граничного условия.

Конечно-элементная модель. На основе конструктивной модели в программном комплексе COMSOL Multiphysics разработана конечно-элементная модель дома, которая состоит из сетки тетраэдров. При построении сетки учитывалось, что между любыми двумя границами будет не меньше 10 конечных элементов.

Неравномерная тетраэдрическая сетка в физическом пространстве должна сгущаться в тех областях, в которых перенос тепла представляет наибольший интерес, и у искривленных поверхностей, чтобы лучше повторять их геометрию.

В расчетно-аналитической модели к граням или к узлам конечно-элементной сетки элементов прикладываются нагрузки и устанавливаются ограничения. Также в данной модели описываются свойства применяемых материалов.

2. Компьютерная реализация предложенной модели

Компьютерная реализации модели осуществлялась в программе COMSOL Multiphysics. Использован модуль *Heat Transfer*, который описывает процессы переноса тепла в одно-, двух- и трехмерной реализации, т. е. математической модели.

Процедура расчета включает следующие этапы:

– импорт созданной в SolidWorks конструктивной модели дома в программу COMSOL Multiphysics;

– наложение на конструктивную модель неравномерной тетраэдрической конечно-элементной сетки;

– приложение к конечно-элементной модели нагрузок и установление ограничений;

– проведение расчетов.

Для анализа тепловых режимов дома с учетом всех заданных параметров (изменения температуры атмосферного воздуха, конвекции воздуха в помещении) необходим большой объем оперативной памяти. Предварительные исследования показали, что при использовании доступных вычислительных средств (*HP Pavilion*) для проведения анализа тепловых режимов дома требуется большой объем оперативной памяти. Поэтому для ускорения процесса вычислений в качестве объекта моделирования была выбрана одна комната анализируемого дома.

Для решения нестационарного переноса тепла в программе COMSOL Multiphysics был выбран модуль Direct (SPOOLES), Time Dependent.

3. Результаты моделирования и их анализ

Для конструкции утепления стеновых панелей (рис. 2 и таблица) с помощью разработанной модели выполнен расчет изменения температуры внутри дома и в сечении стены дома. При проектировании теплоизоляционной оболочки здания на основе многослойных конструкций учитывалось, что слои из тех материалов, которые имеют более высокую теплопроводность и теплоемкость, располагаются с внутренней стороны конструкции, а слои из материалов, которые имеют более низкие соответствующие показатели, – с внешней [8].

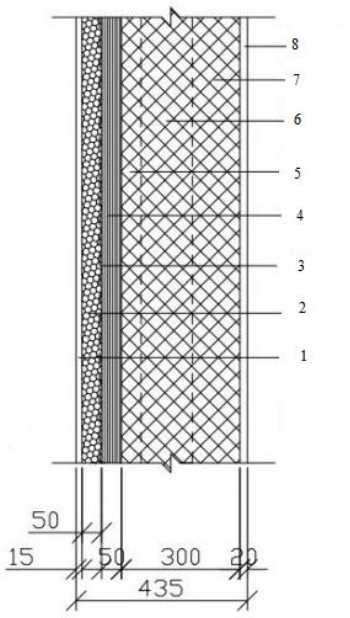


Рис. 2. Конструкция стены дома: 1) армированная штукатурка смесями «Забудова»; 2) плита тростниковая 50 мм (горизонтальное расположение стебля); 3) пароизоляционная пленка; 4) плита тростниковая 50 мм (вертикальное расположение стебля); 5) брусок 50х50 выносной (сосна); 6) заполнение смесью глины и щепы (300 мм); 7) стойка каркаса 120х120 мм (сосна); 8) штукатурка глиняным раствором

Для упрощения модели было рассчитано, какой процент занимают деревянные стойки каркаса от всего объема ограждающей конструкции. Вычисления показали, что эта величина составляет примерно 4 %. Это свидетельствует о незначительном влиянии стоек на тепловое сопротивление ограждающих конструкций и позволяет не учитывать их при моделировании.

Характеристики материалов

Материал	Коэффициент теплопроводности λ, Вт/ м °С	Теплоемкость c_p, Дж/кг °С	Плотность ρ, кг/м^3
Тростник	0,067	2300	120
Древесина (сосна)	0,35	2300	500
Щепа и глина	0,11	1200	500
Цементно-песчаный раствор	0,58	840	1800
Минеральная вата	0,044	840	125

Рассмотрим результаты моделирования динамики изменения температуры внутри комнаты при включении отопления. Комната, площадь которой 13,332 м^2, ограничивается внутренними стенами, внешней стеной с окном и имеет источник тепла (батарею), мощность которого остается постоянной на протяжении всего периода моделирования (Q = const). Мощность батареи принималась из условий, что на 1 м^3 необходим 41 Вт тепловой мощности [9].

Для моделирования были заданы начальные и граничные условия. Начальная температура воздуха в помещении задавалась равной 5 °С, стен, окна и батареи – 12 °С. Температура с внешней стороны комнаты, находящейся внутри дома, условно принималась постоянной и равной 14 °С, температура на внешней стороне ограждающей конструкции и на внешней поверхности окна принималась равной усредненному изменению температуры в течение суток в зимний период в течение всего времени численных исследований (рис. 3, *а*), пол и потолок изолированы:

$$\begin{cases} t_{\text{в}} = 5\ ^\circ\text{C}; \\ t_{\text{ст}} = 12\ ^\circ\text{C}; \\ t_{\text{о}} = 12\ ^\circ\text{C}; \\ t_{\text{бат}} = 12\ ^\circ\text{C}, \end{cases} \tag{2}$$

$$\begin{cases} t_{\text{с}}(0, y, z) = \text{const}, \ 0 \le y \le c, \ \ 0 \le z \le b; \\ t_{\text{с}}(a, y, z) = \text{const}, \ 0 \le y \le c, \ \ 0 \le z \le b; \\ t_{\text{с}}(x, y, 0) = \text{const}, \ 0 \le x \le a, \ \ 0 \le z \le b; \\ t_{\text{с}} = f(x, y, z, \tau); \\ \delta Q = 0, \end{cases} \tag{3}$$

где a – длина комнаты, м; b – ширина комнаты, м; c – высота комнаты, м; $t_{\text{ст}}$ – температура стен, °С; $t_{\text{с}}$ – температура поверхности ограждающей конструкции, °С; $t_{\text{бат}}$ – температура батареи, °С; $t_{\text{о}}$ – температура окна, °С.

Были заданы значения теплового потока для каждой точки поверхности стен и любого момента времени как граничное условие второго рода:

$$q_{\text{п}} = -\lambda \, grad \, t, \tag{4}$$

где $q_{\text{п}}$ – плотность теплового потока на поверхности ограждающей конструкции, т. е. количество теплоты, переданное через изотермическую поверхность в единицу времени; значения λ представлены в таблице.

Граничное условие третьего рода (теплообмен между поверхностью и окружающей средой в процессе охлаждения и нагревания ограждающей конструкции) задавалось с помощью закона Ньютона – Рихмана. Согласно закону Ньютона – Рихмана количество теплоты, отдаваемое еди-

ницей поверхности ограждающей конструкции в единицу времени, пропорционально разности температур поверхности ограждающей конструкции t_c и окружающей среды t_{oc} ($t_c > t_{oc}$):

$$q = \alpha(t_c - t_{oc}),$$

где α – коэффициент теплоотдачи наружной поверхности ограждающей конструкции для зимних условий, 23 Вт/(м^2·ºС) [8].

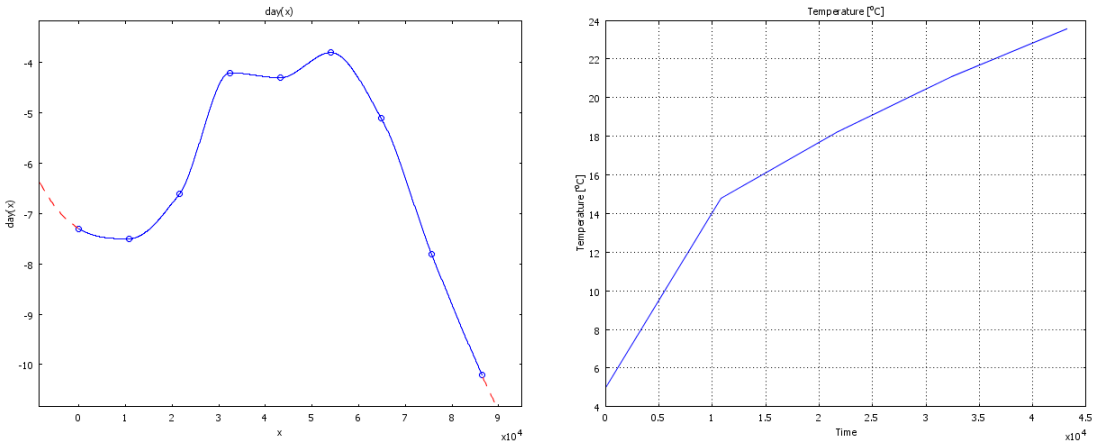
Равенство температур и плотностей теплового потока на границах соприкасающихся слоев ограждающих конструкций принималось граничным условием четвертого рода:

$$t_i = t_j;$$

$$q_i = q_j.$$

Скорость движения воздуха в жилой комнате принималась согласно нормам, которые установлены ГОСТ 30494–96 [10].

Из графика на рис. 3, *б* видно, что температура воздуха в центре комнаты составит примерно 8,3, 14,8, 18 и 23,5 °С через 1, 3, 6 и 12 ч соответственно, т. е. для заданных условий в течение 6 ч после включения источника тепла температура в помещении становится близкой к комфортной (18 °С). Результаты расчета температуры во всем объеме комнаты показаны на рис. 4.



а) *б)*

Рис. 3. Трафики изменения температуры: *а)* в течение суток в зимний период; *б)* центре комнаты при нагреве

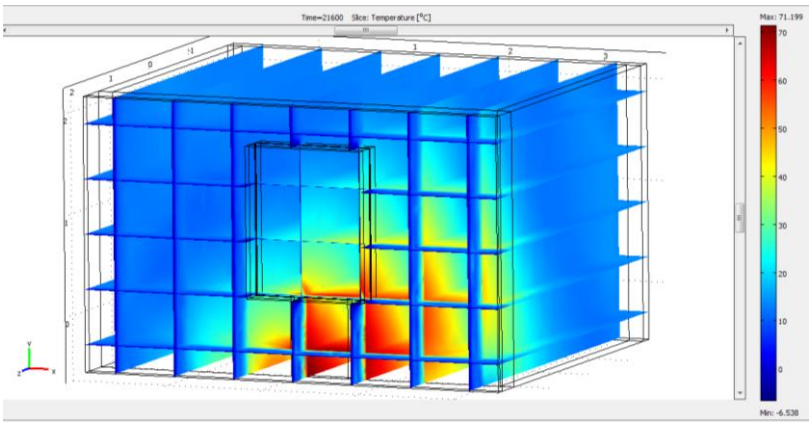


Рис. 4. Распределение температуры внутри комнаты через 6 ч

Рассмотрим моделирование распределения температуры в поперечном сечении стен и крыши дома в течение суток при отключении отопления. Начальная температура на внут-

ренней стороне ограждающих конструкций принималась равной 18 °С, а на внешней границе – в соответствии с усредненным изменением температуры в течение суток в зимний период (рис. 5 в координатах системы (1, 1, *b*)). Исследовалась конструкция утепления стен тростник – щепа и глина.

На рис. 5 видно, что применение тростника в качестве теплоизоляционного материала оказывает положительное влияние на тепловой режим дома: тепловой ноль (рис. 6) находится внутри тростникового теплоизоляционного слоя. Это позволяет демпфировать тепловые расширения и конденсацию влаги, дает возможность стене оставаться сухой, благоприятно отражается на состоянии конструкции и способствует формированию благоприятного климата внутри здания. Таким образом, утепление снаружи препятствует появлению известной проблемы внутренней теплоизоляции, когда в холодное время года на внутренних поверхностях образуется конденсат, а также защищает от деструктивного влияния снегопадов, атмосферных осадков, капиллярной влаги, ледовых образований и температурных перепадов. Все это повышает долговечность постройки. Стена внутри теплоизоляционной «шубы» перестает подвергаться температурным перепадам и, оставаясь постоянно нагретой изнутри, становится своеобразным аккумулятором тепла, способствуя сохранению комфортной температуры.

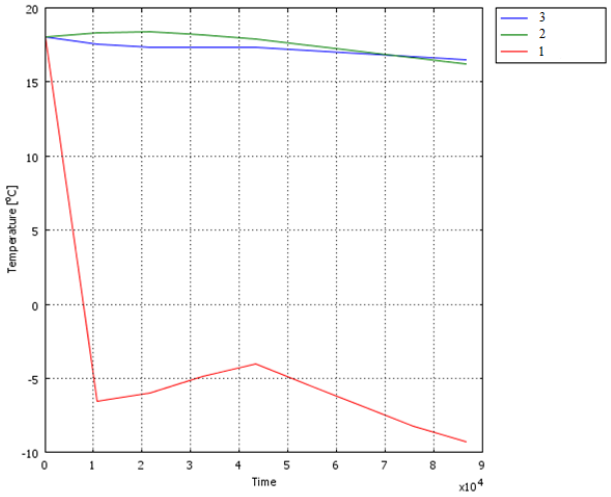


Рис. 5. Изменение температуры в сечении стен дома: 1) на внешней границе первого слоя (изменение температуры атмосферного воздуха); 2) на границе тростник – щепа и глина; 3) на внутренней границе ограждающей конструкции (time – время в секундах)

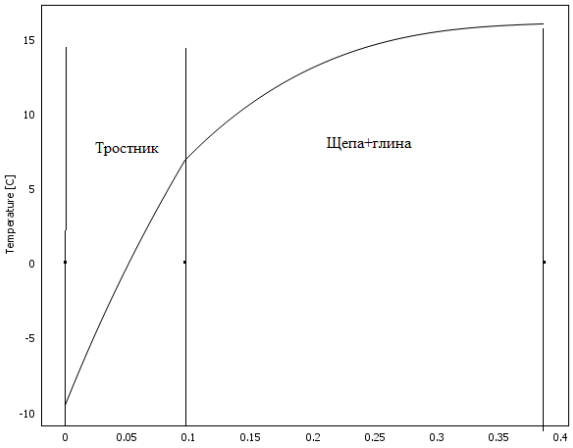


Рис. 6. Изменение температуры в ограждающих конструкциях стены при расположении тростниковой теплоизоляции с внешней стороны стены

Для обоснованного применения разработанных моделей осуществлена их экспериментальная верификация с использованием результатов экспериментальных исследований, проведенных фирмой «ЭкоДом» [11]. Контролировалось изменение температуры внутри дома после его прогрева до температуры 25 °С с последующим отключением отопления и естественным охлаждением (температура окружающей среды составляла –30 °С). В качестве утепляющего материала использовались тростниковые маты.

В течение трех суток температура внутри дома снизилась с 25 до 16 °С. Результаты моделирования изменения температуры внутри дома для аналогичной конструкции и начальных условий показали, что через трое суток температура снизилась до 14,5 °С. Полученное значение ниже, чем экспериментальное. Это можно объяснить тем, что реальные теплотехнические характеристики материалов могут отличаться от справочных данных. Погрешность результатов моделирования не превышает 10 %, что позволяет использовать ее для решения задач анализа тепловых режимов зданий.

Закючение

Разработана трехмерная тепловая модель индивидуального жилого дома для сельской местности, которая включает конструктивную, конечно-элементную и расчетно-аналитическую модели. В качестве объекта анализа выбрана типовая конструкция дома фирмы «ЭкоДом» с применением в качестве утеплителя местных материалов (тростник, щепа и глина). Конструктивная модель создана в программном комплексе SolidWorks. Конечно-элементная и расчетно-аналитические модели и непосредственно расчеты выполнены в программном комплексе COMSOL Multiphysics. Проведенные с помощью разработанной интегрированной модели исследования энергоэффективного дома УНК «Волма» МГЭУ им. А.Д. Сахарова подтвердили ее работоспособность и возможность использования для решения задач анализа тепловых режимов зданий и оптимизации ограждающих конструкций.

Список литературы

1. Кундас, С.П. Компьютерное моделирование технологических систем / С.П. Кундас, Т.А. Кашко. – Минск : БГУИР, 2002. – 164 с.

2. Моделирование процессов термовлагопереноса в капиллярно-пористых средах / С.П. Кундас [и др.]. – Минск : ИТМО НАН Беларуси, 2007. – 292 с.

3. Чечеткин, А.В. Теплотехника / А.В. Чечеткин. – М. : Высшая школа,1986. – 344 с.

4. Лыков, А.В. Теория теплопроводности : учебное пособие / А.В. Лыков. – М. : Высшая школа, 1967. – 600 с.

5. Сегерлинд, Л. Применение метода конечных элементов / Л. Сегерлинд. – М. : Мир, 1979. – 427 с.

6. Энергоэффективный дом УНК «Волма»: применение возобновляемых источников энергии в системе энергоснабжения / С.П. Кундас [и др.] // Возобновляемые источники энергии: потенциал, достижения, перспективы : материалы Междунар. семинара экспертов. – Минск : Ин-т энергетики НАН Беларуси, 2013. – С. 68–78.

7. Блоки стеновые из арболита для малоэтажного строительства. Технические условия : СТБ 1105–98. – Минск : Министерство архитектуры и строительства Республики Беларусь, 1998.

8. Строительная теплотехника. Строительные нормы проектирования : ТКП 45-2.04-43–2006 (02250). – Минск : Стройтехнорм, 2006. – 56 с.

9. Строительные нормы и правила. Отопление, вентиляция и кондиционирование : СНиП 2.04.05–91. – Введ. 01.01.92. – М. : Промстройпроект, 1992. – 81 с.

10. Межгосударственный стандарт. Здания жилые и общественные. Параметры микроклимата в помещениях : ГОСТ 30494–96. – Введ. 01.03.99. – М. : Госстрой России, 1999. – 7 с.

11. ForumHouse. Строительство, ремонт, стройматериалы / Строительство дома, бани, коттеджа. Дома из органического сырья [Электронный ресурс]. – Режим доступа : https://www.forumhouse.ru. – Дата доступа : 24.11.2014.

Поступила 17.09.2014

[1]*Международный государственный*
экологический университет им. А.Д. Сахарова,
Минск, ул. Долгобродская, 23
e-mail: elena-kresova@mail.ru

[2]*Белорусский национальный*
технический университет,
Минск, пр. Независимости, 65
e-mail: kundas@tut.by

E.V. Kresova, S.P. Kundas

THERMAL MODEL OF ENERGY EFFICIENT BUILDING

Thermal 3-D model of a rural area building is proposed. The model includes constructive, finite element, analytical and computational models which are created using SolidWorks software (constructive model) and COMSOL Multiphysics software (finite element, analytical and computational models). An additional object was created for modeling heat transfer process in the whole volume of building. Studies have shown model efficiency and possibility of its application for optimizing building isolation.

УДК 519.95

Г.П. Волчкова, В.М. Котов

ИССЛЕДОВАНИЕ СВОЙСТВ ПЛОТНЫХ РАСПИСАНИЙ ПРИ ОГРАНИЧЕННОМ ЧИСЛЕ ПРИБОРОВ

Для задачи $Om||C_{\max}$ существует гипотеза, что в худшем случае для любого плотного расписания время завершения выполнения последней работы не более чем в $2-1/m$ раз превосходит время завершения в оптимальном расписании. Предлагается подход, который позволяет доказать гипотезу для случая $m \le 9$ и некоторых специальных случаев.

Введение

Известна следующая задача теории расписаний: множество $N = \{1, 2, \ldots, n\}$ работ выполняется в системе из $M = \{1, 2, \ldots, m\}$ приборов. Порядок прохождения приборов не задан и может быть различным для различных работ. Заданы длительности выполнения каждой работы каждым прибором. В любой момент времени каждая работа выполняется не более чем одним прибором и каждый прибор выполняет не более одной работы. Требуется построить расписание (указать для каждой пары «работа – прибор» интервал времени, в котором этот прибор выполняет данную работу), при котором минимизируется общее время обслуживания всех работ. В теории расписаний такая задача обозначается $Om \,||\, C_{\max}$.

Расписание будем искать в классе плотных расписаний, для которых прибор не может простаивать, если некоторая работа готова к выполнению на нем. Существует гипотеза [1] , что для плотного расписания *S* справедливо следующее неравенство:

$$\Delta = \frac{C_{\max}(S)}{C_{\max}(S^*)} \le 2 - \frac{1}{m}, \qquad (1)$$

где S^* – оптимальное расписание в классе всех расписаний.

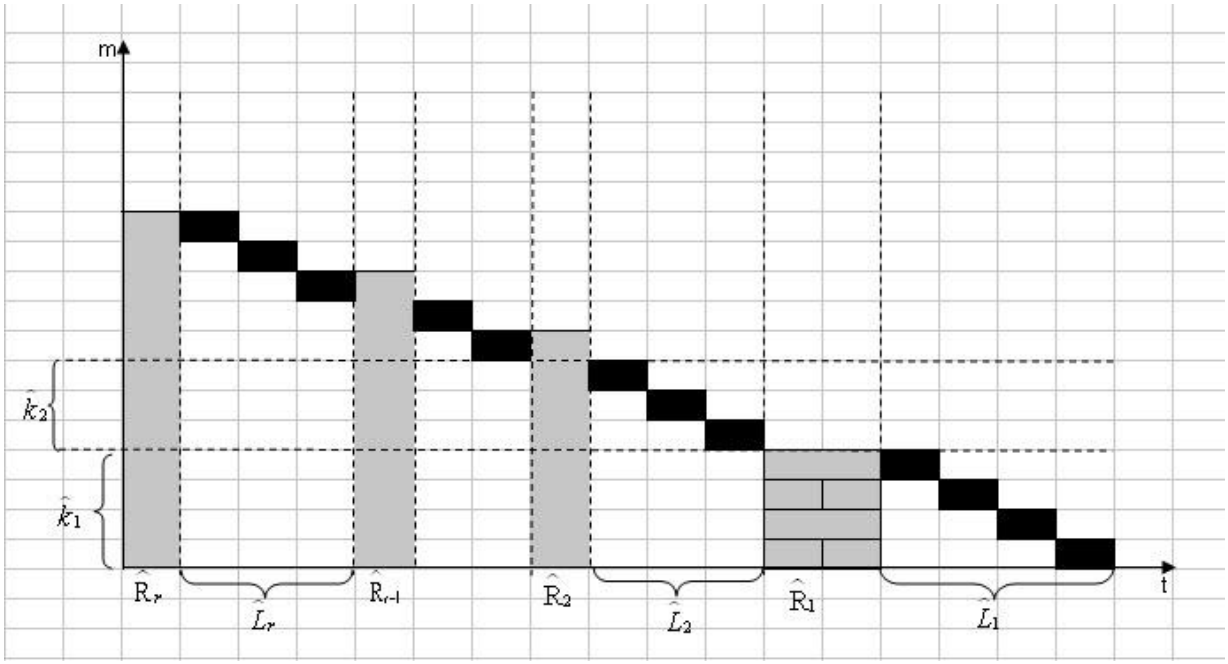
В работе [2] гипотеза доказана для случая, когда в расписании не более одного разрыва (определение см. ниже). Доказано, что гипотеза справедлива для $m \le 6$ [3] и $m \le 8$ [4]. В настоящей статье гипотеза доказана для $m \le 9$ и некоторых специальных случаев.

1. Определения

Под операцией (i, j) будем понимать выполнение работы *i* определенным прибором *j*, где t_{ij} – ее длительность, $i = 1,\ldots,n,\ j = 1,\ldots,m$. Последовательность всех операций работы *i* назовем цепочкой l_i, ее длина $l_i = \sum_{j \in M} t_{ij}$, а $l_{\max} = \max_{i \in N}(l_i)$. Загрузку *j*-го прибора Z_j определим как $Z_j = \sum_{i=1}^{n} t_{ij}$, $j = 1,\ldots,m$, а максимальную загрузку $\mathrm{Z}_{\max}$ – как $\mathrm{Z}_{\max} = \max_{j=1,\ldots,m}(Z_j)$.

Рассмотрим любое плотное расписание *S*. Введем следующие обозначения. Последняя завершаемая работа в S называется *красной* (пусть это работа *k*), и она последовательно выполняется приборами $m, m-1, \ldots, 2, 1$. Временной интервал (область), когда не выполняется ни одна операция работы *k*, называется *разрывом*; нумерация разрывов осуществляется с конца. Пусть *i*-й разрыв цепочки работы *k* – R_i, а R_i – его ширина, $i = 1,\ldots,r$. Временной интервал

(область), в котором операции работы *k* выполняются без задержек, будем обозначать через L_i, а его ширину – L_i, $i=1,\ldots,r$. Множество приборов в L_i, которые выполняют ненулевые операции работы k, обозначим через k_i, а их количество – через k_i, $i=1,\ldots,r$ (рисунок). Работы из разрыва R_1 будем называть *черными*, и пусть n_1 – их количество. Суммарную длительность операций черных работ в разрывах $R_2,\ldots,R_r$ будем обозначать через P.



Общий вид расписания

Для набора входных данных *In* определим $HO(In)=\max\{l_{\max},Z_{\max}\}$ – нижнюю оценку оптимального решения. Набор входных данных *In* и плотное расписание *S* для него будем называть *контрпримером* (S,In), если

$$f(S,In)=\frac{C_{\max}(S)}{HO(In)}>2-\frac{1}{m}. \qquad (2)$$

Если гипотеза не выполняется, то существует набор входных данных *In* и расписания *S*, S^* для него, такие что $\frac{C_{\max}(S)}{C_{\max}(S^*)}>2-\frac{1}{m}$. Однако $HO(In)\le C_{\max}(S^*)$, поэтому $f(S,In)\ge\frac{C_{\max}(S)}{C_{\max}(S^*)}$, т. е. (S,In) – контрпример. Следовательно, отсутствие контрпримера гарантирует справедливость гипотезы.

Контрпример будем называть приведенным, если для расписания *S* и набора входных данных справедливы следующие свойства:

1) первый прибор имеет максимальную загрузку;
2) цепочка красной работы имеет максимальную длину;
3) величина максимальной загрузки равна длине максимальной цепочки;
4) приборы из множества k_1 простаивают в областях L_i, $i=2,...,r$;
5) $t_{k1}=t_{k2}=\ldots=t_{kk_1}=x$;
6) приборы из множества k_1 не простаивают в разрывах R_i, $i=1,\ldots,r$;

7) хотя бы одна черная работа имеет цепочку максимальной длины;

8) все черные работы и работа *k* выполняются в каждой из областей L_i, $i = 2,\ldots,r$;

9) $P + k_1 R_1 \le n_1 k_1 x$;

10) k_1 черных работ не могут выполняться одновременно в интервале $[t', t''] \in R_i$, $i = 2,...,r$;

11) в области L_2 выполняются только красная и черные работы.

Рассмотрим интервал времени (область) $[t', t'']$ ширины $\Delta = t'' - t'$ в *S*. Под *операцией вырезания* будем понимать такое преобразование *S*, при котором длительность всех операций, выполняемых в интервале $[t', t'']$, уменьшается на время их выполнения в этом интервале, а все операции, выполняемые после этого интервала, сдвигаются влево на величину Δ.

2. Основные свойства

Утверждение 1. *Если существует контрпример, то существует и приведенный контрпример.*

Доказательство. Покажем, как контрпример преобразуется в приведенный контрпример:

1. Предположим, что $Z_1 < Z_{\max}$. Тогда в области L_2 существует интервал времени $[t', t'']$ шириной $\Delta = t'' - t'$, $\Delta \le Z_{\max} - Z_1$. После применения операции вырезания к этому интервалу величину x_1 увеличим на Δ. Получим расписание S', в котором загрузка Z_1' не превысила $Z_{\max}$ (увеличилась загрузка только первого прибора), длины цепочек всех работ, кроме работы *k*, не увеличились. Так как работа *k* выполнялась в интервале времени $[t', t'']$ на других приборах, то $l'_{\max}$ не превысила $l_{\max}$. При этом в S' первый прибор имеет максимальную загрузку.

2. Предположим, что в расписании S выполняется $l_k < l_{\max}$. Определим в разрыве R_1 интервал времени $[t', t'']$ шириной $\Delta = \min\{l_{\max} - l_k,\ R_1\}$. Применим операцию вырезания к этому интервалу, а длительность x_1 увеличим на величину Δ. При этом $Z'_{\max}$ не увеличилась, а длина l'_k увеличилась и стала равной $l_{\max}$. Пусть в расписании S $Z_{\max} < l_{\max}$. Определим величину $\Delta = l_{\max} - Z_{\max}$. Ширину разрыва R_r увеличим на величину Δ, добавив на каждый из m приборов операцию длительности Δ. Максимальная загрузка станет равной длине максимальной цепочки в S'.

3. Пусть $Z_1 > l_k$. Определим в разрыве R_i интервал времени $[t', t'']$ шириной $\Delta = \min\{Z_1 - l_k, R_i\}, i = 1,\ldots,r$. Применим операцию вырезания к этому интервалу в разрыве R_i, а длительность x_1 увеличим на Δ. Получим расписание S', в котором l'_k увеличилась на Δ. Если $l'_k < Z_1$, выполним аналогичные действия над следующим разрывом. В результате таких преобразований получим либо расписание с одним разрывом, либо расписание, у которого $Z'_1 = l'_k = Z_1$, а $f(S', In')$ увеличилась.

Пусть $Z_1 < l_k$. Величину R_r увеличим на $l_k - Z_1$. В новом расписании $Z'_1 = l_k$ и $l'_k = l_k$.

4. Пусть в одной из областей $L_2,\ldots,L_r$ прибор с номером j, $j \le k_1$, занят в интервале $[t', t'']$. Применим операцию вырезания к этому интервалу, а длительность x_j увеличим на $t'' - t'$. Получим расписание S', в котором l'_k и Z'_1 не увеличились.

5. В силу п. 1 $t_{k1} \geq t_{ki}$, $i = 2,\ldots,k_1$. Пусть существует i, такое что $t_{k1} > t_{ki}$. Рассмотрим расписание S', в котором $t'_{k,j} = x$, $j = 1,\ldots,k_1$, где $x = L_1 / k_1 < t_{k1}$. В S' загрузка $Z'_i = \sum_{j=1}^{r} R_j + x < \sum_{j=1}^{r} R_j + t_{k1} \leq Z_1$, $i = 1,\ldots,k_1$, а $l'_k = l_k$.

6. Если приборы из k_1 простаивают в областях L_i, $i = 2,...,r$, то в силу плотности расписания операции работы *k* могут выполняться в разрывах.

7. Пусть предпосылка в п. 7 не выполняется. Выберем из разрыва R_1 работу с цепочкой наибольшей длины среди всех черных работ l_u и положим $\Delta = l_{\max} - l_u$. В разрыве R_2 выделим интервал $[t', t'']$ шириной $t'' - t' \leq \Delta$ и применим к нему операцию вырезания. Если $R_2 < \Delta$, то выполним аналогичные действия над следующим по порядку разрывом. Так будем действовать до тех пор, пока суммарная ширина вырезанного интервала не достигнет Δ. Затем длительности операций k_1 черных работ в разрыве R_1 увеличим на величину Δ. При таком преобразовании $l'_{\max}$ и $Z'_{\max}$ в S' не увеличились, а длина цепочки хотя бы одной черной работы станет равна $l'_{\max}$.

8. Если одна из черных или красная работа не выполняются в одной из L_i, $i = 2,\ldots,r$, то она может быть выполнена приборами из множества k_1, что противоречит п. 4.

9. Для суммарной длительности операций черных работ справедливо $n_1 \sum_{i=2}^{r} L_i + P + k_1 R_1 \leq n_1 l_k = n_1 (\sum_{i=2}^{r} L_i + k_1 x)$. Отсюда $P + k_1 R_1 \leq n_1 k_1 x$.

10. Если в некотором интервале времени в разрыве R_i, $i = 2,\ldots,r$, выполняются одновременно k_1 черных работ, то применим к такому интервалу операцию вырезания, а длительности операций этих k_1 работ в разрыве R_1 увеличим на ширину данного интервала.

11. Положим равными нулю в области L_2 длительности операций всех работ, кроме черных и красной. В полученном расписании l'_k не изменилась, а Z'_1 не увеличилась.

Пронормируем входные данные, разделив длительность каждой операции на величину $l_{\max}$. Тогда справедливо

$$Z_1 = R_1 + R_2 + ... + R_r + x = L_1 + L_2 + ... + L_r = 1; \quad (3)$$

$$L_2 + ... + L_r = 1 - L_1 = 1 - k_1 x. \quad (4)$$

Теорема 1. *Не существует контрпримера, для которого* $x \geq 1/m$.

Доказательство. Пусть (S, In) является контрпримером, у которого $x \geq 1/m$. Рассмотрим набор In' из m работ, причем $t'_{ij} = (1-x)/(m-1)$, $i = 1,...,m-1$, а $t'_{mj} = 1/m$, $j = 1,...,m$. Пусть S' – плотное расписание, в котором сначала без простоев выполняются работы $1,...,m-1$, а затем работа m. Тогда $C_{\max}(S') = C_{\max}(S) = 2 - x$, $Z'_{\max} = 1 - x + 1/m \leq Z_{\max}$. Поэтому если $f(S, In) > 2 - 1/m$, то $f(S', In') > 2 - 1/m$. Это противоречит тому, что $f(S', In') = 2 - x$.

Введем следующие обозначения:

NL_i – множество приборов, которые выполняют красную или черную работы в какой-то момент времени в области L_i, $i = 2,\ldots,r$;

$NL_2(t)$ – множество приборов, которые работают в L_2 в момент времени t;

$ID(t_i,t)$ – множество приборов из $NL_2(t)\cup k_1$, которые простаивают в области L_i, $i=3,\ldots,r$, в момент времени t_i;

NR_2 – множество приборов из $NL_2\cup k_1$, которые не простаивают в разрыве R_2;

$NR_2(t)$ – множество приборов, которые работают в разрыве R_2 в момент времени t.

Лемма 1. *Если существует контрпример, то существует контрпример, для которого справедливо* $|ID(t_i,t)|\ge k_1+1$.

Доказательство. Если хотя бы один из приборов множества $NL_2(t)$ в момент времени t_i простаивает, то лемма 1 справедлива. Пусть (S,In) – приведенный контрпример, в котором все приборы из $NL_2(t)$ работают в любой момент времени в интервале L_i, $i=3,\ldots,r$. Применим к этому интервалу операцию вырезания, а длительности операций красной и черных работ, выполняемых соответствующими приборами в L_2, увеличим на L_i. При этом величина l'_k не изменилась, Z'_1 не увеличилась. Последовательность таких преобразований приведет к расписанию с одним разрывом, а для таких расписаний контрпримеров не существует.

В дальнейшем будем рассматривать только наборы входных данных и расписания, которые являются приведенными контрпримерами и удовлетворяют лемме 1.

Работы из разрыва R_2, отличные от красной и черных, будем называть *синими*. Обозначим через $New(t,t_i,t_2)$ минимальное количество синих работ, которые должны выполняться приборами из $ID(t_i,t)$ в момент времени t_2 в разрыве R_2. В силу п. 10 справедливо

$$New(t,t_i,t_2)\ge\left|ID(t_i,t)\cap NR_2(t_2)\right|-(k_1-1). \tag{5}$$

Пусть $m(t_i)$ – количество приборов, необходимых для выполнения работ в L_i, $i=3,\ldots,r$, в момент времени t_i. Тогда

$$m(t_i)\ge n_1+1+\max_{t,t_2}\{ID(t_i,t)+New(t,t_i,t_2)\}. \tag{6}$$

Отметим, что в силу п. 5 $L_1=k_1x$.

Лемма 2. *Не существует контрпримера, у которого* $m<n_1+k_1+4$.

Доказательство. Если в одной из областей L_i, $i\ge2$, в некоторый момент времени t_i простаивает хотя бы один прибор из множества $NR_2\setminus k_1$, то в силу леммы 1 $|ID\ (t_i,t)|\ge k_1+1$, поэтому из (5) и (6) следует $m(t_i)\ge n_1+k_1+4$.

Пусть в областях L_i, $i\ge2$, не простаивает ни один прибор из множества $NR_2\setminus k_1$. Тогда $Z_{k_1+1}=1-(R_1+x)+1-L_1\le1$. Отсюда с учетом $L_1=k_1x$ и п. 9 получим $1\le n_1\cdot x-P/k_1+(k_1+1)\cdot x$. В силу $P\ge0$ имеем $1\le n_1x+(k_1+1)x$. Учитывая теорему 1, $m<(n_1+k_1+1)$, что противоречит (6) в силу леммы 1 и (5).

Теорема 2. *Не существует контрпримера для* $m\le9$, *у которого* $k_1\ge2$.

Доказательство. Если $k_1+n_1\ge6$, то из леммы 2 следует $m>9$. Противоречие.

Пусть $k_1=2$, $2\le n_1\le3$. Обозначим через j_0 прибор из множества NL_2 с простоем максимальной суммарной длительности в разрыве R_2. Если в момент времени t_i в области L_i, $i\ge3$, простаивают не менее двух приборов из $NL_2\setminus j_0$, то $|ID(t_i,t)|\ge k_1+2$. Из (5) и (6) следует, что $m(t_i)>9$.

Случай 1. $|NL_2|=n_1+1$ (ни один прибор из множества NL_2 не простаивает в области L_2).

Пусть в каждой из областей L_i, $i \geq 3$, простаивает не более одного прибора из множества $NL_2 \setminus j_0$. Если прибор $j \in NL_2 \setminus j_0$ простаивает в разрыве R_i, $i \geq 2$, то черная работа, операцию которой он выполняет в L_2, в силу плотности расписания должна выполняться в R_i. При этом в силу п. 10 в R_i в один и тот же момент времени не может выполняться более одной черной работы. Поэтому суммарный простой приборов из $NL_2 \setminus j_0$ в R_i, $i \geq 2$, не превышает P, а для суммарной загрузки приборов из $NL_2 \setminus j_0$ справедливо $(n_1 - 1)(1 - L_1 - L_2) + n_1 L_2 + n_1(1 - (R_1 + x)) - -P \leq (n_1 + 1)Z_1$. Учитывая (4), $L_2 > 0$ и п. 9, получим $(n_1 - 1)(1 - k_1 x) < n_1 k_1 x + +(n_1 - k_1)(n_1 x - P / k_1) + n_1 x < 2n_1 x + (n_1 - 2)n_1 x + n_1 x$. Это соотношение с учетом леммы 2 противоречит теореме 1 при $n_1 = 2, 3$.

Случай 2. $|NL_2| > n_1 + 1$.

Случай 2.1. $k_2 \geq 2$. Пусть в любой момент времени в области L_2, $i \geq 2$, простаивает не более одного прибора из k_2. Тогда для суммарной загрузки приборов из k_2 справедливо $(k_2 - 1)(1 - L_1) + k_2(1 - (R_1 + x)) \leq k_2$. В силу $L_1 = 2x$ имеем $(k_2 - 1) \leq k_2(R_1 + x) + 2(k_2 - 1)x$. С учетом п. 9 получим $x \geq (k_2 - 1)/(2k_2 - 2 + n_1 k_2)$. Тогда при $2 \leq n_1 \leq 3$ получим $x \geq (k_2 - 1)/(5k_2 - 2)$, что в случае $k_2 \geq 2$ с учетом леммы 2 противоречит теореме 1.

Пусть существует момент времени t, когда простаивают приборы $j_1, j_2 \in k_2$ в области L_2 (это возможно только при $k_2 \geq 3$). Рассмотрим приборы множества $NL_2(t) \cup j_1 \cup j_2$.

Если в интервале времени $[t', t'']$ в области L_3 работает $n_1 + 1$ прибор из этого множества, то применим операцию вырезания к интервалу $[t', t'']$, а длительности операций красной и черных работ на соответствующих приборах в области L_2 увеличим на величину $t'' - t'$. При этом величина l'_k не изменилась, Z'_1 не увеличилась.

Если таких интервалов в области L_3 нет, то в любой момент в области L_3 простаивают по крайней мере три прибора из $NL_2(t) \cup j_1 \cup j_2$, причем никакие два из этих трех приборов не могут одновременно простаивать в R_2 (иначе в R_2 в один и тот же момент времени будут выполняться как минимум две черные работы, что противоречит п. 10). Поэтому в L_3 в момент времени t_i с учетом п. 4 простаивают как минимум пять приборов, а новых (синих) работ, выполняющихся в R_2 в силу (5), будет как минимум три. Поэтому в силу (6) $m(t_i) > 9$. Противоречие.

Случай 2.2. $k_2 = 1$. Покажем, что существует черная работа i_1, которая выполняется в области L_2 прибором $j_1 \in NL_2$, причем прибор j_1 работает в разрыве R_2.

Пусть i_1 – одна из черных работ, которая выполняется прибором $j_1 \in NL_2$, причем этот прибор простаивает в разрыве R_2. Рассмотрим другую черную работу i_2, которая выполняется в области L_2 прибором $j_2 \in NL_2$. Прибор j_2 не может простаивать в разрыве R_2 все время, так как в этом случае две черные работы будут выполняться в R_2 в одном и том же интервале времени, что противоречит п. 10.

Если в одной из областей L_i, $i \geq 3$, в один и тот же момент времени простаивают по крайней мере два прибора с номерами $k_1 + 1$ и j_1, то $|ID(t_i, t)| \geq k_1 + 2$ и из (5), (6) следует, что $m(t_i) > 9$. Противоречие.

Пусть в каждой из областей L_i, $i \geq 3$ в любой момент времени простаивает не более одного прибора из приборов с номерами $k_1 + 1$ и j_1. Тогда для их суммарной загрузки справедли-

во $1-L_1+2(1-(R_1+x))-P\le 2Z_1$. Отсюда получим $1-L_1\le P+2R_1+2x$. В силу п. 9 верно $1-L_1\le 2n_1x+2x$. Поэтому в силу (4) верно $1-2x\le 2n_1x+2x$ и $1\le 8x$, что с учетом леммы 2 противоречит теореме 1.

Осталось рассмотреть вариант, когда $k_1=1$. В этом случае $NR_2=NL_2\cup k_1$, $R_1+x=(n_1+1)x$, любая черная работа выполняется в областях L_i и не выполняется в разрывах R_i, $i\ge 2$.

Лемма 3. *Если* $NR_2=NL_2\cup k_1$, *то для любых* t_i, t *справедливо* $|ID(t_i,t)|\ge|NR_2|-n_1$.

Доказательство. В L_2 выполняются n_1+1 работы (черные и красная). Если в L_i в интервале времени $[t',t'']$ на приборах из NL_2 выполняются не менее n_1+1 работ, то применим к этому интервалу операцию вырезания, а длительности операций красной и черных работ в L_2 на соответствующих приборах увеличим на величину $\Delta=t''-t'$. Последовательность таких преобразований приведет либо к расписанию с одним разрывом, либо к расписанию требуемого вида.

Пусть $i_1, i_2,\ldots, i_p$ – синие работы, операции которых выполняет первый прибор в R_2.

Лемма 4. *Если суммарная длительность операций одной из работ* $i_1, i_2,\ldots, i_p$ *в разрыве* R_2 *равна* R_2, *то справедливо* $R_2\le L_1+L_2$.

Доказательство. Пусть суммарная длительность операций одной из работ $i_1, i_2,\ldots, i_p$ в разрыве R_2 равна R_2. В силу того что операции этой работы выполняются в интервалах L_i, $i\ge 3$, в любой момент времени t_i, длина ее цепочки как минимум $1-L_1-L_2+R_2$, что противоречит п. 2 при $R_2>L_1+L_2$.

Пусть $X_3=NL_3\setminus(NL_2\cup k_1)$ – множество приборов, которые выполняют красную и черные работы в области L_3 и не выполняют их в области L_2.

Теорема 3. *Не существует контрпримера для* $m\le 9$, *у которого* $k_1=1$.

Доказательство. Если $|NL_2|\ge n_1+3$, то в силу (5), (6) и леммы 3 справедливо $m(t_i)>9$. Противоречие.

Случай 1. $|NL_2|=n_1+2$.

Если $n_1\ge 3$, то в силу (5), (6) и леммы 3 справедливо $m(t_i)>9$. Противоречие.

Случай 1.1. $n_1=2$. Если существует момент времени t_i, когда в одной из областей L_i, $i\ge 3$, простаивают по крайней мере три прибора из NL_2, то $|ID(t_i,t)|\ge 4$. Поэтому из (5), (6) следует, что $m(t_i)>9$. Противоречие.

Пусть в любой момент времени t_i в каждой из областей L_i, $i\ge 3$, простаивают два прибора из NL_2, тогда $|ID(t_i,t)|=3$. Для загрузки приборов из NL_2 справедливо $4(1-(R_1+x))+2(1-L_1-L_2)+3L_2\le 4$. Отсюда $2+L_2\le 14x$, что противоречит теореме 1.

Случай 1.2. $n_1=1$. Пусть T_1 – длина суммарного интервала времени в областях L_i, $i\ge 3$, когда работает хотя бы один из приборов из NL_2. Для загрузки приборов из NL_2 справедливо $2(1-(R_1+x))+2L_2+T_1\le 2$. С учетом $R_1=2x$ получим

$$T_2+L_2+2x<8x. \tag{7}$$

К интервалам в областях L_i, $i\ge 3$, где заняты приборы из NL_2, применим операцию вырезания, а длительности операций красной и черных работ после разрыва R_2 преобразуем следующим образом. Длительности операций черных работ, выполняемых приборами $NL_2\cup k_1$,

полагаем равными нулю, а длительности операций красной работы полагаем равными $(T_1+L_2+2x)/3$. Получим расписание S', в котором l'_k не изменилась, Z_1 не увеличилась, а $k_1=3$, что противоречит теореме 2.

Случай 2. $|NL_2|=n_1+1$.

Случай 2.1. $n_1\ge 3$. Если существует момент времени t_i, когда в одной из областей L_i, $i\ge 3$, простаивают по крайней мере два прибора из NL_2, то $|ID(t_i,t)|\ge 3$. Поэтому с учетом (5), (6) справедливо $m(t_i)>9$. Противоречие.

Пусть в любой момент времени t_i в каждой из областей L_i, $i\ge 3$, простаивает не более одного прибора из NL_2. Тогда $|ID(t_i,t)|=2$. Для загрузки приборов из NL_2 справедливо $(n_1+1)(1-(R_1+x))+n_1(1-L_1-L_2)+(n_1+1)L_2\le n_1+1$. Учитывая $L_2>0$, получим $n_1<(n_1+1)^2x+n_1x$, что невозможно с учетом теоремы 1 и $n_1\ge 3$.

Случай 2.2. $n_1=2$. Если существует момент времени t_i, когда в одной из L_i, $i\ge 3$, простаивают все приборы из NL_2, то $|ID(t_i,t)|\ge 4$. Поэтому с учетом (5), (6) справедливо $m(t_i)>9$. Противоречие.

Если в любой момент времени t_i в каждой из областей L_i, $i\ge 3$, простаивает не более одного прибора из NL_2, то для загрузки приборов из NL_2 справедливо $3(1-(R_1+x))+2(1-L_1-L_2)+3L_2\le 3$. С учетом $R_1=2x$, $L_1=x$ справедливо $2+L_2\le x+9x$. Согласно теореме 1 не существует приведенного контрпримера, у которого $x\ge 1/m(t_i)$, что противоречит полученному неравенству.

Если существует момент времени t_i, когда в одной из L_i, $i\ge 3$, простаивают по крайней мере два прибора из NL_2, то $|ID(t_i,t)|\ge 3$. Если в этот момент времени простаивает также прибор j, $j\in X_3$, то простаивают четыре прибора. Следовательно, $m(t_i)>9$. Противоречие.

Пусть в момент времени $t_i\in L_i$, $i\ge 3$, все приборы из X_3 работают. Обозначим через I_2 суммарную ширину интервалов, в которых в областях L_i, $i\ge 3$, одновременно простаивают два прибора из NL_2. Для загрузки приборов $NL_2\cup j$ справедливо $3(1-(R_1+x))+$ $+2(1-L_1-L_2-I_2)+3L_2+1-(R_1+x)-R_2+I_2\le 4$. Отсюда $L_2+2\le 4(R_1+x)+R_2+2x$.

Если суммарная длительность операций одной из синих работ $i_1, i_2,\ldots, i_p$ в R_2 равна R_2, то с учетом леммы 4 $L_2+2\le 15x+L_2$. Согласно теореме 1 не существует контрпримера, у которого $x>1/m(t_i)$, что противоречит полученному неравенству при $n_1=2$.

Если такой синей работы нет, то в R_2 на приборах из $ID(t_i,t)\cup j$ выполняются по крайней мере четыре синие работы, поэтому $m(t_i)>9$. Противоречие.

Случай 2.3. $n_1=1$. Для загрузки приборов из NL_2 справедливо $2(1-(R_1+x))+2L_2+I_2\le 2$, $I_2+L_2+2x<6x$. Аналогично случаю 1.2 теоремы 3 получим контрпример, у которого $k_1=3$, что противоречит теореме 2.

Заключение

В статье доказано, что при $m\le 9$ для любого плотного расписания S справедливо $\Delta=\dfrac{C_{\max}(S)}{C_{\max}(S^*)}\le 2-\dfrac{1}{m}$. Доказано также, что это верно и для плотных расписаний, у которых длительность последней выполняемой операции в расписании не меньше $\max\{Z_{\max}, l_{\max}\}/m$.

Работа выполнена при частичной финансовой поддержке БРФФИ (проект № Ф13МЛД-012).

Список литературы

1. Chen, B. Approximation algorithms for three machine open shop scheduling / B. Chen, V.A. Strusevich // ORSA J. Comput. – 1993. – Vol. 5. – P. 321–326.

2. Волчкова, Г.П. К гипотезе о плотных open-shop расписаниях / Г.П. Волчкова // Вестник БГУ. Сер. 1.2. – 2004. – № 2. – С. 58–61.

3. Волчкова, Г. П. О свойстве плотных расписаний для задачи $Om||C_{max}$ / Г.П. Волчкова // Вестник БГУ. Сер. 1.2. – 2010. – № 2. – С. 127–130.

4. Open-shop dense schedules : properties and worst-case performance ratio / B. Chen [et al.] // Journal of scheduling. – 2012. – Vol. 15, no. 1. – P. 3–11.

Поступила 19.06.2014

Белорусский государственный университет,
Минск, пр. Независимости, 4
e-mail: kotovvm@yandex.ru,
volchkovagp@mail.ru

G.P. Volchkova, V.M. Kotov

STUDYING PROPERTIES OF DENSE SCHEDULES UNDER CONDITION OF LIMITED NUMBER OF SERVICE UNITS

There is a conjecture that for any dense schedule in the problem $Om||C_{max}$ the makespan is at most $(2-1/m)$ times the makespan of the optimal schedule, where "m" is the number of machines. In the paper the conjecture is proved for $m \leq 9$ and some other special cases.

УДК 004.94: 004.9.032.26

П.К. Шалькевич[1], С.П. Кундас[2], И.А. Гишкелюк[1]

ТЕХНОЛОГИЯ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ ЗАДАЧИ ТЕПЛОВЛАГОПЕРЕНОСА В ПРОГРАММНОМ КОМПЛЕКСЕ SPS

Предлагается математическая модель, разработанная с учетом существующих численных моделей неизотермического влагопереноса в почве и адаптированная для конечно-элементного решения задачи тепловлагопереноса в природных дисперсных средах в трехмерной постановке. Разрабатывается программный комплекс SPS, позволяющий осуществлять долгосрочное прогнозирование для решения трехмерной задачи неизотермического тепловлагопереноса загрязняющих веществ в почве.

Введение

Для моделирования миграции загрязняющих веществ в природных дисперсных средах необходимо решить проблему неизотермического тепловлагопереноса [1], в основе которой лежит математическая задача, эффективно решаемая с помощью метода конечных элементов (МКЭ). МКЭ применяется в большинстве современных программных комплексов, предназначенных для моделирования сложных физических процессов [2–5], однако они ввиду своей универсальности имеют сложности в адаптации к решению конкретной задачи и не всегда позволяют получать достоверное решение [1].

Задача моделирования неизотермического тепловлагопереноса в одномерном виде успешно реализована авторами в программном комплексе (ПК) SPS (Simulation of Processes in Soil) [1]. Для большего приближения результатов моделирования к реальным процессам актуальна разработка трехмерных моделей миграции загрязняющих веществ в природных дисперсных средах, которая требует больших вычислительных ресурсов и времени выполнения вычислений, в особенности при долговременном прогнозировании. Эту задачу можно решить с помощью технологии параллельных вычислений [6]. Проведенный анализ показал, что применение параллельных вычислительных алгоритмов, выполняемых на базе стандартных инструкций, не дает требуемой эффективности [6]. Поэтому целью настоящей работы является реализация специализированных моделей и программных средств для решения поставленной задачи.

1. Численная модель неизотермического влагопереноса

Система уравнений тепловлагопереноса имеет следующий вид [1]:

$$\begin{vmatrix} C_{\mathrm{v}}\dfrac{\partial T}{\partial t} - L\dfrac{\partial \theta_{\mathrm{liq}}}{\partial t} + L\rho_{\mathrm{liq}}\nabla\upsilon_{\mathrm{liq}} - \nabla\left(\lambda\nabla T\right) \\ C_{\mathrm{hv}}\dfrac{\partial T}{\partial t} + C_{\mathrm{wp}}\dfrac{\partial P_{\mathrm{liq}}}{\partial t} - \nabla\left(K_{\mathrm{hv}}\nabla T\right) - \nabla\left(K_{\mathrm{wv}}\nabla P_{\mathrm{liq}} - K_{\mathrm{w}}\rho_{\mathrm{liq}}g\nabla D\right) \end{vmatrix} = 0, \qquad (1)$$

где коэффициенты находятся из выражений $C_{\mathrm{hv}} = \dfrac{\partial w}{\partial T}$, $C_{\mathrm{wp}} = \dfrac{\partial w}{\partial P_{\mathrm{liq}}}$, $K_{\mathrm{hv}} = \rho_{\mathrm{v}}\dfrac{K_0 K_{\mathrm{v}}}{\eta_{\mathrm{v}}}\dfrac{\partial P_{\mathrm{v}}}{\partial T}$, $K_{\mathrm{ww}} = \rho_{\mathrm{liq}}\dfrac{K_0 K_{\mathrm{liq}}}{\eta_{\mathrm{liq}}} + \rho_{\mathrm{v}}\dfrac{K_0 K_{\mathrm{v}}}{\eta_{\mathrm{v}}}\dfrac{\partial P_{\mathrm{v}}}{\partial P_{\mathrm{liq}}}$, $K_{\mathrm{w}} = \rho_{\mathrm{liq}}\dfrac{K_0 K_{\mathrm{liq}}}{\eta_{\mathrm{liq}}}$; C_{v} – объемная теплоемкость; λ – теплопроводность; w – полное влагосодержание; K_{liq} – коэффициент относительной фазовой проницаемости жидкости; K_{v} – коэффициент относительной фазовой проницаемости пара; P_{v} – давление водяного пара.

Систему уравнений (1) целесообразно решать с помощью МКЭ относительно температуры и давления жидкости [1]. Таким образом, в указанном случае основная идея применения МКЭ состоит в том, что температура (T) и давление жидкости (P_{liq}) аппроксимируются полиномами [1, 7]:

$$T \approx \tilde{T} = \sum_{j=1}^{M} T_j N_j, \quad P_{\text{liq}} \approx \tilde{P}_{\text{liq}} = \sum_{j=1}^{M} P_{\text{liq}\,j} N_j, \tag{2}$$

где T_j и $P_{\text{liq}\,j}$ – значения температуры и давления жидкости в j-м узле; N_j – базисная функция в j-м узле (кусочно-непрерывная функция, определенная на конечном элементе); M – общее количество узлов.

Согласно методу взвешенных невязок в постановке Галеркина система уравнений (1) в общем виде запишется следующим образом [8]:

$$\begin{aligned}
&\frac{1}{\Delta t_n}\int_\Omega \begin{vmatrix} N_i C_{\text{v}} & 0 \\ N_i C_{\text{hv}} & N_i C_{\text{wp}} \end{vmatrix} \begin{vmatrix} T_j^{n+1} \\ P_{\text{liq}\,j}^{n+1} \end{vmatrix} \partial\Omega - \frac{1}{\Delta t_n}\int_\Omega \begin{vmatrix} N_i C_{\text{v}} & 0 \\ N_i C_{\text{hv}} & N_i C_{\text{wp}} \end{vmatrix} \begin{vmatrix} T_j^{n} \\ P_{\text{liq}\,j}^{n} \end{vmatrix} \partial\Omega - \\
&-\frac{1}{\Delta t_n}\int_\Omega \begin{vmatrix} N_i L\theta_{\text{liq}}^{n+1} \\ 0 \end{vmatrix} \partial\Omega + \frac{1}{\Delta t_n}\int_\Omega \begin{vmatrix} N_i L\theta_{\text{liq}}^{n} \\ 0 \end{vmatrix} \partial\Omega + (1-\gamma)\int_\Omega \begin{vmatrix} N_i L\rho_{\text{liq}} \nabla \upsilon_{\text{liq}}^{n} \\ 0 \end{vmatrix} \partial\Omega + \\
&+\gamma\int_\Omega \begin{vmatrix} N_i L\rho_{\text{liq}} \nabla \upsilon_{\text{liq}}^{n+1} \\ 0 \end{vmatrix} \partial\Omega + (1-\gamma)\int_\Omega \begin{vmatrix} N_i \nabla\lambda\nabla N_j & 0 \\ N_i \nabla K_{\text{hv}} \nabla N_j & N_i \nabla K_{\text{wv}} \nabla N_j \end{vmatrix} \begin{vmatrix} T_j^{n} \\ P_{\text{liq}\,j}^{n} \end{vmatrix} \partial\Omega + \\
&+\gamma\int_\Omega \begin{vmatrix} N_i \nabla\lambda\nabla N_j & 0 \\ N_i \nabla K_{\text{hv}} \nabla N_j & N_i \nabla K_{\text{wv}} \nabla N_j \end{vmatrix} \begin{vmatrix} T_j^{n+1} \\ P_{\text{liq}\,j}^{n+1} \end{vmatrix} \partial\Omega - \int_\Omega \begin{vmatrix} 0 \\ \nabla N_i K_{\text{w}} \rho_{\text{liq}} g \nabla D \end{vmatrix} \partial\Omega - \\
&-\int_\Gamma \begin{vmatrix} N_i \alpha T_\infty \\ 0 \end{vmatrix} \partial\Gamma + (1-\gamma)\int_\Gamma \begin{vmatrix} N_i \alpha N_j & 0 \\ 0 & 0 \end{vmatrix} \begin{vmatrix} T_j^{n} \\ P_{\text{liq}\,j}^{n} \end{vmatrix} \partial\Gamma + \gamma\int_\Gamma \begin{vmatrix} N_i \alpha N_j & 0 \\ 0 & 0 \end{vmatrix} \begin{vmatrix} T_j^{n+1} \\ P_{\text{liq}\,j}^{n+1} \end{vmatrix} \partial\Gamma - \\
&-\int_\Gamma \begin{vmatrix} N_i q_h \\ N_i (q_{\text{v}} + q_{\text{liq}}) \end{vmatrix} \partial\Gamma - \int_\Gamma \begin{vmatrix} N_i \varepsilon\sigma T_\infty^4 \\ 0 \end{vmatrix} \partial\Gamma + (1-\gamma)\int_\Gamma \begin{vmatrix} N_i \varepsilon\sigma {N_j}^4 & 0 \\ 0 & 0 \end{vmatrix} \begin{vmatrix} \left(T_j^{n}\right)^4 \\ P_{\text{liq}\,j}^{n} \end{vmatrix} \partial\Gamma + \\
&+\gamma\int_\Gamma \begin{vmatrix} N_i \varepsilon\sigma {N_j}^4 & 0 \\ 0 & 0 \end{vmatrix} \begin{vmatrix} \left(T_j^{n+1}\right)^4 \\ P_{\text{liq}\,j}^{n+1} \end{vmatrix} \partial\Gamma = 0.
\end{aligned} \tag{3}$$

2. Адаптация трехмерной математической модели для конечно-элементной реализации

Численное решение системы уравнений (3) в общем виде осуществляется методом Ньютона – Рафсона посредством подстановки аппроксимаций (2), что подробно рассматривается в [1]. Таким образом, практическая реализация математической модели (1) в трехмерной постановке сводится к расчету значений соответствующих параметров в каждом конечном элементе сетки, где в качестве элемента дискретизации берется тетраэдр с четырьмя узлами [7]. Матрица теплопроводности элемента будет иметь следующий вид:

$$\left[k^{(e)}\right] = \int_{V^{(e)}} \left[N^{(e)}\right]^T [D] \left[N^{(e)}\right] dV + \int_{S^{(e)}} h \left[B^{(e)}\right]^T \left[B^{(e)}\right], \tag{4}$$

где D – матрица, описывающая коэффициенты теплопроводности; B – матрица, содержащая функции формы.

Функции формы в трехмерном случае будут иметь вид

$$B_\beta = a_\beta + b_\beta x + c_\beta y + d_\beta z, \ \beta = i, j, k, l. \tag{5}$$

Подставив (5) в (4), получим

$$\left[k^{(e)}\right] = \frac{K_{xx}}{36V}\begin{bmatrix} b_{ii} & b_{ij} & b_{ik} & b_{il} \\ b_{ij} & b_{jj} & b_{jk} & b_{jl} \\ b_{ik} & b_{jk} & b_{kk} & b_{kl} \\ b_{il} & b_{jl} & b_{kl} & b_{ll} \end{bmatrix} + \frac{K_{yy}}{36V}\begin{bmatrix} c_{ii} & c_{ij} & c_{ik} & c_{il} \\ c_{ij} & c_{jj} & c_{jk} & c_{jl} \\ c_{ik} & c_{jk} & c_{kk} & c_{kl} \\ c_{il} & c_{jl} & c_{kl} & c_{ll} \end{bmatrix} + \\ + \frac{K_{zz}}{36V}\begin{bmatrix} d_{ii} & d_{ij} & d_{ik} & d_{il} \\ d_{ij} & d_{jj} & d_{jk} & d_{jl} \\ d_{ik} & d_{jk} & d_{kk} & d_{kl} \\ d_{il} & d_{jl} & d_{kl} & d_{ll} \end{bmatrix} + \frac{hS_{jkl}}{12}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 1 \\ 0 & 1 & 2 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}. \quad (6)$$

3. Архитектура программного модуля ПК SPS для параллельного решения задачи неизотермического влагопереноса

Решение уравнений математической физики с помощью МКЭ включает три этапа [9]:

– описание геометрии области решения, задание физических характеристик, генерация конечно-элементной сетки;

– решение с помощью МКЭ дифференциальных уравнений;

– визуализацию и интерпретацию полученных результатов.

Все эти этапы на программном уровне выполняются с помощью отдельных модулей (препроцессора, процессора и постпроцессора) (рис. 1). Для реализации параллельных алгоритмов, описанных в работах [10, 11], необходимо использовать несколько модулей процессоров.

Рис. 1. Параллельная архитектура программных средств, базирующихся на МКЭ

Особенность параллельных вычислительных алгоритмов неизотермического переноса [9] заключается в том, что количество процессоров в программной архитектуре соответствует количеству процессоров в компьютерной архитектуре. Программная реализация модулей препроцессора и постпроцессора при решении задач неизотермического переноса влаги и растворимых веществ не имеет принципиальных отличий от реализации этих модулей для решения других конечноэлементных задач по причине унификации ввода исходных данных и визуализации полученных результатов путем применения стандартных интерфейсов.

При программной реализации модуля, ответственного за решение уравнений неизотермического переноса влаги и растворимых веществ в соответствии с алгоритмом параллельных вычислений [10], необходимо решить следующие задачи:

– вычислить базисные функции и их производные в соответствии с выбранной квадратурной формулой;

– вычислить длину, площадь и объем для соответствующих типов конечных элементов;

– вычислить значения физических свойств в конечных элементах;

– создать распределенные массивы для вычисления локальных матриц конечных элементов;

– создать распределенные массивы для вычисления локального вектора невязки и якобиана;

– масштабировать и распределить части массива по *N* процессорам;

– реализовать сборку глобальных матриц;

– реализовать учет граничных условий.

Кроме того, необходимо иметь алгоритмы и подпрограммы для решения систем алгебраических уравнений, а для облегчения реализации вышеперечисленных задач – средства для выполнения алгебраических и вычислительных операций над матрицами.

Для решения приведенных выше задач создан программный модуль (рис. 2). Реализация программных средств осуществляется в среде Microsoft Visual Studio 2008 на языке C++ с использованием современных технологий объектно-ориентированного программирования.

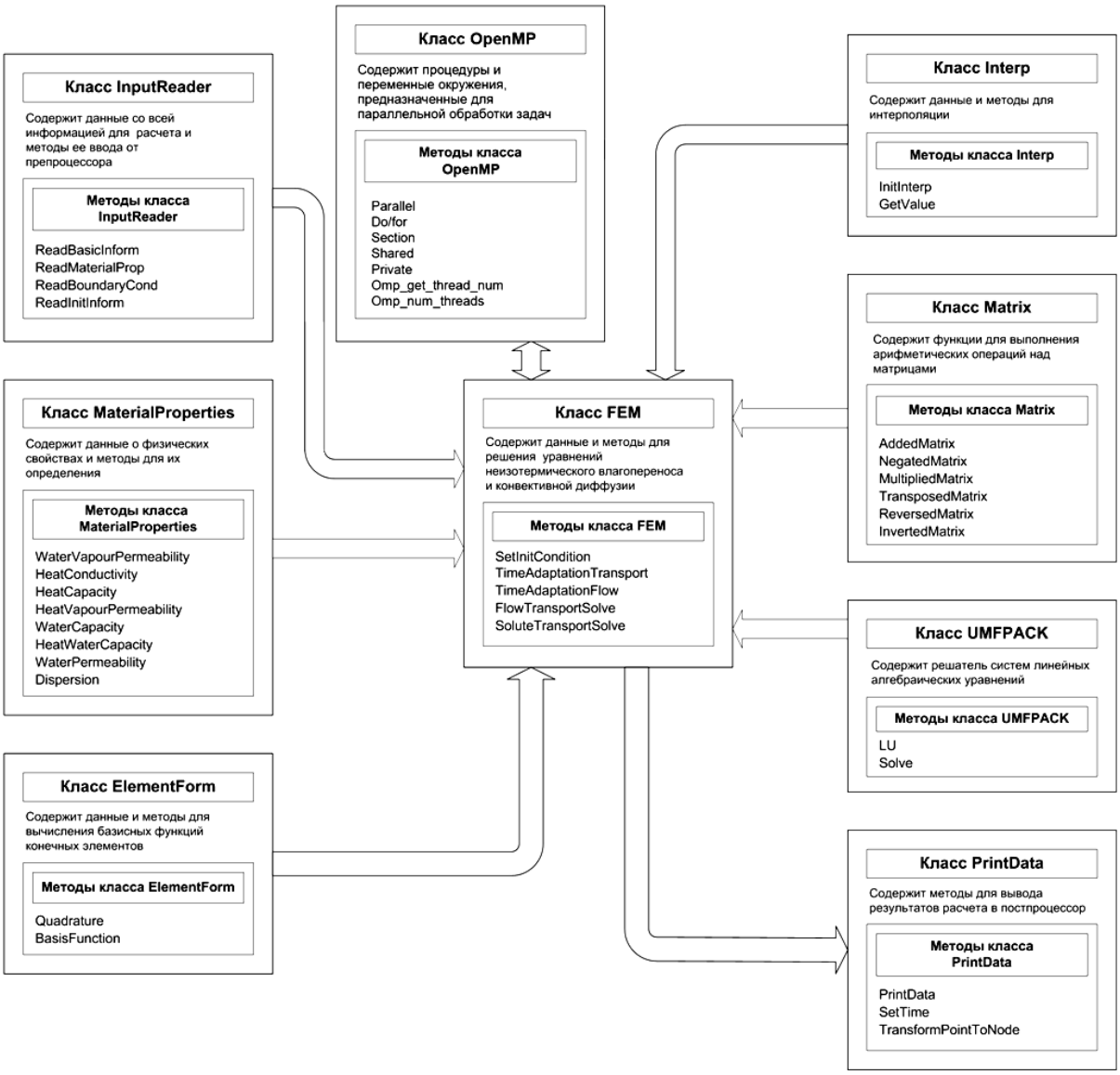


Рис. 2. Структура программного модуля для решения уравнений неизотермического переноса влаги и растворимых веществ

Разработанный программный модуль, реализующий алгоритм параллельного решения задачи неизотермического переноса влаги и растворимых веществ [11, 12], основан на следующей иерархии классов:

класс `FEM` – содержит методы для инициализации начальных условий, вычисления шага по времени в соответствии с разработанным алгоритмом [6], решения уравнений неизотермического переноса влаги и растворимых веществ на текущем временном шаге [6];

класс `InputReader` – содержит методы для передачи необходимых для решения задачи неизотермического переноса влаги и растворимых веществ данных от препроцессора в соответствующие объекты процессора;

класс `MaterialProperties` – содержит физические константы, информацию о количестве сред и их свойствах, методы для вычисления физических свойств в зависимости от температуры и давления жидкости и производных этих зависимостей;

класс `ElementForm` – включает данные и методы для вычисления базисных функций и их производных в соответствии с выбранной квадратурной формулой, а также вычисления длины, площади и объема для определенных типов конечных элементов;

класс `Interp` – содержит реализацию различных методов интерполяции: линейной и квадратичной интерполяции, интерполяции эрмитовыми полиномами и кубическими сплайнами;

в классе `Matrix` – используется свободно распространяемая библиотека NEWMAT, в которой реализованы функции для выполнения арифметических операций над матрицами;

класс `UMFPACK` – содержит свободно распространяемую библиотеку UMFPACK, в которой реализованы процедуры хранения разряженных матриц больших систем алгебраических уравнений, а также прямые методы их решения. При этом в зависимости от вида правой матрицы системы алгебраических уравнений: симметричная, диагональная, комплексная и т. п. – автоматически выбирается и осуществляется оптимальный прямой метод решения;

класс `PrintData` – содержит методы для передачи результатов решения задачи неизотермического переноса влаги и растворимых веществ в соответствующие объекты постпроцессора;

класс `OpenMP` – описывает совокупность директив компилятора, библиотечных процедур и переменных окружения, которые предназначены для программирования многопоточных приложений на многопроцессорных системах с общей памятью.

Управление расчетом и передача данных между задачами неизотермического движения влаги и переноса растворимых веществ осуществляются путем взаимодействия головной программы femHWStransport и класса OpenMP.

Особенностью предложенной иерархии классов является возможность использования:

– широкого класса методов интерполяции функциональных зависимостей [8];

– различных квадратурных формул численного интегрирования [8];

– параллельных методов решения систем алгебраических уравнений, наилучшим образом соответствующих получаемому типу матриц.

На основании разработанной иерархии классов и вычислительных алгоритмов [5, 11] создан соответствующий программный модуль, который позволяет получать конечно-элементную сетку с решением с помощью центрального процессора, совместимую с ПК Comsol Multiphysics [4]. Однако полученное решение является более устойчивым, чем решение, найденное в ПК Comsol Multiphysics [2, 13].

Заключение

Разработана трехмерная численная модель миграции загрязняющих веществ в природных дисперсных средах, основой которой является решение задачи неизотермического тепловлагопереноса. Компьютерная реализация модели осуществлена в ПК SPS с применением технологии параллельных вычислений. Для этих целей разработаны структура программного модуля и иерархия классов, особенностями которых являются:

– возможность реализации параллельных вычислений на максимально возможном количестве ядер процессора;

– наличие различных квадратурных формул численного интегрирования;

– автоматический выбор метода решения системы алгебраических уравнений, наилучшим образом соответствующего получаемому типу матрицы.

На основании предложенной иерархии классов и оригинальных алгоритмов создан программный модуль, позволяющий осуществлять долгосрочное прогнозирование для решения трехмерной задачи неизотермического тепловлагопереноса загрязняющих веществ в почве.

Список литературы

1. Компьютерное моделирование миграции загрязняющих веществ в природных дисперсных средах / С.П. Кундас [и др.]; под общ. ред. С.П. Кундаса. – Минск : МГЭУ им. А.Д. Сахарова, 2011. – 212 с.

2. Кундас, С.П. Моделирование миграции примесей в почве с использованием математического пакета FEMLAB / С.П. Кундас, И.А. Гишкелюк, В.И. Коваленко // Инженерный вестник. – 2006. – № 1 (21) / 3. – С. 203–206.

3. ANSYS Theory Manual. ANSYS Release. – SAS IP, Inc., 2001. – 1266 p.

4. COMSOL Multiphysics. User's Guide. – COMSOL AB, 2007. – 588 p.

5. MSC/NASTRAN Numerical Methods. User's Guide. – MSC, 1998. – 297 p.

6. Шалькевич, П.К. Алгоритм параллельных вычислений задачи неизотермического влагопереноса в природных дисперсных средах / П.К. Шалькевич, С.П. Кундас, И.А. Гишкелюк // Доклады БГУИР. – 2014. – № 5 (83). – С. 90–94.

7. Шалькевич, П.К. Параллельное вычисление задачи взаимосвязанного тепловлагопереноса на суперкомпьютерах с различным числом ядер / П.К. Шалькевич, С.П. Кундас // Сахаровские чтения 2014 года: экологические проблемы XXI века : материалы 14-й Междунар. науч. конф., Минск, 29–30 мая 2014 г. – Минск, 2014. – С. 221–222.

8. Сегерлинд, Л. Применение метода конечных элементов / Л. Сегерлинд; пер. с англ. – М. : Мир, 1979. – 392 с.

9. Математическое моделирование процессов переноса вещества и влаги в почве / С.П. Кундас, И.А. Гишкелюк [и др.] // Экологический вестник. – 2007. – № 1. – С. 62–72.

10. Шалькевич, П.К. Реализация алгоритма параллельных вычислений задачи неизотермического влагопереноса в природных дисперсных средах / П.К. Шалькевич, С.П. Кундас, И.А. Гишкелюк // Информатика. – 2014. – № 4 (44). – С. 44–51.

11. Сабоннадьер, Ж.К. Метод конечных элементов и САПР / Ж.К. Сабоннадьер, Ж.Л. Кулон. – М. : Мир, 1989. – 190 с.

12. Kundas, S. Application of computer modeling for analysis and forecasting of radionuclide's migration in soil / S. Kundas, V. Kovalenko, I. Gishkeluk // J. of the University of Applied Sciences Mittweida (Germany). – 2006. – № 10. – P. 44–49.

13. Моделирование процессов термовлагопереноса в капиллярно-пористых средах / С.П. Кундас [и др.]. – Минск : ИТМО НАН Беларуси, 2007. – 292 с.

Поступила 17.11.2014

[1]*Международный государственный экологический университет им. А. Д. Сахарова,*
Минск, ул. Долгобродская, 23
e-mail: pavel.shalkevich@gmail.com,
gishkeluk@iseu.by

[2]*Белорусский национальный технический университет,*
Минск, пр. Победителей, 65
e-mail: kundas@tut.by

P.K. Shalkevich, S.P. Kundas, I.A. Gishkeluk

PARALLEL COMPUTING IN THE HEAT AND MOISTURE TRANSFER USING SPS SOFTWARE

A numerical model of non-isothermal moisture transfer in soil is developed and adapted for the finite element solution of the three-dimensional heat and moisture transfer in natural dispersed media. An SPS software module which allows carrying out long-term forecasting for the solution of three-dimensional non-isothermal heat and moisture transfer of contaminants in soil is developed.

УДК 681.325

Л.Д. Черемисинова

МНОГОКРАТНАЯ СВЕРТКА РЕГУЛЯРНЫХ СТРУКТУР НА ОСНОВЕ РЕШЕНИЯ ЛОГИЧЕСКИХ УРАВНЕНИЙ

Рассматривается задача минимизации площади регулярных матричных структур заказных СБИС методом многократной свертки. Предлагается метод решения ключевой проблемы многократной свертки – проверки реализуемости множества свертки, который основывается на сведении задачи к решению логического уравнения и проверке выполнимости конъюнктивной нормальной формы.

Введение

Проектирование блоков управляющей логики заказных СБИС ориентировано, как правило, на использование схем с регулярной структурой. Их применение позволяет уменьшить стоимость разработки схемы в силу упрощения задачи генерации топологии по структурному (или функциональному) описанию реализуемого устройства. Регулярная матричная структура состоит из взаимно пересекающихся строк и столбцов, на пересечении которых находятся транзисторы (или схемы из транзисторов). Примерами двухмерных регулярных матричных структур являются структуры типа программируемых логических матриц, матриц Вайнбергера, транзисторных матриц, регулярных схем на базе последовательно соединенных МОП-транзисторов [1–3]. За основной критерий оптимальности при проектировании СБИС на основе регулярных структур принимается площадь кристалла, оцениваемая на логическом уровне произведением чисел столбцов и строк.

Существенным недостатком матричных структур является то, что они, имея регулярную организацию, проигрывают многоуровневым реализациям на основе произвольной логики по площади, занимаемой на кристалле, за счет неэффективного ее использования. Последнее выражается в сильной разреженности матричных структур. Одним из наиболее эффективных методов топологической оптимизации является свертка столбцов и строк матричной структуры, сокращающая число неиспользуемых транзисторов. История развития методов свертки берет начало в работах [4, 5], обзор основных подходов к свертке регулярных структур можно найти в [6, 7].

Свертка основана на разрыве шин матричной структуры и реализации на одной вертикальной (и (или) горизонтальной) шине двух (при простой свертке) или более (при многократной свертке) столбцов и (или) строк. При простой свертке вертикальная (и (или) горизонтальная) шина свернутой структуры разделяется на два сегмента, каждый из которых реализует один столбец (строку) исходной матричной структуры. При многократной свертке число свертываемых столбцов и (или) строк (число сегментов) не ограничивается. Очевидно, что преобразование матричной структуры с использованием свертки не изменяет функциональность схемы. В теоретическом плане задача свертки сводится к комбинаторной задаче поиска оптимального переупорядочивания и совмещения столбцов и (или) строк матричных структур и является, как доказано в [4], NP-трудной.

Центральной частью любой регулярной структуры (ее ядром) является двухмерная матрица. Ее столбцы представляют собой линии подвода входных сигналов, а строки реализуют конъюнкции входных переменных, соответствующих тем столбцам, в которых на пересечении со строками находятся транзисторы. В настоящей работе рассматривается задача сокращения площади такой двухмерной матрицы путем ее многократной свертки. Предлагается решение ключевой задачи многократной свертки – проверки множества свертки на реализуемость.

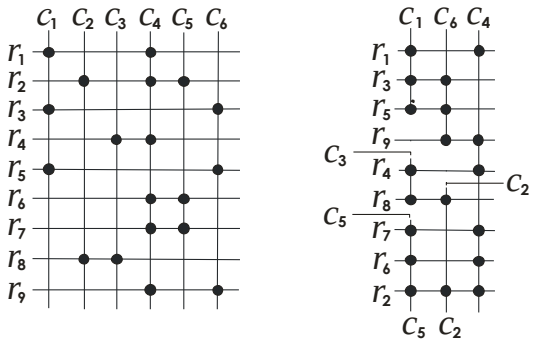
В работе [8] к задаче булевой выполнимости впервые была сведена проблема трассировки соединений. Позднее аналогичным образом к задаче выполнимости была сведена проблема простой столбцовой свертки программируемой логической матрицы [9]. В этих работах решаемые задачи сводились к проверке выполнимости булевой функции, вид которой определялся данными задачами. Для представления анализируемой булевой функции и анализа ее на выполнимость использовался аппарат BDD (Binary Decision Diagrams – диаграммы двоичных решений) [10].

В работе [11] к проверке выполнимости сведена, так же как и в [9], задача анализа множества простой свертки на реализуемость. В отличие от упомянутой работы решение этой задачи формулируется как поиск выполняющего набора для специальным образом формируемой конъюнктивной нормальной формы (КНФ), задающей условия, предъявляемые к реализуемому множеству свертки. Такое представление позволяет использовать для поиска выполняющего набора SAT-решатели (SAT-solver) [12–14], которые обеспечивают проверку выполнимости достаточно сложных КНФ, содержащих до тысячи переменных. Кроме того, задача простой свертки в работе [11] рассматривалась в более общей постановке – для случая неупорядоченных свертываемых наборов.

В настоящей работе рассмотрен более общий вид свертки – многократная свертка регулярных матричных структур заказных СБИС. Показано, как свести анализ множества свертки на реализуемость к задаче поиска решения логического уравнения и проверке выполнимости КНФ.

1. Формализация задачи многократной свертки

Во всех постановках задачи свертки математической моделью регулярной структуры является булева матрица $\boldsymbol{B}$, имеющая множества $C(\boldsymbol{B}) = \{c_1, c_2, \ldots\}$ столбцов и $R(\boldsymbol{B}) = \{r_1, r_2, \ldots\}$ строк. Каждый столбец $c_j \in C(\boldsymbol{B})$ (строка $r_j \in R(\boldsymbol{B})$) порождает множество $R(c_j)$ строк ($C(r_j)$ столбцов), имеющих единицы на пересечении с этим столбцом, т. е. $r_i \in R(c_j)$, если $b^j_i \in \boldsymbol{B}$ имеет значение 1. Площадь матричной структуры определяется как $S = |C(\boldsymbol{B})|\,|R(\boldsymbol{B})|$. Далее будем говорить о столбцовой свертке, имея в виду, что все сказанное может быть отнесено и к строчной свертке.



а) *б)*

Схематичное представление матричной структуры: *а)* исходная структура;
б) ее многократная свертка

На рисунке точками обозначены транзисторы на пересечениях некоторых строк и столбцов. Формальной моделью этой матричной структуры является следующая булева матрица:

$$\boldsymbol{B} = \begin{array}{cccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & \\ 1 & 0 & 0 & 1 & 0 & 0 & r_1 \\ 0 & 1 & 0 & 1 & 1 & 0 & r_2 \\ 1 & 0 & 0 & 0 & 0 & 1 & r_3 \\ 0 & 0 & 1 & 1 & 0 & 0 & r_4 \\ 1 & 0 & 0 & 0 & 0 & 1 & r_5 \\ 0 & 0 & 0 & 1 & 1 & 0 & r_6 \\ 0 & 0 & 0 & 1 & 1 & 0 & r_7 \\ 0 & 1 & 1 & 0 & 0 & 0 & r_8 \\ 0 & 0 & 0 & 1 & 0 & 1 & r_9 \end{array}. \qquad (1)$$

Непересекающиеся столбцы c_k и c_l ($R(c_k) \cap R(c_l) = \varnothing$) не имеют активных транзисторов на пересечении с одними и теми же строками и являются строчно совместимыми. Совместимые

столбцы c_k и c_l образуют свертываемую пару (c_k, c_l), т. е. могут быть свернуты и реализованы на одной вертикальной шине регулярной структуры. Аналогично набор $l_k = (c_{k1}, c_{k2},\ldots, c_{km})$, состоящий из попарно совместимых столбцов c_{ij}, называется свертываемым (если не наложены никакие дополнительные ограничения на тип свертки). Только такие наборы (или пары) столбцов могут быть свернуты. Неупорядоченный свертываемый набор $l_k = (c_{k1}, c_{k2},\ldots, c_{km})$ порождает $m!$ различных упорядоченных наборов $l_k^o = <c_{k1}, c_{k2},\ldots, c_{km}>$. Под упорядочением набора l_k понимается выбор одного из этих упорядоченных наборов l_k^o. Любой упорядоченный свертываемый набор столбцов l_k^o может быть реализован на одной вертикальной шине свернутой двухмерной структуры, на которой c_{k1} помещается над c_{k2}, c_{k2} – над c_{k3} и т. д.: $c_{k(m-1)}$ – над c_{km}. Итак, набор l_k^o порождает такое переупорядочение строк матрицы, что $R(c_{k1}) > R(c_{k2})$ – строки из $R(c_{k1})$ располагаются выше всех строк из $R(c_{k2})$; $R(c_{k2}) > R(c_{k2})$ – строки из $R(c_{k2})$ выше всех строк из $R(c_{k3})$ и т. д.: $R(c_{k(m-1)}) > R(c_{km})$, порождая отношение на множестве $R(\boldsymbol{B})$:

$$T(l_k^o) = \bigcup_{i,j} (R(c_{ki}) \times R(c_{kj})), \quad \text{т. е.} \quad T(l_k^o) = \{ r_p \times r_q / r_p \in R(c_{ki}), r_q \in R(c_{kj}), i < j\}.$$

Это отношение является отношением частичного порядка на множестве $R(\boldsymbol{B})$, так как оно по определению иррефлексивно, асимметрично и транзитивно.

Неупорядоченный свертываемый набор $l_k = (c_{k1}, c_{k2},\ldots, c_{km})$ порождает $m!$ возможных отношений частичного порядка на множестве $R(\boldsymbol{B})$. Для матричной структуры (рисунок, *а*), описываемой булевой матрицей $\boldsymbol{B}$ (1), существует, например, набор, состоящий из трех попарно совместимых столбцов, $l_1 = (c_1, c_3, c_5)$, который порождает шесть упорядоченных свертываемых наборов

$$l_1^o = <c_1, c_3, c_5>,\ l_2^o = <c_1, c_5, c_3>,\ l_3^o = <c_3, c_1, c_5>,\ l_4^o = <c_3, c_5, c_1>,\ l_5^o = <c_5, c_1, c_3>,\ l_6^o = <c_5, c_3, c_1>$$

и, соответственно, шесть частичных упорядочиваний на множестве строк. Первый упорядоченный свертываемый набор $l_1^o = <c_1, c_3, c_5>$ порождает, например, следующий частичный порядок на множестве строк:

$$\{r_1, r_3, r_5\} > \{r_4, r_8\},\ \{r_4, r_8\} > \{r_2, r_6, r_7\}.$$

Два свертываемых набора l_p и l_q совместимы, если они не пересекаются, т. е. если все c_{pi}, c_{qj} (для всех i, j) различны. Множество $L = \{l_1, l_2,\ldots, l_n\}$, состоящее из попарно совместимых свертываемых наборов столбцов, называется неупорядоченным множеством столбцовой свертки (НМС). Аналогично определяются упорядоченные множества свертки (УМС): $L_p^o = \{ l_{p1}^o, l_{p2}^o, ..., l_{pn}^o \}$. Число столбцов, входящих во все свертываемые наборы множества свертки L (или L_p^o), называется ее размером. Задача свертки состоит в поиске наибольшего реализуемого множества свертки, которое порождает свернутую матричную структуру наименьшей площади.

УМС L_p^o порождает отношение порядка на множестве строк $R(\boldsymbol{B})$, которое представляет собой объединение отношений $T(l_k^o)$, порождаемых всеми свертываемыми наборами $l_i^o \in L_p^o$:

$$T(L_p^o) = \bigcup_{i=1}^{n} (T(l_{pi}^o)).$$

Отношение $T(L_p^o)$ в общем случае не есть отношение частичного порядка, так как оно иррефлексивно, асимметрично, но не обязательно транзитивно, но транзитивное замыкание $T^r(L_p^o)$ отношения $T(L_p^o)$ иррефлексивно, транзитивно, но не обязательно асимметрично.

В работе [5] показано, что упорядоченное множество L_p^o свертки реализуемо, если порождаемое им транзитивное замыкание $T^r(L_p^o)$ есть отношение частичного порядка на множестве $R(\boldsymbol{B})$. Это означает, что $T^r(L_p^o)$ асимметрично. Обобщив это определение реализуемости на неупорядоченное множество свертки, будем называть НМС L реализуемым, если существует реализуемое УМС L_p^o, получаемое из L путем упорядочения входящих в него неупорядоченных наборов l_{ki}.

Для рассматриваемого примера матричной структуры существуют четыре неупорядоченных максимальных свертываемых набора:

$$l_1 = (c_1, c_3, c_5),\ l_2 = (c_1, c_2),\ l_3 = (c_2, c_6),\ l_4 = (c_3, c_6).$$

Два набора совместимы и образуют множество свертки максимального размера: $L = \{(c_1, c_3, c_5), (c_2, c_6)\}$. НМС L порождают 12 УМС L_p^o, так как для l_1 возможно шесть упорядочиваний (как показано выше), а для l_2 – два: $<c_2, c_6>$ и $<c_6, c_2>$. Первое УМС $L_1^o = \{<c_1, c_3, c_5>, <c_2, c_6>\}$ порождает следующее отношение на $R(\boldsymbol{B})$:

$$T(L_1^o) = R(c_1) \times R(c_3) \cup R(c_3) \times R(c_5) \cup R(c_2) \times R(c_6) =$$

$$= \{r_1, r_3, r_5\} \times \{r_4, r_8\} \cup \{r_4, r_8\} \times \{r_2, r_6, r_7\} \cup \{r_2, r_8\} \times \{r_3, r_5, r_9\}.$$

Его транзитивное замыкание содержит конфликтные пары (r_3, r_2) и (r_2, r_3), откуда следует, что отношение $T^r(L_1^o)$ не является отношением частичного порядка на $R(\boldsymbol{B})$ и УМС L_1^o нереализуемо. Второе УМС $L_2^o = \{<c_1, c_3, c_5>, <c_6, c_2>\}$ порождает отношение

$$T(L_2^o) = R(c_1) \times R(c_3) \cup R(c_3) \times R(c_5) \cup R(c_6) \times R(c_2) =$$

$$= \{r_1, r_3, r_5\} \times \{r_4, r_8\} \cup \{r_4, r_8\} \times \{r_2, r_6, r_7\} \cup \{r_3, r_5, r_9\} \times \{r_2, r_8\}.$$

Его транзитивное замыкание $T^r(L_2^o)$ обладает свойством асимметричности. Следовательно, УМС L_2^o (а значит, и НМС L) является реализуемым множеством свертки.

Реализуемое множество свертки $L_p^o = \{l_{p1}, l_{p2}, \ldots, l_{pn}\}$ определяет вид свернутой матричной структуры, а его размер – размер свертки: число наборов свертки соответствует числу вертикальных шин свернутой матрицы, которые заменят $\sum_{i=1}^{n} |l^o_{pi}|$ столбцов исходной матричной структуры. Например, множество свертки $L_2^o = \{<c_1, c_3, c_5>, <c_6, c_2>\}$ состоит из двух свертываемых наборов и имеет размер пять. Следовательно, пять столбцов исходной матричной структуры будут заменены двумя столбцами при ее свертке.

Задача свертки матричной структуры формулируется следующим образом: дана булева матрица, ее представляющая, требуется найти реализуемое множество свертки максимального размера. Существуют методы построения множеств многократной свертки. Например, в работах [5, 15] находятся упорядоченные множества свертки, в [16] – неупорядоченные (их существенно меньше). Задача свертки в [17] сведена к поиску максимальных клик графа отношения совместимости столбцов матрицы $\boldsymbol{B}$.

В настоящей работе рассматривается задача проверки на реализуемость и упорядочивание неупорядоченного множества свертки путем сведения этих задач к поиску решения логического уравнения с использованием известных SAT-решателей. При решении задачи поиска реализуемого множества свертки максимального размера предлагаемый метод используется для последовательной проверки реализуемости множеств свертки, упорядоченных в порядке убывания их размеров.

2. Проверка реализуемости упорядоченного множества свертки путем сведения к поиску корня логического уравнения

Задача проверки реализуемости множества свертки формулируется следующим образом: дано упорядоченное множество столбцовой свертки $L^o = \{l_1^o, l_2^o, \ldots, l_n^o\}$; требуется определить, реализуемо ли оно, и если да, то предъявить частичное упорядочение на множестве строк $R(\boldsymbol{B})$. Каждый свертываемый набор $l_k^o = <c_{k1}, c_{k2}, \ldots, c_{km}> \in L^o$ порождает следующее частичное упорядочение строк из $R(\boldsymbol{B})$:

$$(R(c_{k1}) > R(c_{k2})) \cap (R(c_{k2}) > R(c_{k3})) \cap \ldots \cap (R(c_{k(m-1)}) > R(c_{km})). \qquad (2)$$

Условие (2) говорит о том, что строки из $R(c_{k1})$ должны располагаться в свернутой структуре выше строк из $R(c_{k2})$, а они – выше строк из $R(c_{k3})$ и т. д.

Задача проверки реализуемости УМС $L^o = \{l_1^o, l_2^o, \ldots, l_n^o\}$ должна установить, существует ли такое упорядочение строк, для которого выполняется условие (2). Покажем, как эту задачу можно свести к решению сначала логического уравнения, а затем к решению задачи выполнимости КНФ.

Сформируем множество R_L строк, каждая из которых входит по крайней мере в одно из множеств $R(c_{ij})$ ($c_{ij} \in l_i^o$ из L^o), $i \in \{1, 2, \ldots, n\}$, $j \in \{1, 2, \ldots, i_m\}$. Расположение строк из $R(\boldsymbol{B}) \setminus R_L$ в свертываемой согласно L^o регулярной матричной структуре никак не влияет на свертываемость матричной структуры относительно столбцов из наборов множества L^o. Следовательно, допустимо рассматривать далее только строки из R_L. Более того, нет нужды рассматривать и те строки из R_L, каждая из которых входит только в одно из множеств $R(c_{ij})$, так как порядок их следования фиксируется относительно строк только одного множества $R(c_{ij})$ и, следовательно, исключаются конфликты с размещением строк из других множеств $R(c_{sq})$ ($s \neq i$). Таким путем можно сократить число переменных и термов формируемого логического уравнения.

Для задания отношений следования строки из R_L кодируются позиционным кодом $x_1x_2\ldots x_q$, где $q = \lceil \log_2 |R_L| \rceil$ ($\lceil t \rceil$ – наименьшее сверху целое, не меньшее, чем t). Численное значение кода $x^i{}_1x^i{}_2\ldots x^i{}_q$ строки r_i задает ее порядковый номер, указывающий физическое расположение горизонтальной линии свернутой матричной структуры, выделяемой для строки r_i. Тогда условия, порождаемые упорядочениями строк $r_i > r_j$, представляются функциями

$$f_{ij} = (x^i{}_q x^i{}_{q-1}\ldots x^i{}_1) > (x^j{}_q x^j{}_{q-1}\ldots x^j{}_1). \tag{3}$$

С учетом (3) условие (2), порождаемое набором свертки $l_k^o = \langle c_{k1}, c_{k2}, \ldots, c_{km}\rangle$ и накладываемое на порядок следования строк, представляется в виде

$$\bigcap_{q=1}^{m-1} \Big(\bigcap_{r_i \in R(c_{kq})} \Big(\bigcap_{r_j \in R(c_{k(q+1)})} f_{ij} \Big) \Big) = 1. \tag{4}$$

УМС $L^o = \{l_1^o, l_2^o, \ldots, l_n^o\}$ является реализуемым, если и только если следующее уравнение имеет хотя бы одно решение:

$$\bigcap_{k=1}^{n} \Big(\bigcap_{q=1}^{m-1} \Big(\bigcap_{r_i \in R(c_{kq})} \Big(\bigcap_{r_j \in R(c_{k(q+1)})} f_{ij} \Big) \Big) \Big) = 1. \tag{5}$$

Если уравнение (5) имеет корень (набор значений переменных $x^i{}_l$, выполняющих его), то соответствующее УМС L^o реализуемо. Этот корень определяет порядковые номера строк из R_L. В противном случае, если корня уравнения (5) не существует, УМС L^o нереализуемо.

Для примера рассмотрим УМС $L_2{}^o = \{\langle c_1, c_3, c_5\rangle, \langle c_6, c_2\rangle\}$ для приведенной выше матричной структуры. Так как $R(c_1) = \{r_1, r_3, r_5\}$, $R(c_3) = \{r_4, r_8\}$, $R(c_5) = \{r_2, r_6, r_7\}$, $R(c_2) = \{r_2, r_8\}$, $R(c_6) = \{r_3, r_5, r_9\}$, то множество строк, подлежащих упорядочиванию согласно УМС $L_2{}^o$, $R_L = \{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8\}$, можно сократить до $R_L = \{r_2, r_3, r_5, r_8\}$. Соответственно сокращается и число рассматриваемых строк в множествах $R(c_i)$: $R'(c_1) = \{r_3, r_5\}$, $R'(c_3) = \{r_8\}$, $R'(c_5) = \{r_2\}$, $R'(c_2) = \{r_2, r_8\}$, $R'(c_6) = \{r_3, r_5\}$. Четыре строки из R_L можно закодировать двумя булевыми переменными x_2 и x_1, их значения в кодах строк r_i будем обозначать через $x_2{}^i$ и $x_1{}^i$. После кодирования строк уравнение (5) принимает вид

$$\begin{gathered}(f_{3,8} \wedge f_{5,8} \wedge f_{8,2}) \wedge (f_{3,2} \wedge f_{3,8} \wedge f_{5,2} \wedge f_{5,8}) = f_{3,8} \wedge f_{5,8} \wedge f_{8,2} \wedge f_{3,2} \wedge f_{5,2} = \\ = (x^3{}_2x^3{}_1 > x^8{}_2x^8{}_1) \wedge (x^5{}_2x^5{}_1 > x^8{}_2x^8{}_1) \wedge (x^8{}_2x^8{}_1 > x^2{}_2x^2{}_1) \wedge (x^3{}_2x^3{}_1 > x^2{}_2x^2{}_1) \wedge (x^5{}_2x^5{}_1 > x^2{}_2x^2{}_1).\end{gathered} \tag{6}$$

3. Приведение уравнения для проверки реализуемости упорядоченного множества свертки к виду КНФ

Для решения уравнения (5) можно привлечь аппарат решения логических уравнений, но для этого необходимо представить функции $f_{ij} = ((x^i{}_p x^i{}_{p-1}\ldots x^i{}_1) > (x^j{}_p x^j{}_{p-1}\ldots x^j{}_1))$ в виде формул

булевой алгебры. Для решения полученного уравнения можно привлечь известный аппарат проверки выполнимости КНФ, используемый в настоящее время в области автоматизации логического проектирования для решения задач большой размерности (например, Chaff [12], BerkMin [13]). Для этого необходимо представить левую часть уравнения (5), и в частности функции f_{ij}, в виде КНФ. КНФ представляет собой конъюнкцию дизъюнктов, каждый из которых задается дизъюнкцией литералов (булевых переменных и их инверсий).

Для приведения левой части уравнения (5) к виду КНФ необходимо преобразовать к виду КНФ функцию $f_{ij} = (x_q^i x_{q-1}^i \ldots x_1^i) > (x_q^j x_{q-1}^j \ldots x_1^j)$ (3). Найдем представление этой функции в виде минимальной КНФ $C(f_{ij})$. Для случая $q = 1$ (имеются две строки и для их кодирования используется только одна переменная) получается функция $f_{ij} = (x_1^i) > (x_1^j)$ от двух аргументов, ее минимальная КНФ имеет два дизъюнкта:

$$C^1(f_{ij}) = x_1^i \, \bar{x}_1^j.$$

Для случая $q = 2$ (имеются не более чем четыре строки и для их кодирования используются две переменные) получается функция $f_{ij} = (x_2^i x_1^i) > (x_2^j x_1^j)$ от четырех аргументов, ее минимальная КНФ имеет пять дизъюнктов:

$$C^2(f_{ij}) = (x_2^i \vee x_1^i) \wedge (\bar{x}_2^j \vee x_1^i) \wedge (x_2^i \vee \bar{x}_1^j) \wedge (\bar{x}_2^j \vee \bar{x}_1^j) \wedge (x_2^i \vee \bar{x}_2^j).$$

В общем случае КНФ $C^t(f_{ij})$ (для $q = t$) имеет $2t$ аргументов: на два аргумента (x_t^i и x_t^j) больше, чем КНФ $C^{t-1}(f_{ij})$. Если КНФ $C^{t-1}(f_{ij})$ имеет s^{t-1} дизъюнктов, то КНФ $C^t(f_{ij})$ – $s^t = 2s^{t-1}+1$ дизъюнктов. В силу регулярности функции f_{ij} относительно числа аргументов нетрудно показать, что КНФ $C^t(f_{ij})$ может быть получена из КНФ $C^{t-1}(f_{ij})$ путем преобразования каждого ее r-го дизъюнкта d_r^{t-1} в два дизъюнкта КНФ $C^t(f_{ij})$ – $(x_{t+1}^i \vee d_r^t)$ и $(\bar{x}_{t+1}^j \vee d_r^t)$ и добавления в полученную КНФ одного нового дизъюнкта $(x_{t+1}^i \vee \bar{x}_{t+1}^j)$, зависящего только от вновь введенных аргументов.

После представления функции f_{ij} в виде КНФ $C(f_{ij})$ уравнение (5) преобразуется в логическое уравнение, левая часть которого представляет собой КНФ:

$$\bigcap_{k=1}^{n} \Big(\bigcap_{q=1}^{m-1} \Big(\bigcap_{r_i \in R(c_{kq})} \Big(\bigcap_{r_j \in R(c_{k(q+1)})} C(f_{ij}) \Big) \Big) \Big) = 1. \qquad (7)$$

Таким образом, задача проверки реализуемости УМС $L^o = \{l_1^o, l_2^o, \ldots, l_n^o\}$ сведена к задаче поиска корня уравнения (7) путем проверки выполнимости КНФ его левой части с предъявлением выполняющего набора значений переменных (если решение существует).

Для пояснения сути изложенного метода продолжим процесс проверки реализуемости УМС $L_2^o = \{<c_1, c_3, c_5>, <c_6, c_2>\}$. После преобразования к виду КНФ левой части уравнения (6) проверка реализуемости УМС L_2^o сводится к проверке выполнимости следующей КНФ:

$$\begin{aligned}
&(x^3{}_2 x^3{}_1 > x^8{}_2 x^8{}_1) \wedge (x^5{}_2 x^5{}_1 > x^8{}_2 x^8{}_1) \wedge (x^8{}_2 x^8{}_1 > x^2{}_2 x^2{}_1) \wedge (x^3{}_2 x^3{}_1 > x^2{}_2 x^2{}_1) \wedge (x^5{}_2 x^5{}_1 > x^2{}_2 x^2{}_1) = \\
&= ((x^3{}_2 \vee x^3{}_1) \wedge (\bar{x}^8{}_2 \vee x^3{}_1) \wedge (x^3{}_2 \vee \bar{x}^8{}_1) \wedge (\bar{x}^8{}_2 \vee \bar{x}^8{}_1) \wedge (x^3{}_2 \vee \bar{x}^8{}_2)) \wedge \\
&\wedge ((x^5{}_2 \vee x^5{}_1) \wedge (\bar{x}^8{}_2 \vee x^5{}_1) \wedge (x^5{}_2 \vee \bar{x}^8{}_1) \wedge (\bar{x}^8{}_2 \vee \bar{x}^8{}_1) \wedge (x^5{}_2 \vee \bar{x}^8{}_2)) \wedge \\
&\wedge ((x^8{}_2 \vee x^8{}_1) \wedge (\bar{x}^2{}_2 \vee x^8{}_1) \wedge (x^8{}_2 \vee \bar{x}^2{}_1) \wedge (\bar{x}^2{}_2 \vee \bar{x}^2{}_1) \wedge (x^8{}_2 \vee \bar{x}^2{}_2)) \wedge \\
&\wedge ((x^3{}_2 \vee x^3{}_1) \wedge (\bar{x}^2{}_2 \vee x^3{}_1) \wedge (x^3{}_2 \vee \bar{x}^2{}_1) \wedge (\bar{x}^2{}_2 \vee \bar{x}^2{}_1) \wedge (x^3{}_2 \vee \bar{x}^2{}_2)) \wedge \\
&\wedge ((x^5{}_2 \vee x^5{}_1) \wedge (\bar{x}^2{}_2 \vee x^5{}_1) \wedge (x^5{}_2 \vee \bar{x}^2{}_1) \wedge (\bar{x}^2{}_2 \vee \bar{x}^2{}_1) \wedge (x^5{}_2 \vee \bar{x}^2{}_2)).
\end{aligned} \qquad (8)$$

Набором значений переменных, выполняющим КНФ (8), является, например, $x^3{}_2 = x^3{}_1 = x^5{}_2 = x^8{}_1 = 1$; $x^5{}_1 = x^8{}_2 = x^2{}_2 = x^2{}_1 = 0$. Таким образом, строки из $R_L = \{r_3, r_5, r_8, r_2\}$ получают следующие коды: r_3 – 11, r_5 – 10, r_8 – 01, r_2 – 00. Это решение задает частичное упорядочение r_3, r_5, r_8, r_2 строк из R_P и, например, следующее полное упорядочение r_1, r_3, r_5, r_9 r_4, r_8, r_7, r_6, r_2, которое легло в основу варианта свертки (рисунок, *б*) регулярной матричной структуры (рисунок, *а*).

4. Проверка реализуемости неупорядоченного множества свертки

Для случая неупорядоченного множества свертки $L = \{l_1, l_2,\ldots, l_n\}$, где $l_k = (c_{k1}, c_{k2},\ldots, c_{km})$, рассматриваемая проблема формулируется следующим образом: проверить, реализуемо ли НМС L, и, если реализуемо, найти соответствующее реализуемое УМС и частичное упорядочение строк из $R(\boldsymbol{B})$. Этот случай отличается от рассмотренного выше только тем, что для проверки реализуемости необходимо просмотреть все возможные упорядочения наборов свертки $l_i \in L$, т. е. необходимо сгенерировать все возможные перестановки столбцов для каждого набора $l_i \in L$. Это требование следует из того факта, что неупорядоченный набор $l_k = (c_{k1}, c_{k2},\ldots, c_{km})$ порождает множество $P(l_k)$ из $m!$ возможных упорядоченных наборов $l_{ki}^o = \langle c_{i1}^k, c_{i2}^k,\ldots, c_{im}^k\rangle$ ($m = 1, 2,\ldots, m!$), соответствующих различным перестановкам столбцов из l_k. Таким образом, неупорядоченный набор l_k порождает следующие возможные отношения частичного порядка на $R(\boldsymbol{B})$, представляющие собой обобщение представления (2):

$$\bigcup_{l_{ks}^o \in P(l_k)} ((R(c_{s1}^k) > R(c_{s2}^k)) \cap (R(c_{s2}^k) > R(c_{s3}^k)) \cap \ \ldots \cap (R(c_{sm\text{-}1}^k) > R(c_{sm}^k))). \tag{9}$$

После кодирования строк из соответствующего множества R_L, формируемого аналогично тому, как это делалось для упорядоченных наборов свертки l_k^o, (4) принимает следующую форму, которая более сложна, чем соответствующее выражение (4) для l_k^o:

$$\bigcup_{l_{ks}^o \in P(l_k)} \Big(\bigcap_{q=1}^{m-1} \Big(\bigcap_{r_i \in R(c_{sq}^k)} \Big(\bigcap_{r_j \in R(c_{s(q+1)}^k)} f_{ij}\Big)\Big)\Big) = 1. \tag{10}$$

Исходя из (10), по аналогии с (5) НМС $L = \{l_1, l_2,\ldots, l_n\}$ реализуемо, если и только если следующее уравнение имеет хотя бы одно решение:

$$\bigcap_{k=1}^{n} \Big(\bigcup_{l_{ks}^o \in P(l_k)} \Big(\bigcap_{q=1}^{m-1} \Big(\bigcap_{r_i \in R(c_{sq}^k)} \Big(\bigcap_{r_j \in R(c_{s(q+1)}^k)} f_{ij}\Big)\Big)\Big)\Big) = 1. \tag{11}$$

Если уравнение (11) имеет корень (набор значений переменных x^i_l, выполняющих уравнение), соответствующее НМС L реализуемо. Корень определяет порядковые номера строк из R_L и, соответственно, упорядочение наборов свертки $l_k = (c_{k1}, c_{k2},\ldots, c_{km}) \in L$. Если уравнение (11) не имеет корня, НМС L нереализуемо.

Для рассмотренного ранее неупорядоченного множества свертки $L = \{(c_1, c_3, c_5), (c_2, c_6)\}$ аналогично упорядоченному множеству свертки L^o получаются множества $R_L = \{r_2, r_3, r_5, r_8\}$, $R'(c_1) = \{r_3, r_5\}$, $R'(c_3) = \{r_8\}$, $R'(c_5) = \{r_2\}$, $R'(c_2) = \{r_2, r_8\}$, $R'(c_6) = \{r_3, r_5\}$. НМС L порождает 12 возможных УМС L_k^o. После кодирования четырех строк из R_L булевыми переменными x_2 и x_1 уравнение (11) принимает вид

$$\begin{aligned}&((f_{3,8} \wedge f_{5,8} \wedge f_{8,2}) \vee (f_{3,2} \wedge f_{5,2} \wedge f_{2,8}) \vee (f_{8,3} \wedge f_{8,5} \wedge f_{3,2} \wedge f_{5,2}) \vee (f_{8,2} \wedge f_{2,3} \wedge f_{2,5}) \vee \\ &\vee (f_{2,3} \wedge f_{2,5} \wedge f_{3,8} \wedge f_{5,8}) \vee (f_{2,8} \wedge f_{8,3} \wedge f_{8,5})) \wedge ((f_{2,3} \wedge f_{2,5} \wedge f_{8,3} \wedge f_{8,5}) \vee (f_{3,2} \wedge f_{3,8} \wedge f_{5,2} \wedge f_{5,8})).\end{aligned} \tag{12}$$

5. Приведение уравнения для проверки реализуемости неупорядоченного множества свертки к виду КНФ

Для случая неупорядоченного множества свертки после представления в (11) функций f_{ij} в виде КНФ $C(f_{ij})$ получается следующее булево уравнение, левая часть которого в общем случае не есть КНФ:

$$\bigcap_{k=1}^{n} \Big(\bigcup_{l_{ks}^o \in P(l_k)} C_{ks}\Big) = 1. \tag{13}$$

Здесь C_{ks} представляет собой следующую КНФ:

$$C_{ks} = \bigcap_{q=1}^{m-1} \Big(\bigcap_{r_i \in R(c^k_{sq})} \Big(\bigcap_{r_j \in R(c^k_{s(q+1)})} C(f_{ij}) \Big) \Big). \qquad (14)$$

Булева функция $\bigcup_{l^o_{ks} \in P(l_k)} C_{ks}$ в левой части уравнения (13) представляется дизъюнкцией нескольких КНФ. Для того чтобы преобразовать ее к виду КНФ, не производя сложных вычислений, введем, как это предложено в [17], $k_m!$ булевых переменных z_{ks} с целью закодировать унитарным кодом $m!$ КНФ C_{ks}. Нетрудно показать, что условие выполнимости уравнения (13) не изменится, если формулу $\bigcup_{l^o_{ks} \in P(l_k)} C_{ks}$ заменить на

$$\Big(\bigcap_{l^o_{ks} \in P(l_k)} (C_{ks} \vee z_{ks}) \Big) \wedge \Big(\bigcup_{s=1}^{m!} \bar{z}_{ks} \Big). \qquad (15)$$

Здесь $(C_{ks} \vee z_{ks})$ представляется в форме КНФ, которая достаточно просто получается из КНФ C_{ks} применением закона (булевой алгебры) дистрибутивности конъюнкции относительно дизъюнкции (кодирующая переменная z_{ks} добавляется в каждый дизъюнкт $d_i \in C_{ks}$: $d_i \vee z_{ks}$). Для кодирования всех наборов свертки $l_k \in L$ понадобится ввести $\sum_{k=1}^{n} (k_m!)$ кодирующих переменных z_{ks}.

Таким образом, с учетом ранее введенных формул (7), (13)–(15) конструируется уравнение, левая часть которого представлена в форме КНФ:

$$\bigcap_{k=1}^{n} \Big(\bigcap_{l^o_{ks} \in P(l_k)} (C_{ks} \vee z_{ks}) \Big) \wedge \Big(\bigcup_{s=1}^{m!} \bar{z}_{ks} \Big) = 1.$$

Это уравнение решает ключевую задачу свертки регулярных матричных структур: проверяет реализуемость неупорядоченного множества свертки.

Для пояснения процедуры преобразования уравнений к виду КНФ продолжим приведение формулы (12) к виду КНФ для неупорядоченного множества свертки $L = \{(c_1, c_3, c_5), (c_2, c_6)\}$, анализируемого на реализуемость:

$$((x^3_2 \vee x^3_1 \vee z^1_1) \wedge (\bar{x}^8_2 \vee x^3_1 \vee z^1_1) \wedge (x^3_2 \vee \bar{x}^8_1 \vee z^1_1) \wedge (\bar{x}^8_2 \vee \bar{x}^8_1 \vee z^1_1) \wedge (x^3_2 \vee \bar{x}^8_2 \vee z^1_1)) \wedge$$
$$\wedge ((x^5_2 \vee x^5_1 \vee z^1_1) \wedge (\bar{x}^8_2 \vee x^5_1 \vee z^1_1) \wedge (x^5_2 \vee \bar{x}^8_1 \vee z^1_1) \wedge (\bar{x}^8_2 \vee \bar{x}^8_1 \vee z^1_1) \wedge (x^5_2 \vee \bar{x}^8_2 \vee z^1_1)) \wedge$$
$$\wedge ((x^8_2 \vee x^8_1 \vee z^1_1) \wedge (\bar{x}^2_2 \vee x^8_1 \vee z^1_1) \wedge (x^8_2 \vee \bar{x}^2_1 \vee z^1_1) \wedge (\bar{x}^2_2 \vee \bar{x}^2_1 \vee z^1_1) \wedge (x^8_2 \vee \bar{x}^2_2 \vee z^1_1)) \wedge \ldots$$
$$\wedge ((x^2_2 \vee x^2_1 \vee z^6_1) \wedge (\bar{x}^8_2 \vee x^2_1 \vee z^6_1) \wedge (x^2_2 \vee \bar{x}^8_1 \vee z^6_1) \wedge (\bar{x}^8_2 \vee \bar{x}^8_1 \vee z^6_1) \wedge (x^2_2 \vee \bar{x}^8_2 \vee z^6_1)) \wedge$$
$$\wedge ((x^8_2 \vee x^8_1 \vee z^6_1) \wedge (\bar{x}^3_2 \vee x^8_1 \vee z^6_1) \wedge (x^8_2 \vee \bar{x}^3_1 \vee z^6_1) \wedge (\bar{x}^3_2 \vee \bar{x}^3_1 \vee z^6_1) \wedge (x^8_2 \vee \bar{x}^3_2 \vee z^6_1)) \wedge$$
$$\wedge ((x^8_2 \vee x^8_1 \vee z^6_1) \wedge (\bar{x}^5_2 \vee x^8_1 \vee z^6_1) \wedge (x^8_2 \vee \bar{x}^5_1 \vee z^6_1) \wedge (\bar{x}^5_2 \vee \bar{x}^5_1 \vee z^6_1) \wedge (x^8_2 \vee \bar{x}^5_2 \vee z^6_1)) \wedge$$
$$\wedge (\bar{z}^1_1 \vee \bar{z}^2_1 \vee \bar{z}^3_1 \vee \bar{z}^4_1 \vee \bar{z}^5_1 \vee \bar{z}^6_1) \wedge ((x^2_2 \vee x^2_1 \vee z^1_2) \wedge (\bar{x}^3_2 \vee x^2_1 \vee z^1_2) \wedge$$
$$\wedge (x^2_2 \vee \bar{x}^3_1 \vee z^1_2) \wedge (\bar{x}^3_2 \vee \bar{x}^3_1 \vee z^1_2) \wedge (x^2_2 \vee \bar{x}^3_2 \vee z^1_2)) \wedge \ldots \wedge (\bar{z}^1_2 \vee \bar{z}^2_2) = 1.$$

Одним из наборов значений переменных, выполняющих эту КНФ, является набор, найденный ранее для КНФ (8). Он допускает следующее кодирование строк из $R_L = \{r_3, r_5, r_8, r_2\}$: $r_3 - 11$, $r_5 - 10$, $r_8 - 01$, $r_2 - 00$, что согласуется с вариантом свертки рассматриваемой матричной структуры, приведенным на рисунке, *б*.

Заключение

Разработан метод, решающий ключевую задачу многократной свертки – анализ множества свертки на реализуемость. Показано, как свести задачу анализа множества свертки на реализуемость к решению логического уравнения, а поиск корней уравнения – к известной задаче проверки выполнимости КНФ, для решения которой разработаны эффективные методы и программы. В отличие от известных методов проверки на реализуемость, формулируемых относительно упорядоченных множеств свертки, предлагаемый метод ориентирован на неупорядоченные множества, что позволяет снизить размерность задач построения множеств свертки. Описываемый метод попутно решает задачу нахождения такого упорядочения наборов множества свертки (если оно существует), которое обеспечивает реализуемость этого множества.

Список литературы

1. Ульман, Дж. Вычислительные аспекты СБИС / Дж. Ульман. – М. : Радио и связь, 1990. – 480 с.

2. Бибило, П.Н. Кремниевая компиляция заказных СБИС / П.Н. Бибило. – Минск : Ин-т техн. кибернетики АН Беларуси, 1996. – 268 с.

3. Biswas, N.N. Logic design theory / N.N. Biswas. – Prentice-Hall International, 1993. – 306 p.

4. Hachtel, G.D. An Algorithm for optimal PLA Folding / G.D. Hachtel, A.R. Newton, A.L. Sangiovanni-Vincentelli // IEEE Trans. Computer-Aided Design of Integrated Circuit Syst. – 1982. – Vol. CAD-1, no. 2. – P. 63–77.

5. DeMicheli, G.A. Multiple Constrained Folding of Programmable Logic Arrays: Theory and Applications / G.A. DeMicheli, A.L. Sangiovanni-Vincentelli // IEEE Trans. Computer-Aided Design. – 1983. – Vol. CAD-2, no. 3. – P. 151–167.

6. Черемисинова, Л.Д. Минимизация площади регулярных матричных структур заказных СБИС / Л.Д. Черемисинова // Информатика. – 2004. – № 1. – С. 121–131.

7. Минимизация площади заказных СБИС на этапе топологического проектирования цифровых схем / Л.Д. Черемисинова [и др.] // Управляющие системы и машины. – 2011. – № 4 (240). – С. 42–50.

8. Devadas, S. Optimal Layout via Boolean Satisfiability / S. Devadas // Proc. of Intern. Conf. on Computer-Aided Design (ICCAD ’89). – Santa Clara, CA, USA, 1989. – P. 294–297.

9. Optimum PLA Folding through Boolean Satisfiability / J.M. Quintana [et al.] // Asian South Pacific Design Automation Conference (ASP DAC’95). – Chiba, Japan, 1995. – P. 289–293.

10. Bryant, R.E. Graph-based algorithms for Boolean function manipulation / R.E. Bryant // IEEE Trans. Computers. – 1986. – Vol. C-35, no. 8. – P. 677–691.

11. Черемисинова, Л.Д. Свертка регулярных структур на основе решения логических уравнений / Л.Д. Черемисинова // Танаевские чтения : доклады Четвертой Междунар. науч. конф. (29–30 марта 2010, Минск). – Минск : ОИПИ НАН Беларуси, 2010. – С. 129–134.

12. Mahajan, Y. Zchaff2004: An Efficient SAT Solver / Y. Mahajan, Z. Fu , S. Malik // Theory and Applications of Satisfiability Testing (2004 SAT Solver Competition and QBF Solver Evaluation (Invited Papers)). – Berlin, Heidelberg : Springer, 2005. – P. 360–375.

13. Goldberg, E. BerkMin: A Fast and Robust SAT-Solver / E. Goldberg , Y. Novikov // Design, Automation, and Test in Europe. – Paris, 2002. – P. 142–149.

14. Eén, N., MiniSat / N. Eén, N. Sörensson [Electronic resource]. – Mode of access : http://www.cs.chalmers.se/Cs/Research/FormalMethods/ MiniSat. – Date of access : 09.02.2015.

15. Lecky, I.E. Graph theoretic flgorithms for the PLA folding problem / I.E. Lecky, O.I. Murphy, R.G. Abshe // IEEE Trans. Computer-Aided Design. – 1989. – Vol. 8, no. 9. – P. 1014–1021.

16. Cheremisinova, L.D. Some results in optimal PLA folding / L.D. Cheremisinova // Proc. of the Third Intern. Conf. on Computer-Aided Design of Discrete Devices (CAD DD’99). – Minsk : UIIP NAS B, 1999. – Vol. 1. – P. 59–64.

17. Cheremisinova, L. SAT-Based Approach to Verification of Logical Descriptions with Functional Indeterminacy / L. Cheremisinova, D. Novikov // 8th Intern. Workchop on Boolean problems. – Freiberg (Sachsen), 2008. – P. 59–66.

Поступила 09.02.15

Объединенный институт проблем информатики НАН Беларуси, Минск, Сурганова, 6
e-mail: cld@newman.bas-net.by

L.D. Cheremisinova

MULTIPLE FOLDING OF REGULAR STRUCTURES VIA SOLVING LOGIC EQUATIONS

The problem under consideration is to reduce the area of the layout of regular VLSI structures by means of their multiple folding. The method of solving the key problem of multiple folding, which is implementability checking of the folding set, is suggested. The method is based on the task reduction to solving a logic equation and checking Boolean satisfiability of a conjunctive normal form.

ЗАЩИТА ИНФОРМАЦИИ

УДК 004.056:004.62

А.И. Трубей

ГОМОМОРФНОЕ ШИФРОВАНИЕ: БЕЗОПАСНОСТЬ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ И ДРУГИЕ ПРИЛОЖЕНИЯ (ОБЗОР)

Представляются основные понятия гомоморфного шифрования и различные алгоритмы шифрования в соответствии с основополагающими свойствами гомоморфного шифрования. Приводятся практические примеры различных принципов и свойств гомоморфного шифрования, некоторые алгоритмы гомоморфного шифрования, использующие асимметричные ключевые системы, такие как RSA, Эль-Гамаля, Пэйе, а также различные схемы гомоморфного шифрования. Рассматриваются перспективы применения гомоморфного шифрования в области безопасных облачных вычислений, электронного голосования, поиска в зашифрованном тексте, фильтрации шифрованной почты, мобильного шифрования, защищенных систем с обратной связью.

Введение

Наряду с облегченной криптографией [1] значительный интерес для анализа и практического использования представляет гомоморфная криптография, или гомоморфное шифрование. Гомоморфное шифрование – это форма шифрования, позволяющая осуществлять определенные типы вычислений на зашифрованном тексте и получать зашифрованные результаты вычислений, которые при расшифровании соответствуют результатам операций, выполняемых на открытом тексте. Теоретические и практические аспекты гомоморфного шифрования тесно взаимосвязаны с проблемой обеспечения безопасности облачных вычислений. Идея облачных вычислений появилась еще в 1960 г., когда Джон Маккарти высказал предположение, что когда-нибудь компьютерные вычисления будут производиться с помощью «общенародных утилит». Считается, что идеология облачных вычислений обрела популярность с 2007 г. благодаря быстрому развитию каналов связи и стремительно растущим потребностям пользователей.

Облачные вычисления – одно из наиболее быстро развивающихся направлений в современных информационных технологиях (ИТ), приложения для использования в облачных сервисах являются приоритетными проектами ведущих мировых ИТ-компаний. Этот подход позволяет организовать динамическое предоставление услуг, что дает возможность пользователям производить оплату по факту использования ресурсов и регулировать их объем в зависимости от реальных потребностей без долгосрочных обязательств.

Рост мирового рынка облачных технологий оценивается в 20–25 % в год, в то время как рынок в целом увеличивается на 5–10 %. В России рынок ИТ в целом стагнирует, но динамика выручки в его облачном сегменте, по разным подсчетам, приближается к 40–60 %. Такие высокие темпы обоснованы тем, что рост с нуля всегда велик [2]. В Беларуси облачные серверы пока используются в основном для хостинга сайтов с высокой нагрузкой и обеспечения работы интернет-магазинов. Однако в последние два года наметился устойчивый тренд на размещение в облаке комплексных инфраструктурных проектов. Например, в 2013 г. выручка белорусского офиса группы компаний ActiveCloud, одного из крупнейших на постсоветском пространстве разработчиков ИТ-решений для размещения информационных ресурсов и систем в облаке, по услуге IaaS (инфраструктура как сервис) удвоилась по сравнению с 2012 г. [3].

Национальный институт стандартов и технологий NIST (National Institute of Standards and Technology, USA) определяет следующие характеристики облаков [4]:

– возможность высокоавтоматизированного самообслуживания системы со стороны провайдера;

– наличие системы Broad Network Access;

– сосредоточенность ресурсов на отдельных площадках для эффективного распределения;

– быструю масштабируемость (ресурсы могут неограниченно выделяться и высвобождаться с большой скоростью в зависимости от потребностей);

– наличие управляемого сервиса (система управления облаком автоматически контролирует и оптимизирует выделение ресурсов).

Таким образом, облачные вычисления – это способ обеспечения удобного сетевого доступа к разделяемому пулу реконфигурируемых вычислительных ресурсов (например, к сетям, серверам, устройствам памяти, приложениям и сервисам), которые могут быть быстро подобраны и предоставлены с минимальными усилиями для взаимодействия с поставщиком услуг.

Крупные вычислительные облака состоят из тысяч серверов, размещенных в центрах обработки данных. Они обеспечивают ресурсами десятки тысяч приложений, которые одновременно используют миллионы пользователей. Облачные технологии являются удобным инструментом для предприятий, которым слишком дорого содержать собственные ERP, CRM или другие серверы, требующие приобретения и настройки дополнительного оборудования. По модели развертывания облака разделяют на частные, общедоступные и гибридные [5].

Частные (корпоративные) облака – это внутренние облачные инфраструктура и службы организации. Все данные и приложения пользователя остаются внутри организации. Недостатком является то, что только крупные организации могут себе позволить создание частного вычислительного облака, поскольку его инфраструктура может быть достаточно дорогой и требовать высококвалифицированных администраторов.

Общедоступные (публичные) облака – это облачные сервисы, предоставляемые поставщиком. Пользователи данных облаков не имеют возможности управлять облаком или обслуживать его, вся ответственность возложена на владельца облака. Поставщик облачных услуг принимает на себя обязанности по установке, управлению, предоставлению и обслуживанию программного обеспечения, инфраструктуры приложений или физической инфраструктуры.

Гибридные облака представляют собой такое внедрение облачных вычислений, при котором часть системы размещается в публичном облаке, т. е. на базе центров данных облачного провайдера, часть – в частном облаке, т. е. на серверах самой организации. По сути, гибридное облако не является самостоятельным типом облачных вычислений, а лишь указывает на тесную интеграцию публичных и частных облачных систем.

Причины возрастающей популярности облачных технологий очевидны, возможности их применения очень разнообразны. Пользователь экономит как на обслуживании и персонале, так и на инфраструктуре. Нет необходимости приобретать лицензии на программное обеспечение, организацию и обслуживание собственных серверов, нанимать опытных администраторов и т. д. Все эти проблемы перекладываются на провайдера услуг. Кроме того, данный подход позволяет стандартизировать программное обеспечение, даже если на компьютерах предприятия установлены разные операционные системы (Windows, Linux, MacOS и т. д.).

Проблемы защиты информации в облачных технологиях стали активно анализироваться достаточно поздно, когда облака были уже фактически реализованной технологией. Практика применения облачных вычислений показала, что для защиты информации недостаточно уже имеющихся криптографических средств. Поясним это на следующем примере. Предположим, что в облаке S содержится множество пользователей (клиентов) $p_1, ..., p_i, ..., p_l$. У пользователя p_i имеются конфиденциальные данные x_i, хранящиеся в облаке. Такая облачная услуга называется Storage aaS (хранилище как сервис). Пользователь p_i может обратиться к облаку с запросом на вычисление значения некоторой функции F, зависящей от конфиденциальных данных. Запрос должен состоять из описания функции F, идентификатора пользователя и его открытого ключа pk_i. Облако должно проверить полномочия пользователя p_i на вычисление $F(x_i)$. Такая проверка может быть реализована с помощью стандартной процедуры электронной цифровой подписи (ЭЦП). Если пользователь подтвердил свои права на вычисление функции F, то облако должно вычислить значение $E(pk_i, F(x_i))$ и отправить его пользователю. В качестве E можно взять функции шифрования некоторой криптосистемы с открытым ключом.

Пользователь, который размещает в хранилище свои конфиденциальные данные и дает запрос на вычисление функции F, не доверяет облаку и должен принимать соответствующие меры и предъявлять требования по обеспечению их безопасности. Очевидно, что было бы гораздо безопаснее передавать данные в таком виде, чтобы во время операций, которые производятся над ними, никоим образом не распространялась информация об этих данных. Поэтому, во-первых, данные необходимо шифровать, причем они должны поступать на сервер уже в шифрованном виде. Это означает, что шифрование должно осуществляться еще пользователем. Во-вторых, необходимо обрабатывать эти данные без расшифровки, так как для передачи и хранения секретного ключа необходимо соблюдение определенных процедур, особенно сложных, если информация обрабатывается в недоверенной среде.

Оказалось, что защита информации в облачных вычислениях намного сложнее тех задач защиты информации, которые решаются известными криптографическими средствами. Криптосистемы с открытым ключом для решения данной проблемы не всегда подходят. В 1978 г. авторы известного алгоритма с открытым ключом RSA Майкл Дертузос, Рональд Риверст и Леонард Адлеман впервые обосновали, что методом, позволяющим успешно проводить операции над зашифрованными данными, не искажая и не расшифровывая их, является так называемое гомоморфное шифрование [6]. В своей работе они описали концепцию гомоморфного шифрования, а также задались вопросами, возможно ли такое шифрование в принципе и для каких алгебраических систем такой гомоморфизм существует.

В статье на основании изучения большого фактического материала приводятся основные понятия и определения в области гомоморфного шифрования, теоретические и практические проблемы по разработке данных систем шифрования, структура, свойства и операции соответствующих криптоалгоритмов. Осуществлен также краткий обзор существующих и перспективных систем гомоморфного шифрования и их практического применения.

1. Теоретические основы гомоморфного шифрования

Гомоморфное шифрование является формой шифрования, позволяющей осуществить определенную алгебраическую операцию над открытым текстом посредством выполнения алгебраической операции над зашифрованным текстом. Пусть $E(k, m)$ – функция шифрования, где k – ключ шифрования, а m – открытый текст. Функция E называется гомоморфной относительно операции * над открытыми текстами, если существует эффективный алгоритм M (требующий полиномиального числа ресурсов и работающий за полиномиальное время), который, получив на вход любую пару шифрованных текстов вида $E(k, m_1)$, $E(k, m_2)$, выдает шифрованный текст $c = M(E(k,m_1), E(k,m_2))$, такой, что при расшифровании c будет получен открытый текст $m_1 * m_2$ [7].

Как правило, рассматривается следующий частный случай гомоморфного шифрования. Для данной функции шифрования E и операции $*_1$ над открытыми текстами существует операция $*_2$ над шифрованными текстами, такая, что из шифрованного текста $E(k,m_1) *_2 E(k,m_2)$ при расшифровании извлекается открытый текст $m_1 *_1 m_2$. При этом требуется, чтобы по заданным c, $E(k, m_1)$, $E(k, m_2)$, но при неизвестном ключе было бы невозможно эффективно проверить, что шифрованный текст c получен из $E(k, m_1)$ и $E(k, m_2)$.

Любую стандартную систему шифрования можно описать в виде трех операций: генерации ключей (KeyGen), шифрования (Encypt) и расшифрования (Decrypt).

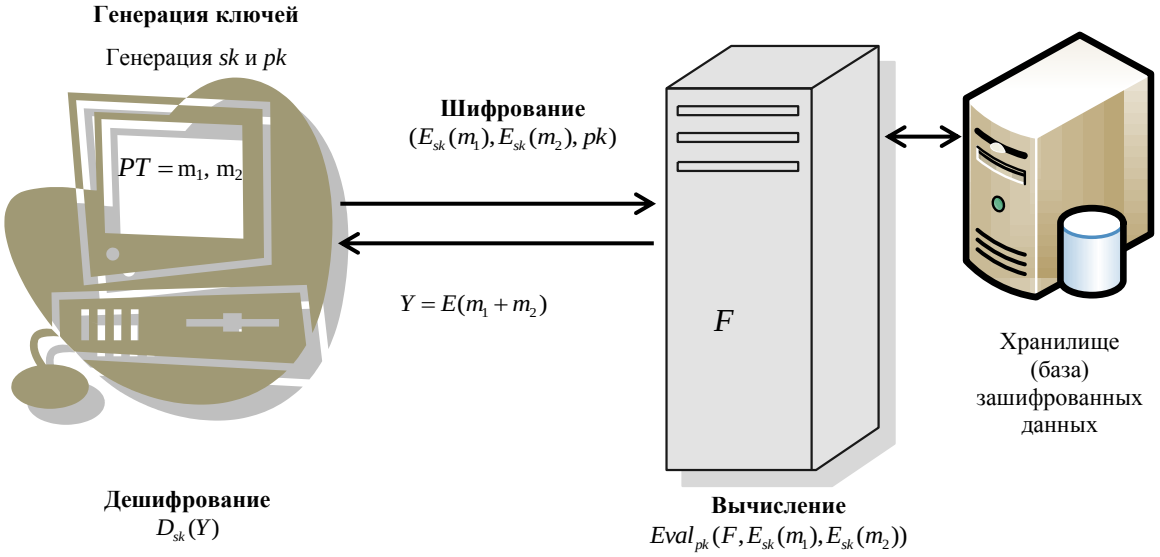
Гомоморфная система шифрования, кроме трех перечисленных выше операций, включает в себя операцию вычислений (Eval). Таким образом, гомоморфное шифрование является последовательностью из четырех операций: генерации ключей, шифрования, вычисления, расшифрования (рисунок) [8]:

– генерация ключей: клиент генерирует открытый ключ pk (public key) и секретный ключ sk (secret key) для шифрования открытого текста;

– шифрование: используя секретный ключ sk, клиент шифрует открытый текст PT (plain text), создает $E_{sk}(PT)$ и вместе с открытым ключом pk отправляет шифрованный текст CT (cipher text) на сервер;

– вычисление: сервер получает функцию F для проведения вычислений над шифрованным текстом CT и выполняет их в соответствии с требованиями данной функции, используя pk;

– расшифрование: для получения искомого результата значение $Eval(F(PT))$, полученное в ходе вычислений, расшифровывается клиентом с использованием своего секретного ключа sk.



Результат $D_{sk}(Eval_{pk}(F, E_{sk}(m_1), E_{sk}(m_2)))$

Операции гомоморфного шифрования

Система шифрования является гомоморфной относительно операции умножения, если

$$D(E(m_1) \otimes E(m_2)) = m_1 \cdot m_2.$$

Система шифрования является гомоморфной относительно операции сложения, если

$$D(E(m_1) \oplus E(m_2)) = m_1 + m_2.$$

Система шифрования является гомоморфной относительно операций умножения и сложения, т. е. полностью гомоморфной, если

$$D(E(m_1) \otimes E(m_2)) = m_1 \cdot m_2;$$
$$D(E(m_1) \oplus E(m_2)) = m_1 + m_2,$$

где $\otimes$, $\oplus$ – операции умножения и сложения над шифрованными текстами, соответствующие операциям умножения и сложения над открытыми текстами; D – функция расшифрования; E – функция шифрования.

Если криптосистема с такими свойствами сможет зашифровать два бита, то, поскольку операции сложения и умножения формируют над битами полный по Тьюрингу базис, становится возможным вычислить любую булеву функцию, а следовательно, и любую другую вычислимую функцию.

Свыше 30 лет оставалась нерешенной задача полностью гомоморфного шифрования − создания системы, гомоморфной относительно операций сложения и умножения одновременно. Только в 2009 г. аспирант Стэндфордского университета и стажер IBM Крейг Джентри теоретически обосновал принципиальную возможность создания такой системы шифрования. В схеме Джентри [9] выполняются свойства гомоморфизма как относительно умножения, так и сложения, т. е. она является алгебраической гомоморфной системой. Предложенная сис-

тема может использоваться для обеспечения конфиденциальности данных при любых видах их обработки в недоверенной среде, например при облачных или распределенных вычислениях. Однако модель Джентри оказалась слишком непрактичной. С увеличением количества операций, производимых над зашифрованным текстом, сложность и размер шифрованного текста увеличиваются с невероятной скоростью. Несмотря на то что за последние годы было проведено множество улучшений данной схемы, она все еще остается скорее теоретической моделью, которая пока не применима на практике.

2. Краткий обзор систем гомоморфного шифрования

В настоящем разделе на примерах конкретных алгоритмов и схем описываются мультипликативные, аддитивные [10, 11] и смешанные [9, 12] свойства гомоморфного шифрования. Приведенные алгоритмы и схемы являются общедоступными, поэтому дадим детальное описание только некоторых из них, в отношении других ограничимся кратким перечислением их основных свойств и областей применения.

Криптосистема RSA. Метод шифрования RSA (аббревиатура от фамилий создателей – Rivest, Shamir, Adleman) предложен в 1977 г. как реализация идеи основоположников криптографии с открытым ключом Диффи и Хеллмана.

Предположим, что открытый текст представлен числом m, таким, что $0 < m < N$. Пользователь B желает, чтобы ему передали секретное сообщение. Для этого он делает общедоступными два числа N (составной модуль) и e (открытый ключ), которые удовлетворяют следующим условиям: $N = pq$, где p, q – большие простые числа, которые B держит в секрете; $p, q \geq 2^{256}$; e выбирается взаимно простым с $\varphi(N) = (p-1)(q-1)$.

Пользователь A, отправивший сообщение m, шифрует его следующим образом:

$$E(m) = m^e \pmod N.$$

Это и есть шифрованный текст, который получает B.

Для расшифрования B находит число d, такое, что $1 \leq d \leq N-1$ и $ed = 1(\mathrm{mod}\,\varphi(N))$. Данное сравнение разрешимо единственным образом, так как $(e, \varphi(N)) = 1$. Для решения уравнения $ed = 1(\mathrm{mod}\,\varphi(N))$ пользователь B должен вычислить $\varphi(N)$, что для него не составляет труда, так как $\varphi(N) = \varphi(pq) = \varphi(p)\varphi(q) = (p-1)(q-1)$. Любой другой пользователь, который знает только N, вынужден находить p и q, т. е. разлагать число N на простые сомножители, а эта задача при больших p и q имеет значительную вычислительную сложность.

Далее, имея в распоряжении $y = E(m) = m^e \pmod N$, пользователь B вычисляет величину $D(y) = y^d \pmod N$, которая является открытым текстом m. Действительно, применяя теорему Эйлера, получаем

$$D(y) = y^d = m^{ed} = m^{\varphi(N)k+1} = (m^{\varphi(N)})^k m = m(\mathrm{mod}\,N).$$

Криптосистема RSA гомоморфна относительно операции умножения открытых текстов. Для любых двух открытых текстов m_1, m_2

$$E(m_1)E(m_2) = m_1^e m_2^e \pmod N = E(m_1 m_2).$$

Криптосистема Эль-Гамаля. Пусть G – циклическая группа порядка p и g – порождающий элемент группы. В качестве секретного ключа выбирается случайный элемент d группы Z_{p-1}. Соответствующий открытый ключ e вычисляется по формуле $e = g^d$.

Функция шифрования для сообщения m выглядит следующим образом:

$$E(e, m) = (e^r m, g^r),$$

где r – случайный элемент группы Z_{p-1}.

Расшифрование криптограммы (c_1, c_2) выполняется следующим образом. Вычисляется

$$c_2^d = g^{rd},$$

откуда

$$m = c_1 / c_2.$$

Криптосистема Эль-Гамаля гомоморфна относительно операции умножения открытых текстов. Если $E(e, m_1) = (e^{r_1} m_1, g^{r_1})$ и $E(e, m_2) = (e^{r_2} m_2, g^{r_2})$, то

$$E(e, m_1 m_2) = (e^{r_1} e^{r_2} m_1 m_2, g^{r_1} g^{r_2}) = E(e, m_1) E(e, m_2).$$

Криптосистема Пэйе. Криптосистема базируется на алгоритме вероятностного асимметричного преобразования и применяется в криптографических протоколах с отрытым ключом.

Пусть p и q – два простых числа, $n = pq$, $\lambda = \text{НОК}(p-1, q-1)$. Выберем случайное число g из $Z_{n^2}^*$ и вычислим $\mu = (L(g^\lambda \bmod n^2))^{-1} (\bmod n)$, где $L(u) = (u-1)/u$.

Открытым ключом является пара (n, g), а закрытым ключом – пара (λ, μ).

Для шифрования открытого текста $m \in Z_n$ выбирается случайное число $r \in Z_{n^2}^*$ и вычисляется $E(m) = c = g^m r^n (\bmod n^2)$.

Расшифрование выполняется по формуле $m = L(c^\lambda (\bmod n^2))\mu (\bmod n)$.

Свойство гомоморфизма выглядит следующим образом:

$$E(m_1) E(m_2) = (\mathrm{g}^{m_1} r_1^n)(\mathrm{g}^{m_2} r_2^n) = \mathrm{g}^{m_1 + m_2} (r_1 r_2)^n = E(m_1 + m_2) (\bmod n).$$

Схема Джентри полностью гомоморфного шифрования. Рассмотрим предложенную схему Джентри на примере вычислений в Z_2.

1. Генерация ключей. Выбирается произвольное нечетное целое число $p = 2k+1$. Данное число p является секретным ключом.

2. Шифрование. Пусть требуется зашифровать бит $m \in (0,1)$. Для этого сгенерируем число $z = 2r + m$, где r – произвольное целое число. Это означает, что $z = m (\bmod 2)$.

Шифрование заключается в том, что всякому числу m ставится в соответствие число $c = pq + z$, где q – произвольное целое число. Следовательно, $E(m) = c = 2r + m + (2k+1)q = 2(r + kq) + m + q$.

Вычислениям подвергается именно это число c.

3. Расшифрование. Пусть даны числа c, p, q, где p, q известны. Проведем расшифрование с помощью секретного ключа p:

$$c (\bmod p) = (z + pq) (\bmod p) = z (\bmod p) + pq (\bmod p) = z (\bmod p) =$$

$$= (2r + m) (\bmod p) = 2(r (\bmod p)) + m (\bmod p).$$

Далее вычисляем

$$(c (\bmod p)) (\bmod 2) = (2(r (\bmod p)) (\bmod 2) = m (\bmod 2) = m.$$

Шифрование является гомоморфным относительно операций сложения и умножения. Рассмотрим пару битов $m_1, m_2 \in (0,1)$. Сопоставим им $z_1 = 2r_1 + m_1$, $z_2 = 2r_2 + m_2$. Выберем сек-

ретный ключ $p = 2k + 1$. Тогда $E(m_1) = c_1 = z_1 + pq_1$, $E(m_2) = c_2 = z_2 + pq_2$ – шифрованные тексты для m_1 и m_2 соответственно.

Операция сложения над зашифрованными числами будет иметь вид

$$E(m_1) + E(m_2) = c_1 + c_2 = z_1 + z_2 + p(q_1 + q_2) = 2(r_1 + r_2) + m_1 + m_2 + p(q_1 + q_2).$$

Операция умножения над зашифрованными числами будет иметь вид

$$E(m_1)E(m_2) = c_1c_2 = z_1z_2 + p(z_1q_2 + z_2q_1) + p^2q_1q_2 =$$
$$= (2r_1 + m_1)(2r_2 + m_2) + \mathrm{p}(\mathrm{z}_1 q_2 + z_2 q_1) + p^2q_1q_2 =$$
$$= 4r_1r_2 + 2(r_1m_2 + r_2m_1) + m_1m_2 + p(z_1q_2 + z_2q_1) + p^2q_1q_2.$$

При расшифровании

$$D(E(m_1) + E(m_2)) = ((c_1 + c_2)(\mathrm{mod}\, p))(\mathrm{mod}\, 2) = m_1 + m_2;$$

$$D(E(m_1)E(m_2)) = ((c_1c_2)(\mathrm{mod}\, p))(\mathrm{mod}\, 2) = m_1m_2.$$

Существенным недостатком данной схемы является то, что выполнение вычислений приводит к накоплению ошибки и, после того как она превышает *p*, расшифровать сообщение становится невозможным. Одним из вариантов решения данной проблемы является перешифровка данных после некоторого количества операций, однако такой вариант снижает производительность вычислений и требует постоянного доступа к секретному ключу. Стойкость схемы Джентри на основе идеальных решеток (решеток со свойствами идеала на некотором кольце чисел) сводится к *NP*-полной задаче нахождения кратчайшего вектора. Появилось немало работ, направленных на развитие предложенных в ней идей и устранение недостатков. В частности, была предложена схема BGV (аббревиатура от фамилий создателей – Brakerski, Gentry, Vaikuntanathan). Авторы представили альтернативный вариант полностью гомоморфного шифрования на основании LWE (Learning With Errors) [13], который позволил уменьшить сложность построения криптосистемы, однако унаследовал основные недостатки схемы Джентри:

– наличие возрастающей ошибки в шифрованном тексте;

– рост размера шифрованного текста.

В зависимости от обстоятельств свойство гомоморфизма может рассматриваться как в качестве достоинства, так и в качестве недостатка криптосистемы. Это относится, например, к криптосистеме RSA, в которой функция расшифрования используется в схеме ЭЦП. Подпись сообщения *m* вычисляется по формуле $s = m^d (\mathrm{mod}\, N)$, где d – секретный компонент ключа. Очевидно, что и это преобразование гомоморфно относительно операции умножения. Следовательно, можно предложить следующий способ подделки подписей. Если известны подписи s_1, s_2 сообщений m_1, m_2, то ЭЦП сообщения m_1m_2 будет являться соответственно s_1s_2. Однако на практике такая уязвимость не представляет угрозы стойкости схемы ЭЦП, так как подписываются не сами сообщения, а значения хэш-функций сообщений. Тем не менее гомоморфизм функции генерации подписей накладывает на хэш-функцию дополнительное требование, которое, вообще говоря, не следует из стандартных определений криптографической хэш-функции. Основные параметры гомоморфных систем шифрования приведены в таблице [8, 14–17].

Большинство алгоритмов гомоморфного шифрования, стойкость которых базируется на сложности дискретного логарифмирования в конечном поле, достаточно легко переносятся на случай эллиптических кривых. Криптосистемы на основе эллиптических кривых превосходят другие системы с открытым ключом по двум важным параметрам: степени защищенности в расчете на каждый бит ключа и быстродействию при программной реализации. Это объясняется тем, что для вычисления обратных функций на эллиптических кривых известны только алгоритмы с экспоненциальным ростом трудоемкости, тогда как для обычных систем предложены также субэкспоненциальные методы. При обеспечении одной и той же стойкости криптографических протоколов вычисления в группе точек эллиптической кривой выполняются примерно

на 20 % быстрее, чем для групп конечного поля [18]. В работе [16] приводятся сравнительные характеристики быстродействия и стойкости алгоритмов на эллиптических кривых.

Сравнительные характеристики некоторых гомоморфных систем шифрования

Система (схема)	Год	Гомоморфизм			Применение
		по сложению	по умножению	полный	
RSA	1977	–	✓	–	Обеспечение безопасности Интернета, банковских транзакций, транзакций по кредитным картам
Гольдвассер – Микали (Goldwasser – Micali)	1982	–	✓	–	Первая вероятностная криптосистема с открытым ключом. Премия Тьюринга за 2012 г.
Эль-Гамаль (ElGamal)	1985	–	✓	–	Гибридные облачные системы. Модификацией данной схемы является схема Шнорра, которая положена в основу СТБ 34.101.45–2013
Бенало (Benaloh)	1988	✓	–	–	Системы электронного голосования
Накаче – Штерн (Naccache Stern)	1998	✓	–	–	Является усовершенствованием схемы Бенало
Окамото – Очияма **(**Okamoto – Uchiyama)	1998	✓	–	–	При проектировании операционной системы EPOC (Electronic Piece Of Cheese*)*
Пэйе (Paillier)	1999	✓	–	–	Системы электронного голосования, пороговые схемы
Дамгард – Джарик (Damgard – Jurik)	2001	✓	–	–	Обобщение схемы Пэйе для больших модулей с целью расширения области применения
Бракерски – Джентри – Вайкунтанатан (Brakerski – Gentry – Vaikuntanathan, BGV)	2012	–	–	✓	Обеспечение безопасности целочисленных многочленов
Неитерактивная экспоненциальная гомоморфная система шифрования (Non-interactive Exponential Homomorphic Encryption Scheme, NEHE)	2012	–	–	✓	Активные сети, электронная коммерция на основе мобильных агентов, грид-вычисления
Алгебраическая гомоморфная система на базе модифицированной схемы Эль-Гамаля (Algebra Homomorphic Encryption Scheme Based On Updated ElGamal, AHEE)	2012	–	–	✓	Протокол конфиденциальных вычислений, электронное голосование и мобильные шифры
Усовершенствованная криптосистема Горти (Gorti's Enhanced Homomorphic Cryptosystem, EHC)	2013	–	–	✓	Обеспечение безопасности передачи сообщений в беспроводных децентрализованных самоорганизующихся сетях (MANET)

На основании вышеизложенного можно утверждать, что для применения гомоморфного шифрования на практике разработанные криптосистемы должны удовлетворять, по крайней мере, следующим требованиям:

– набор поддерживаемых математических функций должен покрывать повседневные нужды программистов;

– точность и скорость вычислений не должны деградировать в течение вычислений;

– стойкость алгоритма должна исключить атаку полным перебором.

3. Области применения гомоморфного шифрования

Применение гомоморфного шифрования может представлять значительный интерес в следующих областях:

Облачные вычисления. Как уже было отмечено выше, гомоморфное шифрование открывает новые возможности по сохранению целостности, доступности и конфиденциальности данных при их обработке в облачных системах. В облачных вычислениях, где производительность является главным приоритетом, следует применять разные алгоритмы, каждый из которых лучше всего справляется с поставленной задачей. Например, для операций умножения зашифрованных данных целесообразно использовать алгоритмы RSA или Эль-Гамаля, а для сложения – Пэйе. Возможно применение комбинированных систем данных алгоритмов. Для операций сравнения и сортировки необходимо использовать другие схемы.

Для практического применения полностью гомоморфной системы шифрования следует ограничивать количество операций, которые можно производить над данными без риска выйти за границу критических пределов ошибки вычислений. При этом предпочтительным является использование гомоморфного шифрования в гибридных облачных системах, так как вычисления, которые не могут быть вынесены в публичное облако в силу законодательных или иных ограничений, могут производиться во внутренней сети.

Электронное голосование. Электронное голосование – еще одна перспективная сфера применения гомоморфного шифрования. Система сможет зашифровать голоса избирателей и провести расчеты над зашифрованными данными, сохраняя анонимность избирателей. Например, в схеме электронного голосования Бенало процесс голосования включает следующие этапы [19]:

– каждый голосующий участник схемы разделяет свой голос (секрет) на составляющие (частичные секреты) по соответствующей ему схеме разделения секрета со свойством гомоморфности по сложению и посылает частичные секреты выборным представителям;

– представители складывают полученные голоса; по свойству гомоморфности (по сложению) суммы голосов являются частичными секретами соответствующего итога выборов, а значит, суммы голосов могут быть вычислены без нарушения конфиденциальности схемы;

– главное доверительное лицо вычисляет конечный итог голосования, используя набор частичных сумм голосов, переданный ему выборными представителями.

Предположим, поставлена задача выбора лучших сотрудников Объединенного института проблем информатики НАН Беларуси. Имеется набор из n кандидатов, из которых формируется список, включаемый в бюллетень. Дирекция, которая обладает криптосистемой, гомоморфной относительно операции сложения, распространяет среди участников тайного голосования бюллетень как вектор $(p_1, ..., p_i, ..., p_n)$, где p_i – фамилия i-го кандидата. Кроме того, голосующим передается открытый ключ системы гомоморфного шифрования pk. Каждый из избирателей составляет вектор предпочтений $(v_1, ..., v_i, ..., v_n)$, где $v_i \in 0,1$. После этого с помощью открытого ключа pk он поэлементно шифрует вектор и отправляет представителю дирекции. Для подведения итогов голосования тот производит вычисления над соответствующими элементами полученных векторов предпочтений и производит расшифрование с помощью секретного ключа sk. Так как криптосистема гомоморфна относительно операции сложения, индексы наибольших элементов результирующего вектора и будут индексами победивших кандидатов. В качестве системы гомоморфного шифрования при тайном голосовании, включая тайное голосование с весами, может использоваться криптосистема Пэйе.

Защищенный поиск информации. Гомоморфное шифрование может предоставить пользователям возможность извлечения информации из поисковых систем с сохранением конфиденциальности: сервисы смогут получать и обрабатывать запросы, а также выдавать результаты обработки, не анализируя и не фиксируя их реальное содержание. Например, метод извлечения записей из базы данных по их индексам можно представить следующим образом.

Пусть $v_1, v_2, ..., v_j, ..., v_n; v_j \in 0,1$ – индекс записи, которую нужно извлечь; $c_1, c_2, ..., c_i, ..., c_{2^n}$ – все проиндексированные записи из базы данных.

Тогда, для того чтобы выбрать требуемую запись, необходимо вычислить следующую функцию F:

$$F(v_1, v_2, ..., v_j, ..., v_n; c_1, c_2, ..., c_i, ..., c_{2^n}) =$$

$$= c_1 \cdot ((v_1 \oplus 1) \otimes (v_2 \oplus 1) \otimes ... \otimes (v_n \oplus 1)) +$$

$$+ c_2 \cdot ((v_1 \oplus 1) \otimes (v_2 \oplus 1) \otimes ... \otimes (v_{n-1} \oplus 1) \otimes v_n) +$$

$$+ c_3 \cdot ((v_1 \oplus 1) \otimes (v_2 \oplus 1) \otimes ... \otimes v_{n-1} \otimes (v_n \oplus 1)) + ... +$$

$$+ c_{2^n} \cdot (v_1 \otimes v_2 \otimes ... \otimes v_n).$$

Если все c_i зашифрованы с помощью гомоморфной криптосистемы, F можно вычислить гомоморфно над шифрованными текстами. Для этого клиенту достаточно побитно зашифровать индекс $v_1, v_2, ..., v_j, ..., v_n$ нужной ему записи и отправить на сервер. Результат гомоморфного вычисления функции F над шифрованными текстами будет искомым шифрованным значением записи c_i, запрашиваемой клиентом. Очевидно, что криптосистема должна обладать как мультипликативными, так и аддитивными гомоморфными свойствами.

Защита беспроводных децентрализованных сетей связи. Беспроводные децентрализованные самоорганизующиеся сети (MANET) – это сети, состоящие из мобильных устройств. Каждое такое устройство может независимо передвигаться в любых направлениях и, как следствие, часто разрывать и устанавливать соединения с соседями. Одной из основных проблем при построении MANET является обеспечение безопасности передаваемых данных. Для решения этой проблемы может применяться гомоморфное шифрование [8], которое встраивается в протоколы маршрутизации для повышения безопасности. В этом случае операции над шифрованными текстами могут безопасно выполняться промежуточными узлами. В частности, для нахождения оптимального пути между двумя узлами необходимо осуществлять линейные операции над зашифрованными данными без их расшифрования. Наличие гомоморфного шифрования не позволяет злоумышленнику найти связь между сообщениями, входящими в узел и выходящими из узла. Поэтому невозможно отследить путь передачи сообщения с помощью анализа трафика [20].

Аутсорсинговые услуги для смарт-карт. В настоящее время наблюдается тенденция к разработке универсальных карт с собственной операционной системой, которая может выполнять разнообразные функции и взаимодействовать с несколькими поставщиками услуг. Высказываются предположения, что некоторые приложения могут работать вне карты на гомоморфно зашифрованных данных. Особо ресурсоемкие приложения, например приложения сервис-провайдеров, а также биометрические проверки (распознавание голоса, отпечатков пальцев или почерка), которым, как правило, требуется значительный объем хранения и большое количество сравнительно простых операций, могут использовать внешние устройства хранения и внешние процессоры, более мощные, чем на карте.

Системы с обратной связью. Гомоморфное шифрование может использоваться, например, в так называемых безопасных гомоморфных системах с обратной связью (secure feedback system) [14], когда необходимо сохранить анонимность пользователя и скрыть промежуточные результаты вычислений. Системы помогают осуществлять анонимный сбор отзывов (комментариев) студентов либо преподавателей об их работе. Полученные таким образом отзывы шифруются и сохраняются для последующих вычислений. Системы с обратной связью могут быть использованы для повышения осведомленности о состоянии дел и улучшения показателей работы.

Установлено, что достоверная обратная связь любой системы или процесса может быть обеспечена только в случаях сохранения анонимности пользователя, неизменности данных, собранных в процессе обратной связи, обеспечения безопасности внутренних операций для анализа данных.

Обфускация для защиты программных продуктов. Впервые о применении обфускации в криптографии было упомянуто в работе Диффи и Хеллмана [21]. В ней предложено использовать для построения асимметричной криптосистемы сложность задачи, заключающейся в анализе программ на низкоуровневом языке программирования (ассемблере, байт-коде). Основной целью обфускации является затруднение понимания функционирования программы [22]. Поскольку все традиционные компьютерные архитектуры используют двоичные строки, применяя полностью гомоморфное шифрование над битами, можно вычислить любую функцию. Следовательно, можно гомоморфно зашифровать целиком всю программу так, что она сохранит свою функциональность [7, 23].

Кроме того, гомоморфные свойства различных криптосистем в перспективе могут быть использованы для архивации и хранения медицинских записей без угрозы их утечки, фильтрации зашифрованной электронной почты, создания стойких к коллизиям хэш-функций.

Заключение

На основании вышеизложенного можно утверждать, что в ближайшей перспективе средства и методы гомоморфного шифрования будут оказывать существенное влияние на рынок облачных услуг и в той или иной степени на облик современных информационных технологий. Однако пока не созданы эффективные алгоритмы полностью гомоморфного шифрования, обеспечивающие уровень производительности, пригодный для практического применения, а тем более для применения в системах реального времени. Все предлагаемые схемы не реализуемы на практике и не готовы к внедрению в реальные системы, так как приводят к накоплению ошибок и быстрому увеличению шифрованных текстов. При этом частично гомоморфные системы (относительно операций сложения или умножения) успешно применяются в облачных вычислениях, электронном голосовании, защищенном поиске информации, системах с обратной связью и т. д.

Следует учитывать, что некоторые гомоморфные криптосистемы могут поддаваться преднамеренным внешним воздействиям (например, принципиально уязвимы к атаке с адаптивно подобранным шифрованным текстом) и поэтому не всегда подходят для безопасной передачи данных. Оценка криптостойкости гомоморфных систем требует отдельного исследования.

В отличие от облегченной криптографии для гомоморфного шифрования пока не разработаны соответствующие международные стандарты, однако активно продолжаются работы по созданию приемлемых решений, позволяющих безопасно обрабатывать конфиденциальные данные в облаках и других приложениях.

Список литературы

1. Поляков, А.С. Анализ возможностей алгоритмов международного стандарта «Облегченная криптография» – ISO/IEC 29192-2:2012 / А.С. Поляков, В.Е. Самсонов // Информатика. – 2014. – № 3. – С. 107–112.

2. Облачные технологии: новые задачи [Электронный ресурс]. – Режим доступа : http://events.cnews.ru/events/oblachnye_tehnologii__novye_zadachi.shtml. – Дата доступа : 03.02.2015.

3. Как ActiveCloud собирается заработать больше, чем обещает рынок облачных технологий [Электронный ресурс]. – Режим доступа : http://probusiness.by/tech/205.html. – Дата доступа : 03.02.2015.

4. Батура, Т.В. Облачные технологии: основные понятия, задачи и тенденции развития / Т.В. Батура, Ф.А. Мурзин, Д.Ф. Семич // Программные продукты и системы. – 2014. – № 3. – С. 64–72.

5. Афанасьев, С.В. Облачные сервисы, онтологическое моделирование таксономии / С.В. Афанасьев // Труды СПИИРАН. – 2012. – № 23. – С. 392–399.

6. Rivest, R.L. On data banks and privacy homomorphisms / R.L. Rivest, L. Adleman, M.L. Dertouzos // Foundations of secure computation. – 1978. – Vol. 32, no. 4. – P. 169–178.

7. Варновский, Н.П. Гомоморфное шифрование / Н.П. Варновский, А.В Шокуров // Труды Ин-та системного программирования РАН. – 2006. – Т. 12. – С. 27–36.

8. Survey of various homomorphic encryption algorithms and schemes / P.V. Parmar [et al.] // Intern. J. of Computer Applications. – 2014. – Vol. 91, no. 8. – P. 26–32.

9. Gentry, C. A Fully homomorphic encryption using ideal lattices / C. Gentry // Symposium on the Theory of Computing (STOC). – Bethesda, USA, 2009. – P. 169–178.

10. Математические и компьютерные основы криптологии : учеб. пособие / Ю.С. Харин [и др.]. – Минск : Новое знание, 2003. – 381 с.

11. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes / P. Paillier // Advances in cryptology – EUROCRYPT’99. – Berlin, Heidelberg : Springer, 1999. – P. 223–238.

12. Жиров, А.О. Безопасные облачные вычисления с помощью гомоморфной криптографии / А.О. Жиров, О.В. Жирова, С.Ф. Кренделев // Безопасность информационных технологий. – 2013. – Т.1. – С. 6–12.

13. Gentry, C. Homomorphic Encryption from Learning With Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based / C. Gentry, A. Sahai, B. Waters // Advances in cryptology – CRYPTO-2013, 33rd Annual Cryptology Conf. – Santa Barbara, CA, USA, 2013. – Part 1. – P. 73–93.

14. Ahmad, I. Survey: homomorphic encryption schemes / I. Ahmad, D. Adiga // Proc. of 9th IRF Intern. Conf. – Pune, India, 2014. – P. 89–94.

15. Jain, N. Implementation and analysis of homomorphic encryption schemes / N. Jain, S.K. Pal, D.K. Upadhyay // Intern. J. on Cryptography and Information Security (IJCIS). – 2012. – Vol. 2, no. 2. – P. 27–44.

16. Patel, S.J. Comparative Evaluation of Elliptic Curve Cryptography Based Homomorphic Encryption Schemes for a Novel Secure Multiparty Computation / S.J. Patel, A. Chouhan, D.C. Jinwala // J. of Information Security. – 2014. – № 5. – P. 12–18.

17. Batch fully homomorphic encryption over the integers / J.H. Cheon [et al.] // Advances in Cryptology – EUROCRYPT'2013 (LNCS). – 2013. – Vol. 7881. – P. 315–335

18. Степанян, А.Б. Актуальные вопросы обеспечения надежности и безопасности программного обеспечения систем дистанционного банковского обслуживания / А.Б. Степанян, А.И. Трубей, В.В. Анищенко // Электроника инфо. – 2014. – № 12. – С. 35–40.

19. Шенец, Н.Н. Модулярное разделение секрета и системы электронного голосования / Н.Н. Шенец // Вестник БГУ. Сер. 1. – 2011. – № 1. – С. 101–104.

20. Габидулин, Э.М. Защита информации в телекоммуникационных сетях / Э.М. Габидулин, Н.И. Пилипчук, О.В. Трушина // Труды МФТИ. – 2013. – Т. 5, № 3. – С. 97–111.

21. Diffie, W. New directions in cryptography / W. Diffie, M. Hellman // IEEE Trans. Inf. Theory. – 1976. – Vol. 22, no. 11. – P. 644–654.

22. Сергейчик, В.В. Особенности обфускации VHDL-описаний и методы оценки ее сложности / В.В. Сергейчик, А.А. Иванюк // Информатика. – 2014. – № 1. – С. 116–125.

23. On the relationship between functional encryption, obfuscation, and fully homomorphic encryption / J. Alwen [et al.] // Cryptography and Coding – 14th IMA Intern. Conf., IMACC-2013. – Oxford, UK, 2013. – P. 65–84.

Поступила 21.01.2015

Объединенный институт проблем информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: trubeia@newman.bas-net.by

A.I. Trubei

HOMOMORPHIC ENCRYPTION: CLOUD COMPUTING SECURITY AND OTHER APPLICATIONS (A SURVEY)

Homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on cipher text and to obtain an encrypted result which matches the result of operations performed on the plain text. The article presents a basic concept of the homomorphic encryption and various encryption algorithms in accordance with the fundamental properties of the homomorphic encryption. The examples of various principles and properties of homomorphic encryption, some homomorphic algorithms using asymmetric key systems such as RSA, ElGamal, Paillier algorithms as well as various homomorphic encryption schemes are given. Prospects of homomorphic encryption application in the field of secure cloud computing, electronic voting, cipher text searching, encrypted mail filtering, mobile cipher and secure feedback systems are considered.

УДК 004.056.53

В.В. Сергейчик, А.А. Иванюк

ОБЗОР МЕТОДОВ РЕАЛИЗАЦИИ АППАРАТНЫХ ВОДЯНЫХ ЗНАКОВ В ЦИФРОВЫХ УСТРОЙСТВАХ ПРОГРАММИРУЕМОЙ ЛОГИКИ

Рассматривается применение технологии водяных знаков для защиты цифровых устройств и их проектных описаний. Приводятся основные определения, модели, категории атак, характеристики, классификация водяных знаков для данной области. Описываются примеры использования аппаратных водяных знаков.

Введение

Рост интереса к программируемым логическим устройствам (ПЛУ), и в частности к программируемым логическим интегральным схемам (ПЛИС), вызван сокращением разницы в производительности между заказными интегральными схемами (ИС) и ПЛУ, отсутствием в случае ПЛУ дорогой и продолжительной фазы изготовления проекта в кремнии (расходы на маску составляют 2 млн долл. для технологии 32 нм [1]), возможностями быстрого прототипирования. В 2010 г. были начаты около 110 000 проектов на базе ПЛИС и 2500 на базе заказных ИС [2].

Разрыв между конструированием и проектированием [3] осложняется высокими требованиями к производительности, функциональности, надежности цифровых устройств и скорости выхода на рынок. Одним из решений данной проблемы является методология повторного использования, основанная на применении готовых, оптимизированных и протестированных модулей, называемых компонентами интеллектуальной собственности (IP-компонентами) [4]. Механизмы приобретения и распространения IP-компонентов недостаточно отработаны: производители IP-компонентов оказываются уязвимыми перед пиратством, не имея возможности обнаружить и предотвратить неправомерное использование своих компонентов.

Угрозы включают: копирование, обратное проектирование, аппаратные трояны, извлечение секретной информации по сторонним каналам. В индустрии ущерб от пиратства и подделок оценивается в 169 млрд долл. в год [5]. Аппаратные трояны – это злонамеренные изменения схемотехники устройства с целью снижения уровня защиты, передачи секретной информации из работающего устройства, нарушения работоспособности. Извлечение секретной информации осуществляется путем измерения физических параметров схемы (тока, задержки, электромагнитного излучения), которые затем могут быть скоррелированы с внутренними вычислениями, секретными ключами. Обратное проектирование дает представление о внутреннем устройстве и функционировании схемы, облегчая внедрение трудно обнаружимых аппаратных троянов, проведение атак по сторонним каналам, клонирование и несанкционированное использование.

Существует несколько подходов к защите. При шифровании IP-компонент интегрируется в проект без раскрытия его внутренней структуры. Подход привязан к конкретным системам автоматизированного проектирования (САПР), что снижает гибкость процесса проектирования и открывает уязвимость перед атаками на САПР и ключи [2]. Примеры промышленного использования: Synopsis (DesignWare), Xilinx (Core Generator), Microsemi (Direct Cores). Второй вариант подхода – шифрование бит-образа – требует аппаратной поддержки в ПЛИС и уязвим к атакам на ключи по сторонним каналам. Аппаратные водяные знаки (АВЗ) – это цифровые водяные знаки (ЦВЗ), используемые для защиты цифровых устройств (далее в тексте термин ЦВЗ будет употребляться в определениях, в целом справедливых для водяных знаков). АВЗ скрывают информацию об авторстве внутри описания или схемы. Отпечатки пальцев – это АВЗ, уникальные для каждого пользователя, что позволяет отслеживать источник нелегального копирования. Суть идентификации состоит в доказательстве присутствия IP-компонента в про-

ектном описании путем сравнения характеристик проекта и характеристик компонента. Активное измерение использует уникальные неклонируемые различия, возникающие при производстве ИС. Компонент начинает работу в нефункциональном состоянии, определяемом модулем физически неклонируемой функции. Для перевода в стартовое состояние требуется подача входной последовательности, различной для каждой ИС. В отличие от АВЗ, помечающих проектное описание, а не ИС, при пассивном измерении помечаются конкретные ИС с целью последующего наблюдения или отслеживания. Обфускация скрывает смысл описания, структуры или функционирования схемы, усложняя понимание проектного описания и схемы, а также обратное проектирование. Примерами использования обфускации могут служить IP-компоненты Microsemi Direct Cores [6].

1. Модель АВЗ

Существуют различные определения ЦВЗ. Например, ЦВЗ – это технология, обеспечивающая безопасность, идентификацию и защиту авторского права для цифровых данных [7]. ЦВЗ – это данные, внедряемые в информационный объект с целью контроля его использования [8]. ЦВЗ представляет собой метод встраивания информации, применяемый с определенной целью, например для идентификации и защиты авторского права [3]. Для лучшего понимания текста вводятся следующие определения: контейнер – информационный объект, в котором скрыт ЦВЗ [8]; сообщение – встраиваемые данные; сторона – лицо или группа лиц, осуществляющих общую деятельность: владелец IP-компонента, пользователь, поставщик IP-компонентов.

В коммуникационных моделях технология ЦВЗ рассматривается как передача проверяющей стороне сообщения от встраивающей стороны. В зависимости от степени использования моделью свойств контейнера он рассматривается как шум, шум со вспомогательной информацией, другое информационное сообщение, передаваемое вместе с ЦВЗ путем мультиплексирования [9].

Процедура использования АВЗ состоит из двух этапов: встраивания и извлечения.

В ходе этапа встраивания из исходного проектного описания V с помощью метода Wm постановки АВЗ и сообщения K, идентифицирующего автора, получают описание V^*, содержащее некоторое свойство (инвариантное преобразованиям, производимым при переходе к более низким уровням абстракции и при оптимизациях), которое позволяет доказать авторство: $V^* = Wm(V, K)$. Под оптимизациями здесь и далее понимаются оптимизационные преобразования, осуществляемые САПР на различных этапах проектирования.

Подготовка встраиваемого сообщения часто включает хеширование, шифрование, генерацию псевдослучайной последовательности, добавление помехоустойчивых кодов. Результатом процедуры синтеза DD описаний V и V^* являются схемные представления Sch и Sch^*, различающиеся, но соответствующие одной и той же функциональной спецификации $func$:

$$DD(V) = Sch;\ DD(V^*) = Sch^*;\ \ func(Sch^*) = func(Sch).$$

В ходе извлечения АВЗ процедурой обнаружения D определяется присутствие или отсутствие сообщения K в описании и (или) синтезированной из него схеме:

$$D(V^*, K) = true;\ D(Sch^*, K) = true.$$

Доказательство авторства при использовании АВЗ опирается на аппарат теории вероятностей. Важнейшей характеристикой этапа встраивания является вероятность совпадения P_u, указывающая на возможность того, что незапланированный АВЗ будет обнаружен в проектном описании [10]. Длина последовательности сообщения АВЗ выбирается в соответствии с требуемым значением P_u. Фаза извлечения характеризуется двумя величинами: P_m – вероятностью не обнаружить существующий АВЗ и P_f – вероятностью ложного обнаружения.

Важнейшим требованием к АВЗ является сохранение функциональной корректности проектного описания. Выделяют следующие основные характеристики АВЗ [11]. *Стой-*

кость – устойчивость к искажениям (в том числе вызванным синтезом, оптимизациями) и атакам. Чтобы увеличить стойкость, АВЗ желательно сделать функциональной частью IP-компонента. *Емкость* – максимальное количество данных, которые можно внедрить в контейнер. *Затраты на встраивание (извлечение)* – вычислительная сложность встраивания (извлечения), необходимость наличия дополнительного оборудования, технологий, экспертов. *Вносимые издержки* – степень ухудшения качества помеченного проектного описания (и синтезированной из него схемы) по сравнению с проектным описанием (схемой) до внедрения. *Скрытность* – степень сходства свойств (в том числе статистических) АВЗ со свойствами окружающей области внедрения [12]. *Дополнительные характеристики*: доказательство происхождения (проектное описание было создано конкретной стороной, а не просто содержит АВЗ, указывающий на эту сторону), доказательство целостности (отсутствия вмешательства в проектное описание после создания), возможность опровержения (см. ниже атаку подделыванием авторства), невозможность отрицания передачи IP-компонента другим сторонам. Важной характеристикой АВЗ является *прозрачность* для средств проектирования [13]: этап внедрения АВЗ должен легко интегрироваться в процесс проектирования.

В модели оценки [14] производительность АВЗ вычисляется как функция пяти переменных: Em_Cost, Trace_Cost – стоимость встраивания, стоимость извлечения (время работы алгоритма на компьютере), Coin_Pro – вероятность совпадения, Security – устойчивость к конкретным видам атак, Overhead – увеличение аппаратных издержек.

Атаки на ЦВЗ делятся на четыре категории [9]:

1. К категории *неавторизованного удаления* относятся маскирующие атаки и атаки удалением. Маскирующие атаки затрудняют обнаружение и извлечение АВЗ, не изменяя его. Атаки удалением направлены на разрушение, ухудшение качества АВЗ до уровня, когда он больше не может использоваться для доказательства авторства.

2. *Неавторизованное встраивание* включает атаки встраиванием и атаки копированием. Атаки встраиванием направлены на добавление другого (не авторского) АВЗ в проектное описание. В атаках копированием (подделыванием авторства) злоумышленник внедряет АВЗ владельца, извлеченный из одного проектного описания, в другое проектное описание, утверждая, что это проектное описание создано владельцем, например, с целью дискредитации последнего.

3. Иногда возможность извлечения сообщения АВЗ или различения его частей (например, идентификаторов конкретных пользователей) должна быть доступна только ограниченному кругу лиц. Нарушение таких ограничений – это *атака неавторизованного обнаружения*.

4. *Атаки системного уровня* направлены на ошибки использования АВЗ, ошибки реализации, ключи или на саму концепцию как таковую.

2. Классификация методов АВЗ

Приведем классификацию методов АВЗ (рис. 1). АВЗ делят на статические и динамические [15]. Динамические строятся во время функционирования IP-компонента и представляют собой некоторое свойство его состояния. Эти методы особенно перспективны в силу того, что свойство проявляется только при подаче определенных входных данных. Статические представляют собой свойство проектного описания некоторого уровня абстракции.

АВЗ классифицируются по типу IP-компонента, используемого как контейнер. Выделяют программные, аппаратные и фирменные IP-компоненты [4]. Программные IP – это высокоуровневые описания, аппаратные IP – синтезированные описания после трассировки и размещения, фирменные IP – описания в форме списка соединений элементов для целевой ПЛИС до трассировки и размещения.

По устойчивости к изменению АВЗ делятся на устойчивые и хрупкие. В [16] предлагается метод встраивания символов хрупкого АВЗ в выходные последовательности КА. Хрупкие АВЗ не используются для доказательства авторства, а служат индикатором изменения или повреждения проектного описания. Изменение IP-компонента приводит к разрушению хрупких АВЗ.

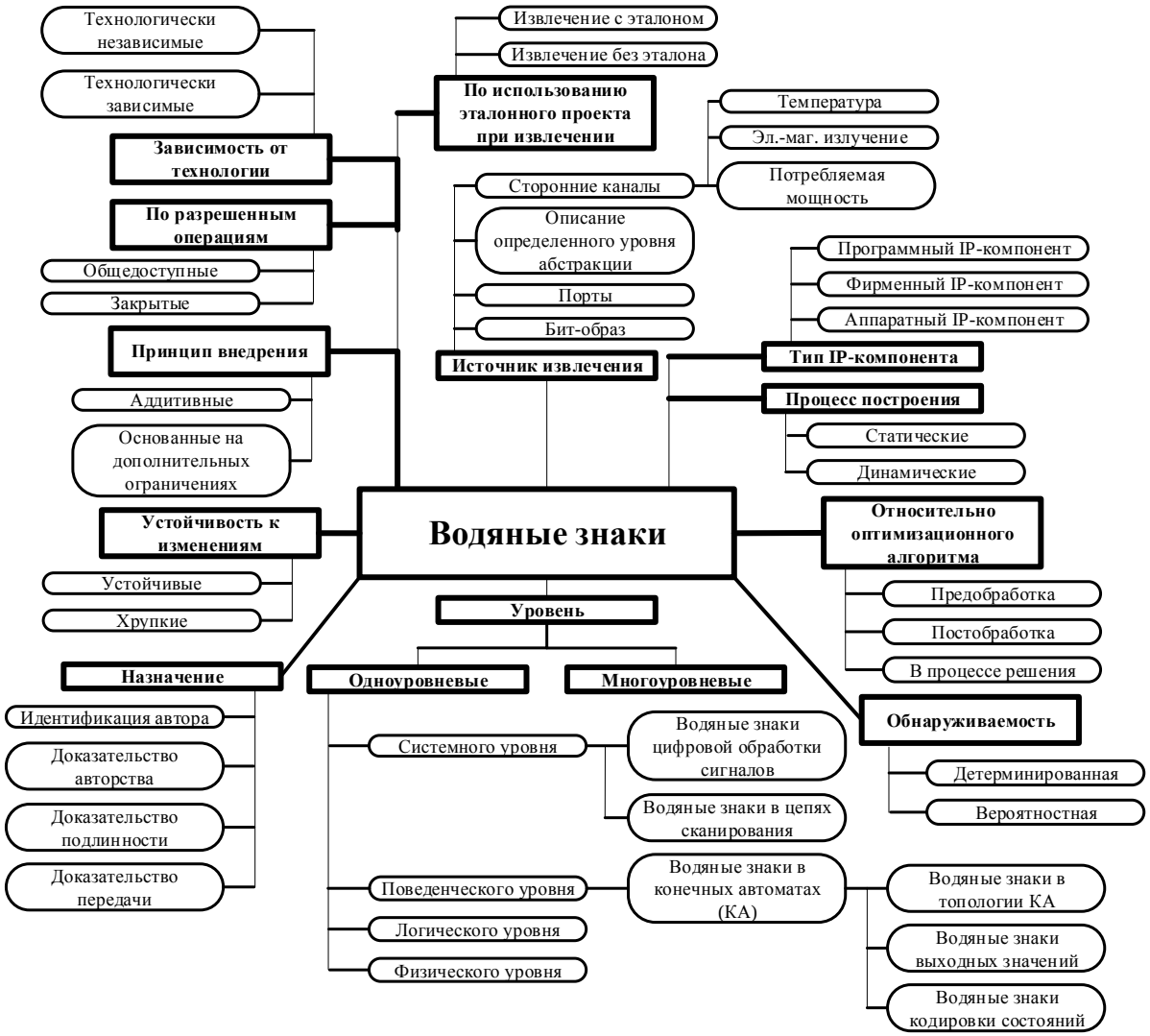


Рис. 1. Классификация методов АВЗ

По времени встраивания АВЗ относительно алгоритма решения оптимизационной задачи методы делят на три группы: предобработку, постобработку и методы в процессе решения [14, 17]. Методы предобработки внедряют АВЗ в описание до передачи алгоритму решения оптимизационной задачи. АВЗ, сохранившийся после решения оптимизационной задачи, труднее удалить. В методах постобработки сначала решается оптимизационная задача, затем в результат решения добавляется АВЗ. Легкость реализации таких методов компенсируется тем, что они в большей степени подвержены атакам. В методах третьей группы встраивание АВЗ является частью алгоритма оптимизации и осуществляется во время решения оптимизационной задачи. Сложность представляет интеграция в процесс разработки.

В зависимости от необходимости использования при извлечении эталона (проектного описания до встраивания водяного знака) АВЗ делятся на слепые и неслепые [9]. В слепых методах АВЗ может быть извлечен без эталона. В неслепых методах проводится сравнение, ищется корреляция свойств проверяемого и эталонного описаний.

По критерию обнаружения методы делят на детерминированные и вероятностные. В детерминированных методах при проверке АВЗ требуется побитовое равенство, также возможно использование пороговых значений. Вероятностные методы опираются на некоторое статистическое свойство.

АВЗ делят на общедоступные и закрытые в зависимости от доступности операции обнаружения. В закрытых АВЗ лишь привилегированная сторона (например, владелец IP-компонента) может извлечь и декодировать исходное сообщение.

Методы, зависящие от конструктивных особенностей конкретной платформы ПЛИС, называют технологически зависимыми, остальные – технологически независимыми.

По принципу внедрения выделяют две основные группы методов: аддитивные (основанные на добавлении новых элементов или свойств в проектное описание) и основанные на введении дополнительных ограничений.

По уровню абстракции проектного описания АВЗ для ПЛИС делятся на следующие группы: системного, поведенческого, логического, физического уровней и многоуровневые. Процесс проектирования и место АВЗ в нем показаны на рис. 2. Описания более низких уровней представляют собой детализацию более высоких уровней, поэтому АВЗ более высокого уровня сохраняются в описаниях низкого уровня, т. е. содержатся (в разной форме) в нескольких контейнерах. В случае обратного проектирования при достижении более высокого уровня абстракции АВЗ низших уровней не сохраняются, так как изначально на этом уровне не существует нужный контейнер.

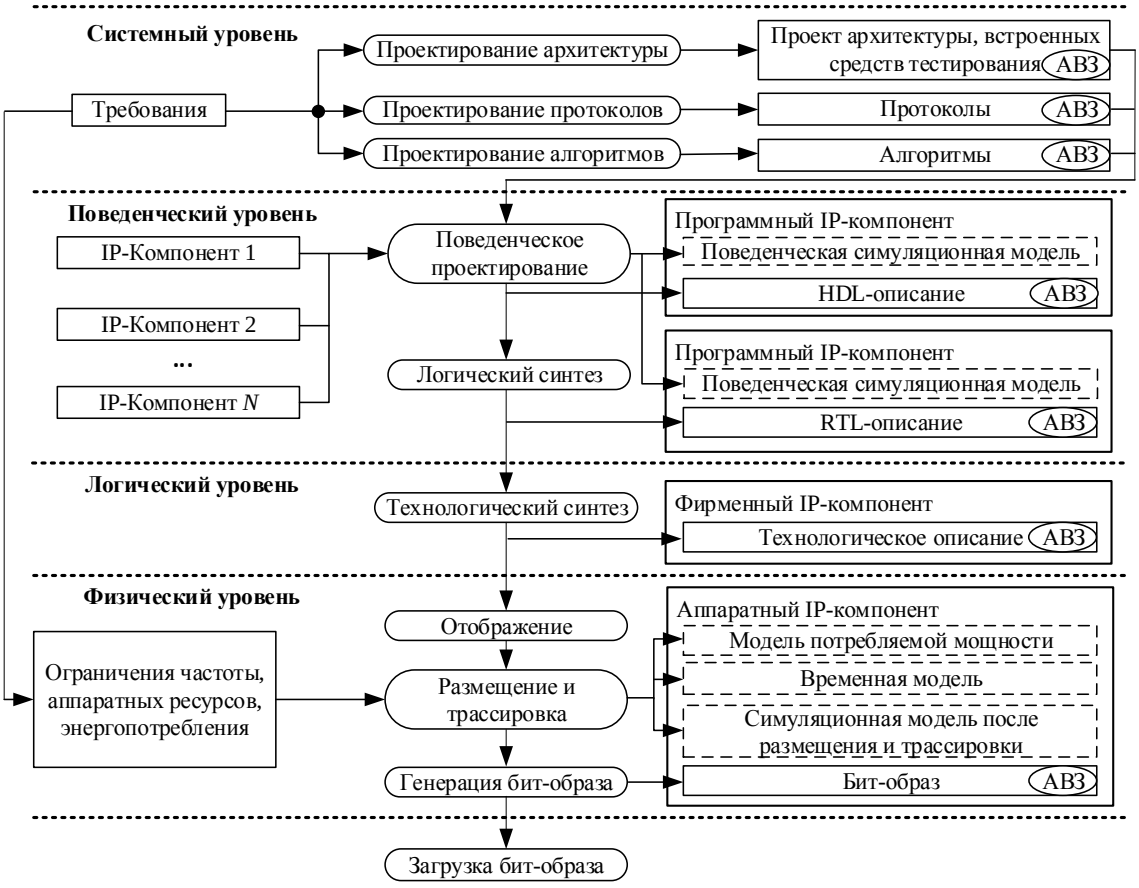


Рис. 2. Процесс проектирования цифровых устройств

2.1. Классификация методов по уровню абстракции

АВЗ системного уровня. Методы системного уровня используют особенности реализуемого алгоритма, предметной области, архитектуры разрабатываемого цифрового устройства. В [18] описывается метод встраивания АВЗ в спецификацию цифрового фильтра на алгоритмическом уровне путем разбиения полосы пропускания на отрезки с увеличением или уменьшением амплитуды в пределах отрезка на величину Δ в соответствии с битом АВЗ. Если бит АВЗ равен единице, то единица отнимается от верхней границы амплитуды отрезка; в противном случае единица прибавляется к нижней границе [19].

Следующая группа методов предоставляет доступ к АВЗ через встроенные средства тестирования и самотестирования. Метод [20] основан на NP-сложной проблеме упорядочивания сканирующих ячеек с целью минимизации времени или потребляемой мощности в ходе сканирующего теста. Перестановка π – это отображение один к одному набора сканирующих ячеек $R = \{r_i\}$ на набор позиций $P = \{p_j\}$ так, что j-й бит тестового вектора загружается в i-ю сканирующую ячейку, как только полный тестовый вектор установлен в сканирующую цепь, где $i, j = 1, 2, \ldots, N$. В процессе оптимизации вводится ограничение на значения определенных сканирующих ячеек для конкретных тестовых векторов, при подаче которых на выходе сканирующей цепи получается ответ, содержащий биты АВЗ на заданных позициях.

В работе [21] используются несколько сканирующих цепей. При внедрении вычисляется хэш-значение сообщения. Для расщепления хэш-значения на группы генерируется последовательность случайных чисел $R_n = \{r_1, r_2, \ldots, r_n\}$, представляющих число битов в группах. Биты групп представляются в виде целого числа – номера сканирующей ячейки в сканирующей цепи, значение которого будет инвертировано в режиме извлечения водяного знака (высокий уровень сигнала wmEn). При извлечении один и тот же вектор подается на вход сканирующей цепи в обоих режимах (wmEn = 1 и wmEn = 0). Определяются номера позиций, в которых ответы различаются. С помощью R_n из номеров восстанавливаются биты соответствующей группы водяного знака.

АВЗ поведенческого уровня. Среди АВЗ поведенческого уровня лучше изучены методы, связанные со свойствами конечных автоматов (КА). Методы рассматриваются на примере КА (рис. 3, *а*).

Идея метода [22] заключается в таком преобразовании графа передачи состояний (ГПС), что последовательность состояний r_i, посещаемая при подаче бит АВЗ a_i, имеет специальные топологические свойства: каждое r_i может быть достигнуто только из r_{i-1} и только при подаче на вход a_i. Для этого ГПС при построении дублируется, а между копиями создается единственный путь (рис. 3, *в*). Проверка присутствия АВЗ сводится к подтверждению с помощью техник тестирования наличия указанных топологических свойств для последовательности АВЗ.

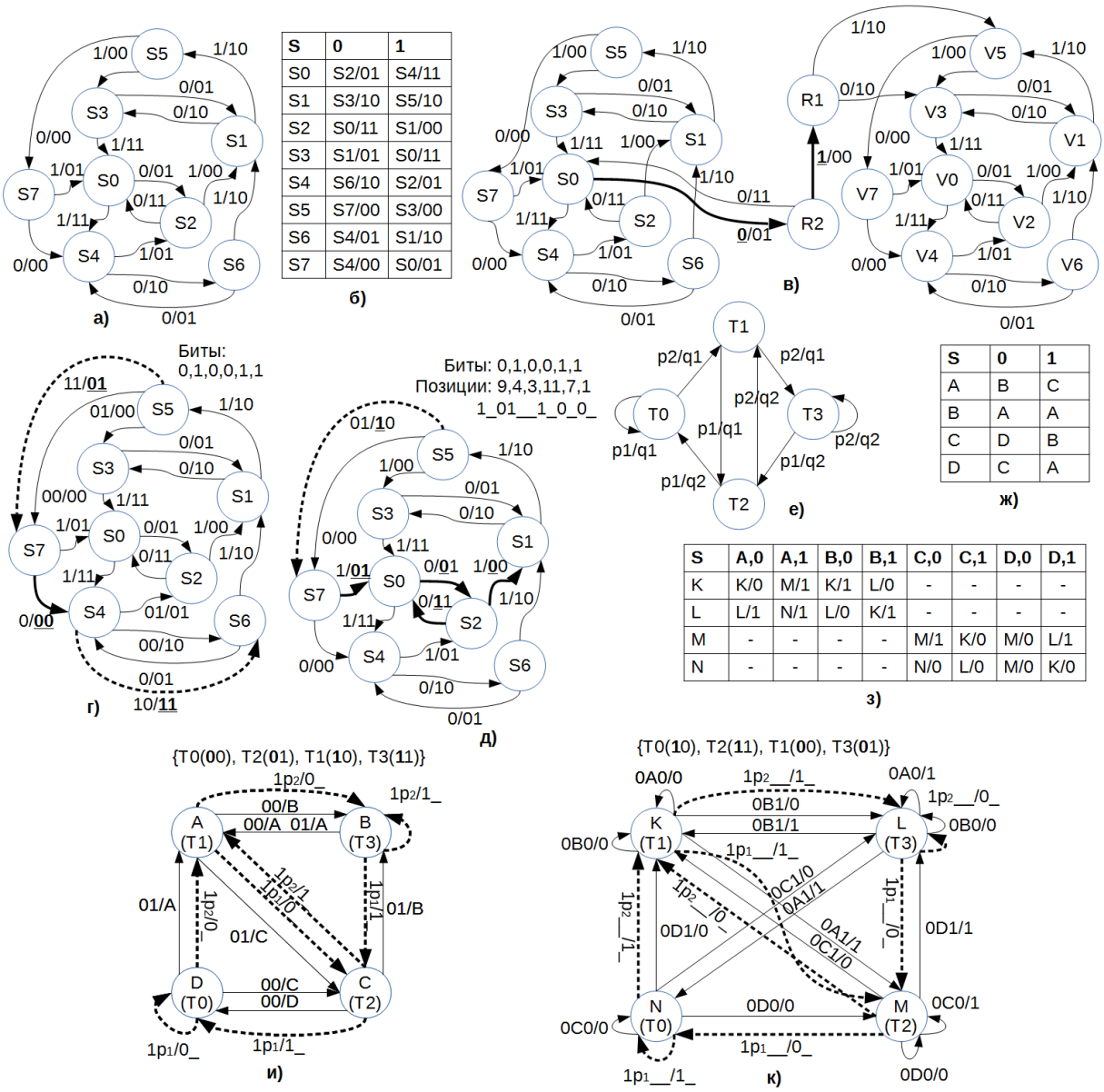
Метод, описанный в [10], опирается на поиск неиспользуемой последовательности входных и выходных символов, соответствующей АВЗ. Если пар входных и выходных символов недостаточно, то вводятся новые входные переменные. Авторы используют геномный поиск для вероятностного подтверждения присутствия АВЗ в случае его повреждения.

Подобно [10], в [15] в качестве АВЗ используется последовательность входных и выходных символов. Встраивание начинается с произвольного состояния, при этом не осуществляется ресурсоемкий поиск путей в ГПС, а выбираются переходы, выходные символы которых совпадают с битами АВЗ. Если совпадений нет, то добавляются новые переходы, если больше переходов добавить нельзя, то вводится новый входной символ. Символ принимает значение 0 для существующих состояний и 1 для добавляемых (рис. 3, *г*).

В [23] предусматривается несколько модификаций метода [15] (рис. 3, *д*). Генерируется псевдослучайная последовательность индексов позиций. Начиная с произвольного состояния, выбираются переходы, в которых биты выходных значений, заданных индексами, совпадают с битами АВЗ под теми же индексами. Это позволяет лучше защитить АВЗ, повысить вероятность нахождения существующего перехода с требуемыми выходными сигналами, снизить издержки и увеличить эффективность внедрения в КА с большим количеством выходных переменных. Добавленным входным переменным присваиваются произвольные значения, а в [15] требуются фиксированные, что может облегчить проведение атак.

Метод [24] внедрения АВЗ в кодировку состояний КА в методологии тестопригодного проектирования опирается на широкое использование существующей проектной логики для реализации тестовой логики, помеченной АВЗ. Последовательностная схема из N триггеров может быть представлена в виде множества связанных объектных автоматов (ОА). Каждому ОА из n триггеров можно поставить в соответствие обобщенный тестовый автомат (ТА) с двумя входными, двумя выходными переменными и 2^n состояниями (рис. 3, *е*). ТА устанавливается в любое состояние синхронизирующей последовательностью длины n, состояние на выходе идентифицируется с помощью разделяющей последовательности длины n. АВЗ представляет собой ограничения на кодировку состояний ТА, выбранных псевдослучайно среди всех ТА проектного описания. Например, значение первого триггера для первого состояния может быть равно значению

бита водяного знака. Устанавливая ТА в определенные состояния и проверяя выходные значения, можно определить кодировку, выбранную для конкретного автомата, а следовательно, и биты водяного знака. Декомпозиция автомата *а* на два ОА показана на рис. 3, *ж*, *з*, результирующие КА со встроенными ТА и АВЗ 01 – на рис. 3, *и*, *к*.



S	0	1
S0	S2/01	S4/11
S1	S3/10	S5/10
S2	S0/11	S1/00
S3	S1/01	S0/11
S4	S6/10	S2/01
S5	S7/00	S3/00
S6	S4/01	S1/10
S7	S4/00	S0/01

S	0	1
A	B	C
B	A	A
C	D	B
D	C	A

S	A,0	A,1	B,0	B,1	C,0	C,1	D,0	D,1
K	K/0	M/1	K/1	L/0	-	-	-	-
L	L/1	N/1	L/0	K/1	-	-	-	-
M	-	-	-	-	M/1	K/0	M/0	L/1
N	-	-	-	-	N/0	L/0	M/0	K/0

Рис. 3. Примеры методов АВЗ: *а), б)* исходный КА и его таблица переходов и выходов; *в)* метод [22]; *г)* метод [15]; *д)* метод [23]; *е)* ТА; *ж), з)* результат декомпозиции *а*; *и), к)* декомпозированные компоненты со встроенными ТА и АВЗ [24]

В [25] используется концепция иерархических КА, в соответствии с которой выделяют главный и подчиненный КА. Входной и выходной алфавиты подчиненного КА выступают подмножествами соответствующих алфавитов главного. Подчиненный КА отвечает на реакцию главного. Реакция иерархического КА определяется следующим образом: если данное состояние простое, то иерархический КА ведет себя как обычный КА, в противном случае реагируют и подчиненный и главный КА, т. е. переключаются два состояния и выполняются два действия. Для постановки АВЗ некоторые состояния расщепляются, создается подчиненный КА, добавляются ложные переходы. АВЗ извлекают из выходных значений переходов аналогично [15].

Среди немногочисленных не связанных с КА методов следует упомянуть внедрение АВЗ в поведенческое описание в ходе высокоуровневого синтеза [17]. В процессе синтеза устройство управления изменяется так, чтобы транслировать выбранные внутренние промежуточные значения на выходные порты в течение выбранных свободных временных окон.

АВЗ логического уровня. В работе [13] АВЗ внедряется в неиспользуемые и частично занятые LUT (Look Up Table) на уровне списка соединений элементов (при этом LUT преобразуются в сдвиговые регистры SRL16 для предотвращения минимизации в процессе синтеза). К неиспользуемым входам SRL16 подключается сигнальный источник 0 для уменьшения его динамически адресуемого пространства. Свободные биты SRL16 заполняются битами АВЗ. Метод затрагивает функциональные элементы проектного описания, что затрудняет удаление АВЗ. При этом снижается скрытность: заземленные выводы SRL16 достаточно заметны. АВЗ извлекается путем поиска по содержимому LUT в бит-образе или путем поиска по заданным позициям.

АВЗ физического уровня. В работе [13] предлагается внедрение АВЗ в неиспользуемые LUT бит-образа, что требует знания элементов формата бит-образа. В [26] АВЗ внедряется в неиспользуемые LUT технологического описания, затем осуществляется размещение элементов проекта вокруг LUT с АВЗ. В [27] биты АВЗ внедряются в ILUT (ILUT – нефункциональный, пустой LUT в используемом CLB (Configurable Logic Block)). Для обеспечения дополнительной скрытности ILUT соединяются с неиспользуемыми входами функциональных LUT.

В [28] биты АВЗ и отпечатка пальца встраиваются в неиспользуемые LUT. Группы символов отпечатка и АВЗ размещаются в одинаковой позиции относительно друг друга: АВЗ – в первом, а отпечаток – во втором LUT неиспользуемого CLB. Тем самым облегчается извлечение АВЗ и отпечатка пальца. Концепция плиток применяется для преодоления уязвимости к атаке сговором. Плитка – это секция из CLB, среди которых хотя бы один неиспользуемый. Плитка 2x2 имеет четыре конфигурации (по положению неиспользуемого CLB). САПР размещает такой CLB в произвольной позиции, соединяя его выводы с безразличными выводами соседних элементов, а затем вокруг него – остальные CLB, входящие в плитку. Для каждого пользователя генерируется уникальный вариант проектного описания. Плитки сокращают время генерации с помощью САПР.

В [29] биты АВЗ вносятся в неиспользуемые позиции бит-образа. АВЗ не связан с исходным проектом, что упрощает его удаление. Приводится метод внедрения АВЗ в биты конфигурации мультиплексоров в неиспользуемых CLB.

Многоуровневые, иерархические методы. После того как IP-компонент интегрирован в систему на кристалле и упакован, извлечение информации об авторстве непосредственно в рабочих условиях затруднено. Многие системы снабжены средствами тестирования или самотестирования, связывающими порты с внутренними компонентами. Постановка АВЗ непосредственно на цепи сканирования выглядит недостаточно надежной мерой из-за того, что цепи добавляются на конечных стадиях разработки и могут быть удалены или заменены сравнительно легко. Поэтому появляются гибридные, или иерархические, методы, оперирующие на различных уровнях абстракции и стадиях разработки.

В работе [30] АВЗ вносятся на нескольких уровнях абстракции: на поведенческом уровне помечаются КА путем встраивания в выходные сигналы существующих и неспецифицированных переходов [23], в ходе тестопригодного проектирования помечается сканирующая цепь [20]. Подключение КА к сканирующей цепи для тестирования и пометка этой цепи АВЗ позволяют легко извлекать водяной знак в ходе функционирования. Здесь сочетаются два метода: более уязвимый для атак, но и более простой для извлечения метод встраивания АВЗ в сканирующую цепь и более устойчивый к атакам, но и более сложный для извлечения метод встраивания АВЗ в переходы КА. Менее надежный АВЗ дополнительно используется в качестве хрупкого, указывая на попытки повреждения проектного описания.

2.2. Источники извлечения АВЗ

В случае бит-образа возникают две проблемы: формат бит-образа держится в секрете производителями ПЛИС; в некоторых ПЛИС предусматривается шифрование бит-образа с запретом чтения. Пример извлечения АВЗ из LUT в бит-образе ПЛИС Xilinx приведен в [13].

Сторонние каналы интересны тем, что открывают возможность извлечения АВЗ из упакованного устройства даже в случае зашифрованных списков соединений элементов и за-

шифрованных бит-образов [13]. Среди сторонних каналов можно выделить потребляемую мощность, температуру, электромагнитное излучение. Необходимость применения дополнительного оборудования (например, высокоточного осциллографа) является недостатком таких источников извлечения.

Извлечение АВЗ в случае использования стороннего канала потребляемой мощности осуществляется путем модуляции ее профиля в соответствии с битами АВЗ. В [13] в качестве модулятора используется потребитель мощности (сдвиговый регистр). Если бит водяного знака равен единице, то сдвиговый регистр выполняет сдвиг, создавая пик потребляемой мощности; если бит равен нулю, то сдвига не происходит. В [31] предлагается усовершенствованная по ресурсам ПЛИС модификация метода, в которой в качестве модулятора используются буферы синхронизации, потребляющие 50 % динамической мощности ПЛИС. Эксперименты [13] показывают, что потребляемая мощность как источник извлечения водяных знаков имеет достаточно высокий уровень шума, поэтому требуется поиск способов модуляции и кодирования АВЗ в сигнале.

В случае электромагнитного излучения извлечение осуществляется так же, как и в случае извлечения АВЗ из потребляемой мощности [13], однако дополнительная информация о пространственном расположении источника АВЗ повышает точность извлечения. Извлечение через такой источник может осложняться наличием металлического корпуса, не пропускающего излучение, а также необходимостью дополнительного оборудования.

Проблема применения температуры как источника извлечения водяных знаков слабо освещена в литературе. В [32] описан метод использования пассивной температурной метки. Имеется внешний управляемый источник тепла, например лампа накаливания, а в цифровом устройстве находится компонент пассивной тепловой отметки, который при повышении температуры включает остальную схему. Синхронизатор определяет период передаваемого сообщения, декодер в зависимости от относительного изменения температуры на каждом такте передачи возвращает бит 0 или 1. При равенстве переданной и хранимой меток схема генерирует ответ, например отключает компонент. Недостатки данного метода: необходимость дополнительного оборудования, высокое время извлечения (16 мин для 64 битов), необходимость знания метки до извлечения для устройства (или полный перебор меток).

Порты ввода-вывода не требуют дополнительного оборудования или специфических знаний в отличие от бит-образа. Однако использовать порты не всегда возможно, потому что IP-компонент может оказаться полностью внутренним и непосредственно не контактировать с ними. В таком случае можно применить методы тестопригодного проектирования, предполагающие использование встроенных средств тестирования с доступом к тестовым портам.

Источником извлечения многих статических АВЗ являются проектные описания. Этот источник характеризуется простотой извлечения и легкостью проведения атак на АВЗ при условии, что описание незашифровано.

3. Актуальные проблемы в области АВЗ

В настоящее время важнейшей проблемой является отсутствие методологии оценки качества АВЗ. Сравнение АВЗ необходимо не только для выбора более эффективного метода, но и для разрешения споров об авторстве, когда в проектном описании содержатся АВЗ нескольких сторон. В [33] описаны разрозненные критерии оценки, в [14] предложена методология оценки, однако практически отсутствуют методы измерения конкретных критериев. Например, в [14] стоимость встраивания приравнивается к стоимости извлечения и измеряется в процессорном времени работы алгоритмов внедрения и извлечения без учета необходимости дополнительного оборудования и экспертов. Метрикой оценки издержек выступает количество проводников и новых элементов, при этом остается в стороне возможное изменение производительности, энергопотребления, надежности и других свойств проекта. Не исследован вопрос оценки стоимости проведения атак: может оказаться, что атака вполне достижима и реализуема, однако дороже, чем законное приобретение IP-компонента. Не изучены способы оценки скрытности АВЗ в проектных описаниях. Скрытность увеличивает стоимость осуществления атаки.

Важное значение имеет проблема извлечения АВЗ из упакованных IP-компонентов. Первые шаги в этом направлении представлены методами АВЗ, основанными на тестировании,

а также извлечении по сторонним каналам. Практически не развиты многоуровневые, гибридные методы, обеспечивающие комплексную защиту на разных уровнях абстракции.

Заключение

ЦВЗ широко применяются для защиты авторских прав на мультимедийные данные. Использование ЦВЗ для защиты цифровых устройств ставит ряд новых задач: внедрение ЦВЗ без нарушения функционирования, обеспечение устойчивости перед алгоритмами синтеза и оптимизациями, гарантирование высокой достоверности и низкого уровня проектных издержек. Рассмотренные методы успешно решают некоторые из этих задач, однако они все еще далеко не универсальны.

Список литературы

1. Architecture and Design Flow for a Highly Efficient Structured ASIC / H. Man-Ho [et al.] // IEEE Transactions on VLSI Systems. – 2012. – Vol. 21, iss. 3. – P. 424–433.

2. Majzoobi, M. Introduction to hardware security and trust / M. Majzoobi, F. Koushanfar, M. Potkonjak. – N.Y. : Springer, 2011. – 427 p.

3. Qu, G. Intellectual Property Protection in VLSI Design Theory and Practice / G. Qu, M. Potkonjak. – Dordrecht : Kluwer Publishing, 2003. – 203 p.

4. System-on-Chip: Reuse and Integration / R. Saleh [et al.] // Proceedings of the IEEE. – 2006. – Vol. 94, no. 6. – P. 1050–1069.

5. Can EDA Combat the Rise of Electronic Counterfeiting? / F. Koushanfar [et al.] // Design Automation Conference. – San Francisco, USA, 2012. – P. 133–137.

6. IP DirectCores. Microsemi [Electronic Resource]. – Mode of access : http://www.microsemi.com/products/fpga-soc/design-resources/ip-cores/direct-cores. – Date of access : 5.01.2015.

7. Singh, P. A Survey of Digital Watermarking Techniques, Applications and Attacks / P. Singh, R. Chadha // Intern. J. of Engineering and Innovative Technology. – 2013. – Vol. 2, iss. 9. – P. 165–175.

8. Защелкин, К. Метод внедрения цифровых водяных знаков в аппаратные контейнеры с LUT-ориентированной архитектурой / К. Защелкин, Е. Иванова // Информатика и математические методы в моделировании. – 2013. – Т. 3, № 4. – С. 369–384.

9. Digital Watermarking and Steganography / I. Cox [et al.]. – Burlington : Elsevier, 2008. – 587 p.

10. Torunoglu, I. Watermarking-Based Copyright Protection of Sequential Functions / I. Torunoglu, E. Charbon // IEEE J. of Solid-State Circuits. – 2000. – Vol. 35. – P. 434–440.

11. A Survey of Techniques for VLSI Protection / W. Liang [et al.] // Information Technology Journal. – 2013. – Vol. 12. – P. 2324–2332.

12. Collberg, C. Software Watermarking: Models and Dynamic Embeddings / C. Collberg, C. Thomborson // Proc. of the 26th ACM SIGPLAN-SIGACT symposium on Principles of programming languages. – N.Y., 1999. – P. 311–324.

13. Ziener, D. Techniques for Increasing Security and Reliability of IP Cores Embedded in FPGA and ASIC Designs / D. Ziener. – Erlangen, 2010. – 325 p.

14. Watermarking / T. Nie [et al.]. – Rijeka : InTech, 2012. – 276 p.

15. A Public-Key Watermarking Technique for IP Designs / A. Abdel-Hamid [et al.] // Design, Automation and Test in Europe, Proceedings. – 2005. – Vol. 1. – P. 330–335.

16. Abdel-Hamid, A. Fragile IP Watermarking Techniques / A. Abdel-Hamid, S. Tahar // NASA Conference on Adaptive Hardware and Systems. – Noordwijk, Netherlands, 2008. – P. 513–519.

17. Bossuet, L. Automatic low-cost IP watermarking technique based on output mark insertions / L. Bossuet, B. Gal // Design Automation for Embedded System. – 2012. – Vol. 16. – P. 71–92.

18. Rashid, A. Hierarchical Watermarking for Protection of DSP Filter Cores / A. Rashid, W. Mangione-Smith, M. Podkonjak // IEEE Custom Integrated Circuits Conference. – San Diego, USA, 1999. – P. 39–42.

19. Chapman, R. IP Protection of DSP Algorithms for System on Chip Implementation / R. Chapman, T. Durrani // IEEE Transactions on Signal Processing. – 2000. – Vol. 48. – P. 854–861.

20. Cui, A. An Improved Publicly Detectable Watermarking Scheme Based on Scan Chain Ordering / A. Cui, C.H. Chang // IEEE ISCAS, 2009. – Taipei, 2009. – P. 29–32.

21. Sequential Circuit-Based IP Watermarking Algorithm for Multiple Scan Chains in Design-for-Test / W. Liang [et al.] // Radioengineering. – 2011. – Vol. 20. – P. 533–539.

22. Oliveira, A. Robust Techniques for Watermarking Sequential Circuit Designs / A. Oliveira // Design Automation Conference. – New Orleans, USA, 1999. – P. 837–842.

23. A Robust FSM Watermarking Scheme for IP Protection of Sequential Circuit Design / A. Cui [et al.] // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 2011. – Vol. 30, no. 5. – P. 678–690.

24. Zhang, L. State Encoding Watermarking for Field Authentication of Sequential Circuit Intellectual Property / L. Zhang, C. H. Chang // IEEE ISCAS, 2012. – Seoul, 2012. – P. 3013–3016.

25. Meenakumari, M. Improving the Protection of FPGA Based Sequential IP Core Designs Using Hierarchical Watermarking Technique / M. Meenakumari, G. Athisha // J. of Theoretical and Applied Information Technology. – 2014. – Vol. 63, no. 3. – P. 701–708.

26. Lach, J. Signature Hiding Techniques for FPGA Intellectual Property Protection / J. Lach, W. Mangione-Smith, M. Podkonjak // IEEE/ACM Intern. Conf. on Computer-Aided Design. – San Jose, USA, 1998. – P. 186–189.

27. Watermarking FPGA Bitfile for Intellectual Property Protection / J. Zhang [et al.] // Radioengineering. – 2012. – Vol. 21, iss. 2. – P. 764–771.

28. Lach, J. Fingerprinting Techniques for Field-Programmable Gate Array Intellectual Property Protection // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 2001. – Vol. 20, no. 10. – P. 1253–1261.

29. Constraint-Based Watermarking Techniques for Design IP Protection / A. Kahng [et al.] // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. – 2001. – Vol. 20, no. 10. – P. 1236–1252.

30. Cui, A. A Hybrid Watermarking Scheme for Sequential Functions / A. Cui, C. H. Chang, L. Zhang // IEEE ISCAS, 2011, Rio de Janeiro. – 2011. – P. 2333–2336.

31. Clock-Modulation Based Watermark for Protection of Embedded Processors / J. Kufel [et al.] // Design, Automation & Test in Europe Conference and Exhibition. – Dresden, 2014. – P. 1–6.

32. Marsh, C. Protecting Designs with a Passive Thermal Tag / C. Marsh, T. Kean, D. McLaren // 15th IEEE Intern. Conf. on Electronics, Circuits and Systems. – Malta, 2008. – P. 218–221.

33. Abdel-Hamid, A. IP Watermarking Techniques: Survey and Comparison / A. Abdel-Hamid, S. Tahar, E. Aboulhamid // The 3rd IEEE Intern. Workshop on System-on-Chip for Real-Time Applications. – Calgary, Canada, 2003. – P. 60–65.

Поступила 27.01.2015

Белорусский государственный университет
информатики и радиоэлектроники,
Минск, ул. П. Бровки, 6
e-mail: vovasq@mail.ru
ivaniuk@bsuir.by

V.V. Sergeichik, A.A. Ivaniuk

A SURVEY OF HARDWARE WATERMARKING FOR PROGRAMMABLE LOGIC DEVICES PROTECTION

Application of watermarking technology for the protection of digital devices and their descriptions is considered. Primary definitions, models, categories of attacks, characteristics and classification of watermarks are described. Hardware watermarking examples are shown.

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 681.511.4; 004.896

В.А. Сычёв

ПРИМЕНЕНИЕ ДИНАМИЧЕСКИХ СИСТЕМ С ХАОТИЧЕСКИМ ПОВЕДЕНИЕМ В РОБОТОТЕХНИКЕ

Приводится обзор современных направлений исследований в области хаотической динамики, результаты которых находят применение в задачах управления мобильными роботами. К числу таких направлений относятся исследования в области стабилизации нелинейных динамических систем с хаотической динамикой, целенаправленной генерации хаотических колебаний, применения их результатов для обработки информации. Формулируются задачи из области управления мобильными роботами, решение которых позволит повысить автономность и функциональность роботов, действующих в недетерминированной среде.

Введение

В последние три десятилетия возросло количество исследований как естественных, так и искусственных нелинейных динамических систем (НДС), которые демонстрируют хаотические режимы работы. Термин «хаотический» применяется к таким детерминированным системам, фазовые траектории которых обнаруживают сильную зависимость от начальных условий [1, 2].

Некоторые исследователи полагают, что новый порядок (самоорганизация) в сложной системе возникает через динамический хаос – хаотический режим функционирования сложной системы [3, 4]. При этом сложной считается система, состоящая из множества взаимодействующих подсистем и обретающая в силу их взаимодействия свойства, отсутствующие у каждой из подсистем в отдельности [5, 6]. Известны практические примеры использования описываемых НДС для решения инженерных задач, в том числе в робототехнике [7].

Робототехника объединяет в себе механику, электронику и информатику. В каждой из областей, где присутствуют нелинейные элементы, могут иметь место хаотические колебания [1]. В механических системах роботов источниками нелинейности являются элементы с трением, мертвым ходом, зазором. В первую очередь это шасси мобильных роботов и редукторы [1, 8, 9]. В области электроники источником нелинейности могут служить полупроводниковые приборы, электрические и магнитные силы, прочие активные и пассивные электронные компоненты [1, 10]. В информатике хаотическая динамика чаще всего моделируется целенаправленно, с использованием различных систем дифференциальных уравнений и дискретных отображений.

Таким образом, хаотическая динамика может иметь место в ходовых и манипуляционных системах роботов, в системах управления движением, защищенной радиосвязи, хранения и обработки информации от сенсоров.

Интерес исследователей в области робототехники к НДС с хаотической динамикой объясняется тем, что до настоящего времени не созданы универсальные надежные методы управления непромышленными робототехническими аппаратами [11–13]. Исследователи продолжают вести активный поиск новых подходов к обработке и распознаванию сенсорных данных, управлению движением, осуществлению радиосвязи, хранению данных и знаний.

Благодаря уникальным свойствам НДС с хаотической динамикой предпринимаются попытки решения каждой из перечисленных задач с использованием новых знаний о хаосе.

Целью настоящей работы является повышение степени автономности мобильных роботов социального, бытового или образовательного назначения, действующих в недетерминированной среде, для чего требуется выполнить обзор актуальных задач робототехники, которые могут быть решены с использованием знаний о хаотической динамике.

1. Примеры хаотической динамики в естественных и технических системах

Хаотические режимы возникают в любых динамических, т. е. изменяющихся во времени системах, содержащих нелинейность. Некоторые примеры хаотической динамики, такие как турбулентные течения жидкости и газа, могут быть обнаружены невооруженным взглядом. Важной областью исследований в данном примере является поиск условий перехода от упорядоченного течения к турбулентности. Примером подобных исследований [14] является разработка метода бифуркационного анализа хаотических аттракторов НДС на основе теории матричной декомпозиции векторных функций в пространстве состояний [15], который позволяет выявить точку бифуркации в пространстве состояний, соответствующую либо новой ветви стационарного решения, либо периодической траектории.

Теория матричной декомпозиции находит применение и для исследования динамики таких систем, как искусственные нейронные сети [16]. Значительно более сложную динамику демонстрируют биологические нейронные сети. Примеры хаотической динамики в биологии были затронуты, в частности, У. Фрименом [17–19], эксперименты которого по исследованию процессов распознавания запахов млекопитающими выявили сложность происходящих при этом процессов. У. Фрименом высказано предположение о том, что моделирование переходов между хаотическими состояниями играет важную роль для понимания процессов, происходящих в мозгу живых существ.

Помимо нейронов и обонятельных рецепторов, хаотические процессы возникают и в других биомедицинских средах, в частности в сердечной мышце, где они могут приводить к фибрилляции либо способствовать облегченному «запуску сердца» [20, 21].

Есть основания полагать, что случайные колебания используются живыми существами для выполнения поисковых движений. В работе [22] описывается общая схема поискового адаптивного поведения с инерционным переключением между поисковыми тактиками. Как было указано в данной работе, эффективное поисковое движение может быть реализовано сочетанием перемещения на значительные расстояния, частыми случайными переменами в направлении движения и инерционностью переключения между каждым из видов движения.

Хаотическая динамика была выявлена и в таких технических устройствах, как оптические и электромеханические системы, системы массового обслуживания, где она проявляется как побочный и часто нежелательный эффект в процессе функционирования систем [7, 8, 10, 23, 24]. Однако позже стали появляться радиофизические и вычислительные устройства, предназначенные для целенаправленной генерации хаотических колебаний, которые находят применение в радиосвязи, криптографии, бытовой технике [1, 7].

Исследования в области хаотической динамики можно условно разделить на два направления. Одним из направлений, имеющих практическое применение, является стабилизация НДС с хаотической динамикой, т. е. предотвращение возникновения хаотических колебаний. При этом решение данной задачи актуально даже для устройств, первоначально функционирующих в стационарных режимах, однако в силу старения материалов и комплектующих способных со временем перейти в хаотическое состояние [20, 23, 24].

Практический интерес представляет и обратная задача, заключающаяся в целенаправленной генерации хаотических колебаний, а также связанная с ней область исследований, посвященных синхронизации НДС с хаотической динамикой. Особой развивающейся областью является информационное применение НДС с хаотической динамикой, примером которого может служить хаотический процессор [25, 26].

2. Нелинейные процессы в механических системах роботов

Малогабаритные роботы, в том числе микророботы, представляют собой развивающийся класс робототехнических аппаратов, к числу которых относятся роботы сервисного и социально-бытового назначения, функционирующие в помещениях. Как правило, размеры рабочей зоны измеряются десятками квадратных метров, т. е. сопоставимы с размерами жилых комнат, залов и т. д. Чаще всего роботы данного класса оснащаются колесным либо гусеничным шасси [11–13]. В обоих случаях используются электромоторы и многозвенные редукторы, которые

наряду с гусеничным шасси имеют нелинейные характеристики [8] и могут стать причиной возникновения хаотических процессов в ходовой части мобильного робота.

Функция жесткости однозвенного редуктора, показанная ниже, является кусочно-линейной функцией

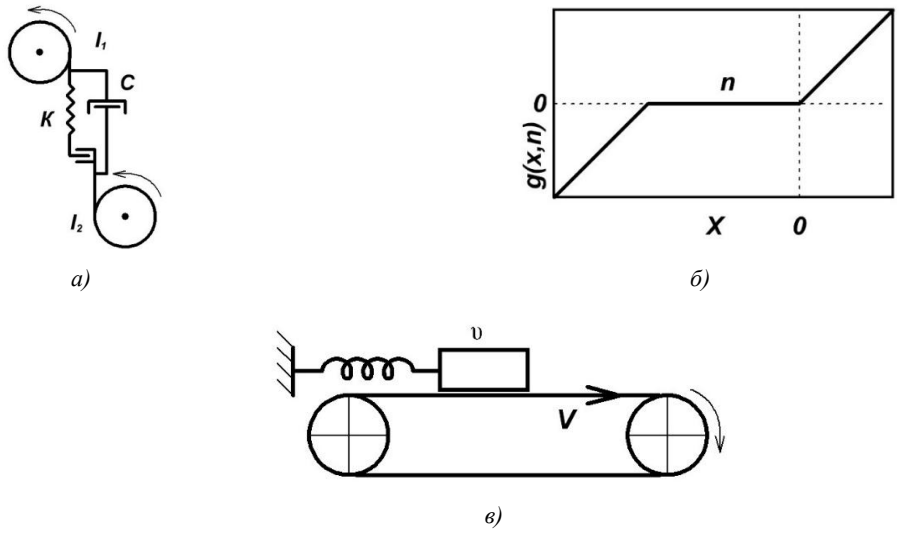
$$g(x,n)=\begin{cases} x \text{ при } x \geq 0, \\ 0 \text{ при} - n < x < 0, \\ x+n \text{ при } x \leq -n, \end{cases} \tag{1}$$

где $g(x,n)$ – функция жесткости однозвенного редуктора; n – свободный ход шестерен; x – жесткость.

На рис. 1, *а* через I_1 и I_2 обозначены моменты инерции первой и второй шестерни соответственно, K обозначает коэффициент сцепления шестерен, C – коэффициент демпфирования. На рис. 1, *б* показан график функции жесткости однозвенного редуктора (1). На рис. 1, *в* в упрощенном виде изображено шасси мобильного робота на гусеницах, представляющее собой пример сухого трения между линейным осциллятором (массой) и движущимся ремнем, где V – скорость ленты, а υ – скорость осциллятора [1, 9]. В том случае если трение между ними является сухим трением скольжения, зависящим от относительной скорости осциллятора и ленты $\upsilon - V$, выражение для силы трения имеет вид

$$R(\upsilon)=\begin{cases} R_0 \text{ при } (V-\upsilon)>0, \\ -R_0 \text{ при } (V-\upsilon)<0, \end{cases} \tag{2}$$

где R – сила трения.



а) *б)* *в)*

Рис. 1. Примеры нелинейных систем в механике: *а)* однозвенный редуктор; *б)* кусочно-линейная функция жесткости однозвенного редуктора; *в)* гусеничное шасси

Традиционные навигационные средства, такие как инерциальные навигационные системы или системы глобального позиционирования, не позволяют обеспечить требуемой точности определения местоположения робота. В то же время внутренние датчики оборотов колес или гусениц не дают возможности определить пройденный роботом путь, в том числе и в силу хаотических явлений в ходовой части. Следовательно, высокие возможности навигации подобных роботов в помещении могут быть обеспечены комплексом мер, включающим не только повышение точности используемых датчиков, но и устранение причин возникновения ошибок позиционирования.

Для исследования процессов, происходящих в ходовой части малогабаритного мобильного робота, был сконструирован экспериментальный стенд, функциональная схема которого по-

казана на рис. 2, *а*. Стенд представляет собой гусеничное шасси, оснащенное двумя коллекторными электромоторами с редукторами и измерительной аппаратурой. Так как нагрузка на оси электромотора отражается на величине потребляемого им тока, то для наблюдения за процессами в ходовой части мобильного робота измерялась величина тока, потребляемого одним из электромоторов М. Частота дискретизации составляла 9615,4 Гц при разрядности аналого-цифрового преобразователя (АЦП) 10 бит.

График на рис. 2, *б* иллюстрирует изменение во времени величины тока, потребляемого электромотором, при движении робота по ровной поверхности. Нестабильность токопотребления свидетельствует о неравномерности нагрузки на ось электромотора. Значительный вклад в создание данной неравномерности вносит резиновый трак гусеничного шасси. На рис. 2, *в* показано токопотребление электромотора с частично демонтированным редуктором – ось электромотора соединена только с одним свободно вращающимся зубчатым колесом. График токопотребления электромотора без редуктора изображен на рис. 2, *г*.

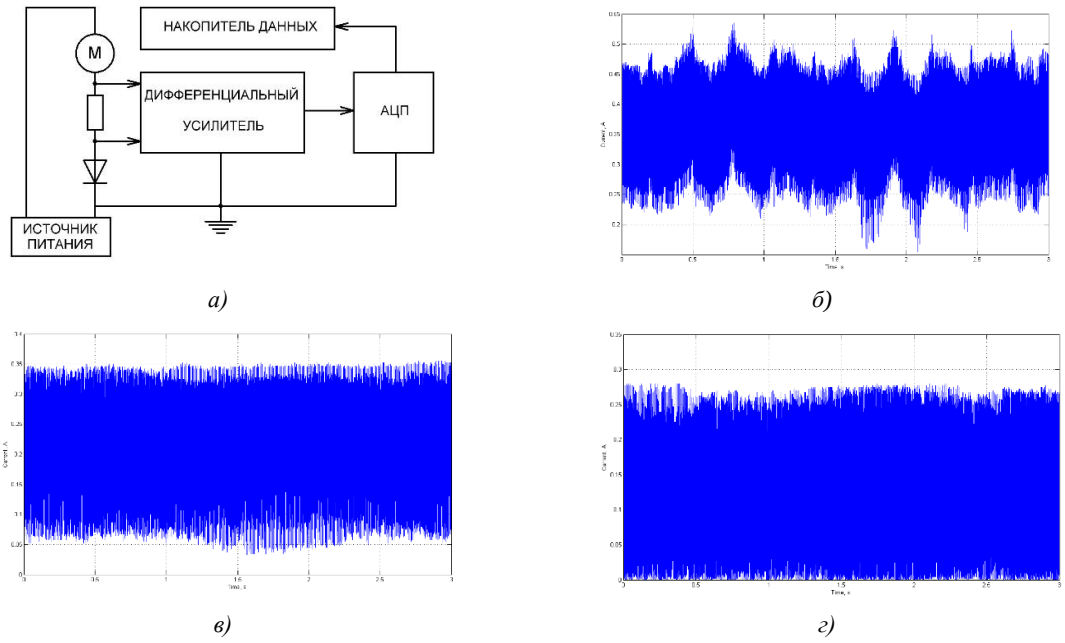


Рис. 2. Экспериментальное исследование процессов в ходовой части мобильного робота: *а)* функциональная схема экспериментального стенда; *б)* токопотребление электромотора при движении робота; *в)* токопотребление электромотора с одним зубчатым колесом; *г)* токопотребление электромотора без нагрузки

Неравномерность нагрузки на электромотор приводит к колебаниям скорости движения робота и, следовательно, к погрешности его позиционирования. Важной задачей является исследование характеристик колебаний, показанных на графиках выше, и определение их влияния на движение робота, что позволило бы повысить его стабильность и снизить ошибки позиционирования.

3. Генерация хаотических колебаний

Такие задачи, как радиосвязь на хаотических несущих, криптозащита с помощью хаоса, и многие другие практические приложения требуют целенаправленного создания хаотических колебаний, представленных как в аналоговой, так и в дискретной формах. В основе каждого генератора хаотических колебаний лежит НДС с непрерывным либо дискретным временем.

Примерами НДС с непрерывным временем, которые могут быть использованы для генерации хаотических колебаний, являются НДС, описываемые уравнениями Дуффинга, ван дер Поля, Лоренца, Ресслера, схема Чжуа [1, 10, 27]. Для генерации хаотических колебаний в системах с дискретным временем выполняется интегрирование системы дифференциальных уравнений либо итерационное вычисление дискретного отображения, для которых известны параметры, обеспечивающие хаотическую динамику [1].

В области исследований, посвященных генерации хаотических колебаний, известен ряд задач. В частности, в сфере аналоговой электроники существует потребность в развитии методов анализа и синтеза электронных схем, демонстрирующих хаотические режимы работы. Для решения данной задачи может найти применение теория матричной декомпозиции [15] как инструмент перехода от системы дифференциальных уравнений к матрицам, описывающим процессы в электронных схемах [28, 29].

В дискретных системах, оперирующих числами конечной длины, хаотические колебания используются чаще всего для криптозащиты информации и для генерации управляющих последовательностей. Нередко требуются идентичность хаотических колебаний, сгенерированных на различных программно-аппаратных платформах, и переносимость алгоритмов генерации на различные платформы (так называемое свойство кросс-платформенности), чтобы обеспечить возможность корректного кодирования и декодирования информации. Недостаточно высокая точность вычислений приводит к постепенному переходу НДС из хаотического в периодический или квазипериодический режим либо вовсе препятствует возникновению хаотического режима. Вычисления, выполненные на различных программно-аппаратных платформах, могут дать неодинаковый результат. В силу чувствительности НДС с хаотической динамикой к начальным условиям отличия в правилах хранения и обработки данных на различных платформах приводят к отличиям в хаотических аттракторах.

Таким образом, для реализации дискретного генератора хаотических колебаний необходимо использовать методы и алгоритмы, обеспечивающие необходимую точность вычислений и переносимость результатов. Важной задачей является исследование моделей дискретных НДС с хаотическими режимами для определения возможностей их использования в системах управления робототехническими аппаратами.

4. Реализация поискового движения мобильного робота

Существует ряд практических задач в области мобильной робототехники, решение которых требует выполнения поискового движения. В их числе – поиск объектов или определенных условий на местности, поиск пути при неполной или недостоверной информации об окружающей среде [11]. Один из подходов к решению различных задач робототехники, в том числе и задачи поискового движения, получил название анимат-подхода [22] и заключается в имитации поведения живых существ. Упомянутая выше схема поискового адаптивного поведения с инерционным переключением между поисковыми тактиками предполагает существование двух тактик поведения, первая из которых – движение в выбранном направлении, а вторая – случайное изменение направления движения. Переключение между тактиками поведения управляется величиной $M(t)$, которая зависит от времени следующим образом:

$$M(t) = k_1 M(t-1) + \xi(t) + I(t), \tag{3}$$

где k_1 – параметр, характеризующий инерционность переключения тактик ($0 < k < 1$); $\xi(t)$ – нормально распределенная случайная величина со средним, равным 0, и средним квадратическим отклонением σ; $I(t)$ – интенсивность раздражителя.

Предполагается, что робот (анимат в терминах [22]) в соответствии с данной тактикой движется в двухмерном пространстве *x, y*. Его задачей является поиск максимума функции $f(x, y)$. Время t робота является дискретным; следовательно, он оценивает изменение текущего значения функции $f(x, y)$ по сравнению с предыдущим тактом времени следующим образом:

$$\Delta f(t) = f(t) - f(t-1). \tag{4}$$

Интенсивность раздражителя $I(t)$ предусматривает следующие возможности:

$$I(t) = k_2 \Delta f(t) \tag{5}$$

и

$$I(t) = k_2 \Delta f(t) / f(t-1), \quad (6)$$

где $k_2 > 0$. Эмпирически установлено, что формула (6) применяется при $f(t) > 0$. Данная схема является наиболее общим и простым описанием механизма поискового поведения живых существ с инерционным переключением тактик. В то же время известны и другие реализации схем поискового поведения. Не исключено, что применение подобной схемы для управления мобильным роботом позволит эффективно решать поисковые задачи, а одним из путей реализации данной схемы поискового движения является применение генератора хаотических колебаний, потенциал которого состоит в возможности генерирования случайных величин, обеспечения требуемой инерционности и оценки интенсивности раздражителя.

Существуют работы, посвященные применению НДС для генерирования сигналов управления движением мобильного робота, например [30]. Однако в данной работе модель поискового поведения [22] не реализована, и нелинейный генератор хаотических колебаний используется лишь в контуре реактивного управления.

Реализация схемы поискового адаптивного поведения с инерционным переключением между поисковыми тактиками на основе генератора хаотических колебаний и включение ее в контур реактивного управления мобильного робота являются перспективными способами реализации поискового движения для мобильных роботов, действующих в недетерминированной среде.

5. Информационное применение систем с хаотической динамикой

Один из примеров использования систем с хаотической динамикой для обработки информации предложен в работе [31] и основывается на чувствительности НДС с хаотической динамикой к начальным условиям. Принцип, предложенный в данной работе, подразумевает включение в цепь управления генератора хаотических колебаний датчика какой-либо физической величины таким образом, чтобы малейшее изменение состояния датчика приводило бы к значительным изменениям динамики генератора. По результатам идентификации динамики генератора может быть определено изменение состояния датчика. Ограничивает же применение описанного подхода сложность идентификации генератора.

Другим вариантом информационного применения систем с хаотической динамикой является хаотический процессор [25, 26]. В основу хаотического процессора положена гипотеза о существовании общих принципов и закономерностей обработки информации в системах со сложной динамикой, не зависящих от конкретного вида и реализации самих систем. Эта гипотеза выдвигается на основании анализа экспериментальных данных и теоретических представлений об информационных процессах в живых системах. В работе [26] описана модель НДС, способная реализовать процесс хранения и извлечения данных, записанных на предельном цикле одно- или многомерного отображения. Вариантом практического применения хаотического процессора является реализация функций ассоциативной памяти [32].

Несмотря на перспективность применения хаотического процессора в робототехнических задачах, ему препятствует ряд трудностей. К примеру, одним из возможных применений ассоциативной памяти является фильтрация информации. В этом случае хаотический процессор должен получать на вход предварительно обработанные данные, в то время как было бы желательно осуществлять все этапы обработки с помощью самого хаотического процессора.

В том случае, когда хаотический процессор используется в качестве ассоциативной памяти, а на его вход подаются данные, в памяти не содержащиеся, на выходе процессора формируются хаотические колебания. Трудность при этом представляет автоматическая идентификация выходного состояния процессора.

Заключение

Паразитные хаотические процессы выявляются во многих системах роботов. В то же время целенаправленно сгенерированные хаотические колебания могут быть использованы для решения ряда практических задач.

На основании проведенного обзора можно выделить ряд приоритетных задач для исследований, проводимых на стыке нелинейной динамики и робототехники, решение которых позволит повысить автономность мобильных роботов. В области механики это задачи предотвращения возникновения хаотических процессов в редукторах и ходовой части роботов. В аналоговой электронике интерес представляет разработка новых методов анализа и синтеза электронных схем с хаотическими режимами. В системах с дискретным временем важной задачей является разработка кросс-платформенных методов генерации хаотических колебаний.

Практическое применение НДС с хаотической динамикой является перспективным для реализации поискового движения мобильных роботов, а также для хранения и обработки информации.

Работа выполнена при поддержке гранта БРФФИ–ГФФИУ № Ф13К-144 «Разработка методов оперативной обработки и передачи информации для эффективного управления мобильными роботами и подвижными системами».

Список литературы

1. Moon, F. Chaotic Vibrations: An Introduction for Applied Scientists and Engineers / F. Moon. – John Wiley&Son, 2004. – 309 p.

2. Кроновер, Р.М. Фракталы и хаос в динамических системах. Основы теории / Р.М. Кроновер. – М. : Постмаркет, 2000. – 352 с.

3. Пригожин, И. Порядок из хаоса / И. Пригожин, И. Стенгерс. – М. : Прогресс, 1986. – 431 с.

4. Малинецкий, Г.Г. Хаос. Структуры. Вычислительный эксперимент: введение в нелинейную динамику / Г.Г. Малинецкий. – М. : Эдиториал УРСС, 2000. – 256 с.

5. Хакен, Г. Тайны природы. Синергетика: учение о взаимодействии / Г. Хакен. – Москва – Ижевск : Институт компьютерных исследований, 2003. – 320 с.

6. Лоскутов, А.Ю. Основы теории сложных систем / А.Ю. Лоскутов, А.С. Михайлов. – Москва – Ижевск : Институт компьютерных исследований, 2007. – 620 с.

7. Chaos in Automatic Control / W. Perruquetti [et al.] ; ed. W. Perruquetti. – CRC Pres, 2005. – 564 p.

8. Radons, G. Nonlinear Dynamics of Production Systems / G. Radons, R. Neugebauer ; ed. Günter Radons, Reimund Neugebauer. – Weinheim : John Wiley & Sons, 2006. – 647 p.

9. Эбелинг, В. Образование структур при необратимых процессах / В. Эбелинг. – М. : Мир, 1979. – 279 с.

10. Nonlinear Dynamics in Circuits / T. Carroll [et al.] ; ed. T. Carroll. – World Scientific, 1995. – 344 p.

11. Каляев, И.А. Модели и алгоритмы коллективного управления в группах роботов / И.А. Каляев, А.Р. Гайдук, С.Г. Капустян. – М. : Физматлит, 2009. – 280 с.

12. Интеллектуальные роботы : учебное пособие для вузов / В.А. Каляев [и др.] ; под общей ред. Е.И. Юревича. – М. : Машиностроение, 2007. – 360 с.

13. Юревич, Е.И. Основы робототехники / Е.И. Юревич. – СПб. : БХВ-Петербург, 2010. – 416 с.

14. Baldin, V.A. The development of model for boundary layers past a concave wall with usage of nonlinear dynamics methods / V.A. Baldin, A.M. Krot, H.B. Minervina // Advances in Space Research. – 2006. – Vol. 37, no. 3. – P. 501–506.

15. Крот, А.М. Анализ аттракторов сложных нелинейных динамических систем на основе матричных рядов в пространстве состояний / А.М. Крот // Информатика. – 2004. – №1. – С. 7–16.

16. Krot, A.M. Nonlinear analysis of the Hopfield network dynamical states using matrix decomposition theory / A.M. Krot, R.A. Prakapovich // Chaotic modeling and simulation . – 2013. – Vol. 1. – P. 133–146.

17. Freeman, W.J. Spatial properties of an EEG in the olfactory bulb and olfactory cortex / W.J. Freeman // Electroencephalography and Clinical Neurophysiology. – 1978. – № 44. – С. 586–605.

18. Freeman, W.J. Mesoscopic neurodynamics: From neuron to brain / W.J. Freeman // Mesoscopic neurodynamics: From neuron to brain. – 2000. – Vol. 94, iss. 5–6. – С. 303–322.

19. Malsburg, C. von der. The correlation theory of brain function / C. von der Malsburg // Max-Plank. Institut for Biophys. Chem. – Germany, 1981. – Vol. 81-2.

20. Чжуа, Л.О. Хаотические системы / Л.О. Чжуа // Хаотические системы. Тематический выпуск : труды Института инженеров по электротехнике и радиоэлектронике. – 1987. – Т. 75, № 8. – С. 4–5.

21. Dailyudenko, V.F. Linearized Analysis of Complexity and Stability For Propagating Recovery Model of Active Medium / V.F. Dailyudenko // Chaos and Complexity Research Compendium. – N.Y. : Nova Science Publishers, 2013. – Vol. 3. – P. 113–140.

22. Непомнящих, В.А. Бионическая модель адаптивного поискового поведения / В.А. Непомнящих, Е.Е. Попов, В.Г. Редько // Известия РАН. Теория и системы управления. – 2008. – № 1. – С. 85–93.

23. Паркер, Т.С. Введение в теорию хаотических систем для инженеров / Т.С. Паркер, Л.О. Чжуа // Хаотические системы. Тематический выпуск : труды Института инженеров по электротехнике и радиоэлектронике. – 1987. – Т. 75, № 8. – С. 6–40.

24. Handbook of Chaos Control / E.Schöll [et al.] ; ed. E.Schöll. – 2nd ed. – Wiley-VCH Verlag GmbH&Co.KGaA, 2008. – 819 p.

25. Dmitriev, A.S. Basic principles of direct chaotic communications / A.S. Dmitriev [et al.] // Nonlinear Phenomena in Complex Systems. – 2003. – Vol. 6, no. 1. – P. 488–501.

26. Андреев, Ю.В Хаотические процессоры / Ю.В. Андреев, А.С. Дмитриев, Д.А. Куминов // Успехи современной радиоэлектроники. – 1997. – № 10. – С. 50–79.

27. Хаслер, М.Ж. Электрические схемы с хаотическим поведением / М.Ж. Хаслер // Хаотические системы. Тематический выпуск : труды Института инженеров по электротехнике и радиоэлектронике. – 1987. – Т. 75, № 8. – С. 40–54.

28. Анго, А. Математика для электро- и радиоинженеров / А. Анго. – М. : Наука, 1967. – 780 с.

29. Мэзон, С. Электронные цепи, сигналы и системы / С. Мэзон, Г. Циммерман. – М. : Наука, 1963. – 619 с.

30. Clarck, M. Coupled Oscillator Control of Autonomous Mobile Robots / M. Clarck, T. Anderson, R. Skinner // Autonomous Robots. – Kluwer Academic Publishers, 2000. – P. 189–198.

31. Чернухо, Е.В. Моделирование и анализ свойств синергетического метода измерения : автореф. дис. … канд. техн. наук : 05.11.13 / Е.В. Чернухо ; НАН Беларуси, Ин-т прикладной физики. – Минск, 2000. – 24 с.

32. Прокопович, Г.А. Нейросетевые модели интеллектуальных систем управления робототехническими аппаратами : автореф. дис. … канд. техн. наук : 05.13.18 / Г.А. Прокопович ; ОИПИ НАН Беларуси. – Минск, 2013. – 24 с.

Поступила 10.11.2014

Объединенный институт проблем
информатики НАН Беларуси,
Минск, ул. Сурганова, 6
e-mail: vsychyov@robotics.by

U.A. Sychou

APPLICATION OF CHAOTIC DYNAMICAL SYSTEMS IN ROBOTICS

An overview of modern directions of research in the field of chaotic dynamics is provided. The results of the research can be used for the control of mobile robots. There are such directions of research as stabilization of nonlinear dynamical systems with chaotic dynamics, generation of chaotic vibrations and their applications in information processing. The tasks of mobile robots control that allow to increase functionality and autonomy of robots operating in non-deterministic environment are formulated.

ПРАВИЛА ДЛЯ АВТОРОВ

1. Статьи принимаются в редакцию через электронную систему подачи по адресу **http://jinfo.bas-net.by** в формате файлов текстовых редакторов Microsoft Word 97 и Word 2000 для Windows. Основной текст статьи набирается с переносами шрифтом Times New Roman 11 пт, интервал между строками – одинарный, абзацный отступ 1 см, поля по 2,5 см со всех сторон.

2. Статья должна иметь индекс УДК (универсальная десятичная классификация).

3. Название статьи, фамилии всех авторов и аннотация должны быть переведены на английский язык. Для каждого из авторов приводится развернутое название учреждения с полным почтовым адресом, а также номер телефона и электронный адрес (e-mail) для связи с редакцией.

4. Формулы, иллюстрации, таблицы, встречающиеся в статье, должны быть пронумерованы в соответствии с порядком цитирования в тексте. Ссылки на рисунки и таблицы в тексте обязательны. Необходимо избегать повторения одних и тех же данных в таблицах, графиках и тексте статьи.

Рисунки должны быть выполнены с хорошим разрешением в масштабе, позволяющем четко различать надписи и обозначения. Подрисуночные подписи с расшифровкой всех позиций, представленных на рисунке, набираются шрифтом гарнитуры основного текста, размер символов 9 пт. Цветные иллюстрации печатаются только в том случае, когда это необходимо для понимания излагаемого материала.

5. Набор формул выполняется в формульных редакторах Microsoft Equation или Math Type и должен быть единообразным по применению шрифтов и знаков по всей статье.

Прямо (_) набираются: греческие и русские буквы; математические символы (sin, lg, ∞); символы химических элементов (C, Cl, $CHCl_3$); цифры (римские и арабские); векторы; индексы (верхние и нижние), являющиеся сокращениями слов.

Курсивом (~) набираются: латинские буквы – переменные, символы физических величин (в том числе и в индексе).

6. Сокращения в тексте статьи (за исключением единиц измерения) могут быть использованы только после упоминания полного термина. Единицы измерения физических величин следует приводить в Международной системе СИ.

7. Литература приводится автором общим списком в конце статьи. Ссылки на литературу в тексте идут по порядку и обозначаются цифрой в квадратных скобках. Ссылаться на неопубликованные работы не допускается. С примерами оформления библиографического описания в списке литературы можно ознакомиться в приложении 2 к *Инструкции по оформлению диссертации, автореферата и публикаций по теме диссертации* на сайте Высшей аттестационной комиссии Республики Беларусь **http://vak.org.by**.

8. Поступившие в редакцию статьи направляются на рецензирование специалистам. Основным критерием целесообразности публикации является новизна и информативность статьи. Если по рекомендациям рецензента статья возвращается автору на доработку, а переработанная рукопись вновь рассматривается редколлегией, датой поступления считается день получения редакцией ее окончательного варианта. Статьи не по профилю журнала возвращаются авторам после заключения редколлегии.

9. Статьи, направляемые на доработку, должны быть возвращены в исправленном виде с ответами на все вопросы.

10. Редакция журнала предоставляет возможность первоочередного опубликования статей, представленных лицами, которые осуществляют послевузовское обучение (аспирантура, докторантура, соискательство) в год завершения обучения.

11. Авторы несут ответственность за направление в редакцию статей, уже опубликованных ранее, или статей, принятых к публикации другими изданиями.

12. Редакция оставляет за собой право на редакционные изменения, не искажающие основное содержание статьи.

Журнал «Информатика» включен Высшей аттестационной комиссией Республики Беларусь в список научных изданий для опубликования результатов диссертационных исследований (журнал «Аттестация», № 2, 2004, с. 81).

Индексы

00827

для индивидуальных подписчиков

008272

для предприятий и организаций

ИНФОРМАТИКА 1 (45) январь–март 2015