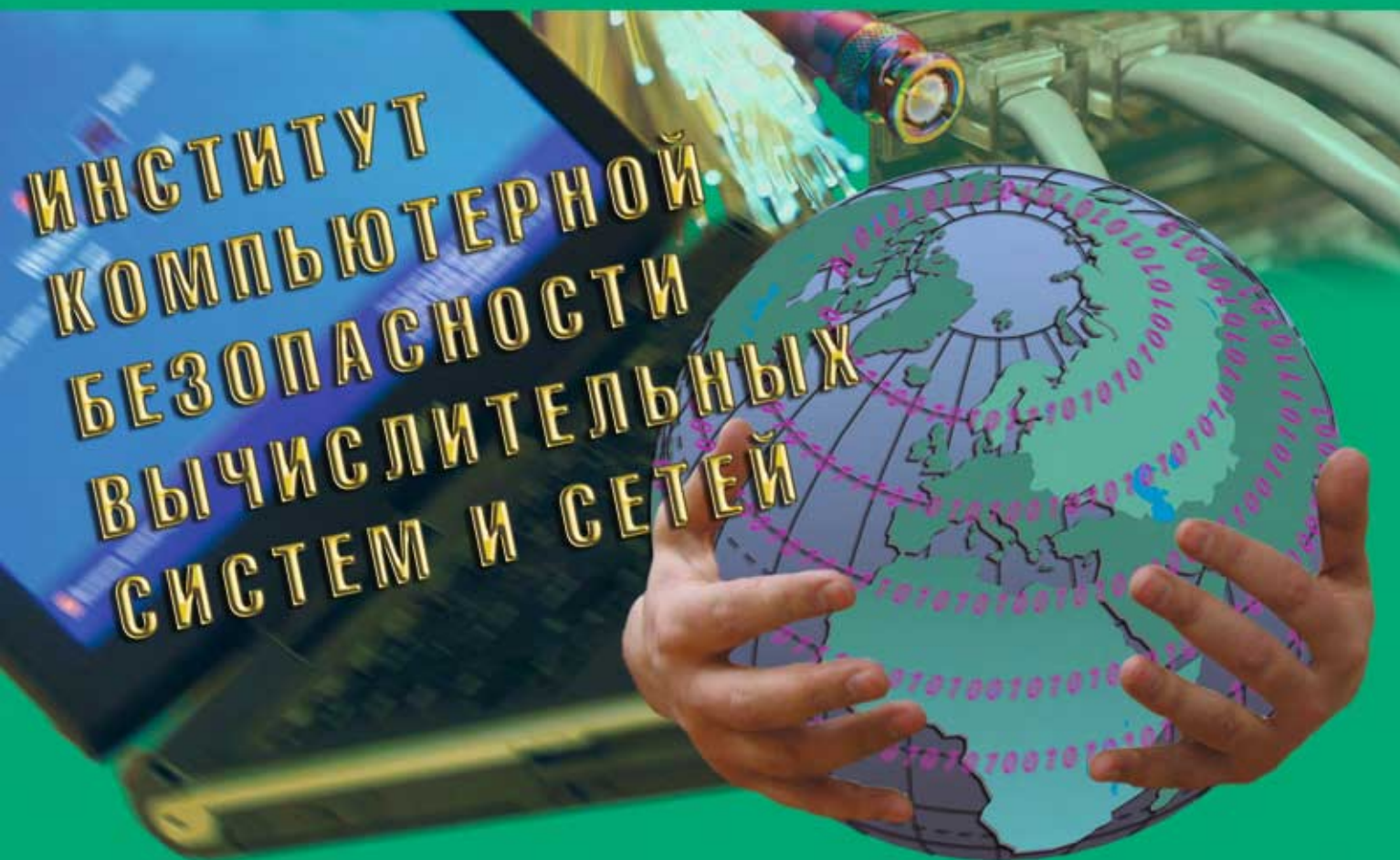


ИНФОРМАЦИОННО- УПРАВЛЯЮЩИЕ СИСТЕМЫ

НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ



1(14)/2005

Главный редактор
М. Б. Сергеев,
доктор технических наук, профессор

Зам. главного редактора
Г. Ф. Мощенко

Редакционный совет:

Председатель А. А. Оводенко,
доктор технических наук, профессор
В. Н. Васильев,
доктор технических наук, профессор
В. Н. Козлов,
доктор технических наук, профессор
Ю. Ф. Подоплекин,
доктор технических наук, профессор
Д. В. Пузанков,
доктор технических наук, профессор
В. В. Симаков,
доктор технических наук, профессор
А. Л. Фрадков,
доктор технических наук, профессор
Л. И. Чубраева,
доктор технических наук, профессор, чл.-корр. РАН
Р. М. Юсупов,
доктор технических наук, профессор

Редакционная коллегия:

В. Г. Анисимов,
доктор технических наук, профессор
В. Ф. Мелехин,
доктор технических наук, профессор
А. В. Смирнов,
доктор технических наук, профессор
В. А. Фетисов,
доктор технических наук, профессор
В. И. Хименко,
доктор технических наук, профессор
А. А. Шалыто,
доктор технических наук, профессор
А. П. Шелета,
доктор технических наук, профессор
З. М. Юлдашев,
доктор технических наук, профессор

Редакторы: О. А. Рубинова, Л. М. Манучарян
Корректор: Т. Н. Гринчук
Дизайн: М. Л. Черненко
Компьютерная верстка: О. В. Васильева,
А. А. Буров
Ответственный секретарь: О. В. Муравцова

Адрес редакции: 190000, Санкт-Петербург,
Б. Морская ул., д. 67
Тел.: (812) 110-66-42, (812) 313-70-88
Факс: (812) 313-70-18
E-mail: ius@aanet.ru

Журнал зарегистрирован в Министерстве РФ по делам печати,
телерадиовещания и средств массовых коммуникаций
Свидетельство о регистрации ПИ № 77-12412 от 19 апреля 2002 г

Журнал распространяется по подписке. Подписку можно оформить в любом
отделении связи по каталогам агентства «Роспечать».
«Газеты и журналы» – № 15385.
«Издания органов НТИ» – № 69291

ОБРАБОТКА ИНФОРМАЦИИ И УПРАВЛЕНИЕ

Гагарин К. Ю. Математические модели и быстрые алгоритмы векторно-разностного кодирования цифровых речевых сигналов 2

МОДЕЛИРОВАНИЕ СИСТЕМ И ПРОЦЕССОВ

Зикратов И. А., Степаненко К. В. Обоснование масштаба цифровых карт местности, используемых при расчете напряженности поля радиосигналов 10

ПРОГРАММНЫЕ И АППАРАТНЫЕ СРЕДСТВА

Шамгунов Н. Н., Корнеев Г. А., Шалыто А. А. State Machine – расширение языка Java для эффективной реализации автоматов 16

КОДИРОВАНИЕ И ПЕРЕДАЧА ИНФОРМАЦИИ

Белоголовый А. В., Крук Е. А. Многопороговое декодирование кодов с низкой плотностью проверок на четность 25

Овчинников А. А. К вопросу о построении LDPC-кодов на основе Евклидовых геометрий 32

Трифонов П. В. Адаптивная передача в многопользовательских многочастотных системах вещания 41

Мельник И. А. Новый метод оценки алгоритмов распределения QVSF-кодов в стандарте WCDMA 46

ХРОНИКА И ИНФОРМАЦИЯ

Юсупов Р. М. К 90-летию академика Е. П. Попова 51

Памяти Виктора Ильича Варшавского 58

Международный семинар «Образование для всех» 59

СВЕДЕНИЯ ОБ АВТОРАХ

60

АННОТАЦИИ

62

ЛР № 010292 от 18.08.98.
Сдано в набор 30.12.2004. Подписано в печать 15.02.2005. Формат 60×90/8.
Бумага офсетная. Гарнитура Pragmatica. Печать офсетная.
Усл. печ. л. 8,0. Уч.-изд. л. 9,0. Тираж 1000 экз. Заказ 20.

Оригинал-макет изготовлен
в отделе электронных публикаций и библиографии ГУАП.
190000, Санкт-Петербург, Б. Морская ул., 67

Отпечатано с готовых диапозитивов
в отделе оперативной полиграфии ГУАП.
190000, Санкт-Петербург, Б. Морская ул., 67

УДК 621.3

МАТЕМАТИЧЕСКИЕ МОДЕЛИ И БЫСТРЫЕ АЛГОРИТМЫ ВЕКТОРНО – РАЗНОСТНОГО КОДИРОВАНИЯ ЦИФРОВЫХ РЕЧЕВЫХ СИГНАЛОВ

К. Ю. Гагарин,

канд. техн. наук, докторант

Санкт-Петербургский государственный политехнический университет

Представлены математические модели и алгоритмы векторно-разностных схем кодирования речевых сигналов на основе многоканальных линейных систем. Предложены алгоритмы вычисления автокорреляционных матриц через быстрые гиперкомплексные преобразования Фурье.

In this paper we present a mathematical models and algorithms of vector-differential coding of speech signals based on multi-channel linear systems theory. Efficient algorithms of autocorrelation arrays computation through hyper complex Fourier transformations are also proposed.

Разностное кодирование относится к методам сжатия цифровых сигналов через кодирование их формы.

Наиболее широкое практическое применение получили скалярно-разностные алгоритмы кодирования, которые строятся на основе скалярно-разностных уравнений

$$d(n) = x(n) - \hat{x}(n), \quad (1)$$

где $x(n)$ – n -й отсчет кодирования сигнала; $\hat{x}(n)$ – предсказанное значение для $x(n)$; $d(n)$ – погрешность предсказания.

На основе разностного уравнения (1) разработаны алгоритмы дельта-модуляции (двухуровневое квантование) и дифференциальной импульсно-кодовой модуляции (многоуровневое квантование), а также алгоритмы с кратко- и долгосрочным предсказанием [1, 2].

В данной статье предложены математические модели векторно-разностного кодирования речевых сигналов с многоканальным линейным предсказанием, которые могут служить одним из способов теоретического обобщения других известных методов сжатия, например, с кратко- и долгосрочным предсказанием или с векторным квантованием [2].

В работах [3, 4] было предложено использовать для кодирования формы речевых сигналов векторно-разностные (ВР) уравнения

$$\bar{d}(m) = X(m) - \bar{X}(m), \quad (2)$$

где $\bar{d}(m)$, $X(m)$ и $\bar{X}(m)$ – векторы соответственно погрешностей предсказания, последовательностей отсчетов сигнала $\{x(n)\}$ и последовательности

предсказанных значений $\{\hat{x}(n)\}$. Схема кодера ВР-кодирования (рис. 1) отличается от скалярно-разностных схем наличием Ф_В-блока формирования векторов и операции сложения-вычитания и квантования выполняются над векторами, т. е. над каждым элементом вектора отдельно и независимо друг от друга. Такой способ кодирования относится к кодированию формы сигналов, образованных одноименными $x_i(k)$ компонентами последовательности векторов $\{X(k)\}$.

Для случая, когда длина вектора определяется величиной периода основного тона речевого сигнала, на рис. 2 приведены графики сигналов $\{x_i(k)\}$ и $\{x(n)\}$. Приведенные примеры форм сигналов $x_i(k)$ и $x(n)$ указывают на высокую степень корреляции между соседними отсчетами внутри и между векторов.

Поэтому как практический, так и теоретический интерес представляют алгоритмы ВР-кодирования, позволяющие учитывать оба вида корреляции отсчетов сигнала.

В работе [5] были предложены математические модели предсказателей в схемах ВР-кодирования в виде многоканальных линейных систем. Рассмотрим более подробно методы ВР-кодирования с учетом особенностей используемых математических моделей.

ВР-кодирование с независимой межканальной обработкой отсчетов сигнала

При независимой межканальной обработке отсчетов сигнала квантование и кодирование осуществляется для каждого i -го сигнала, по которому пе-

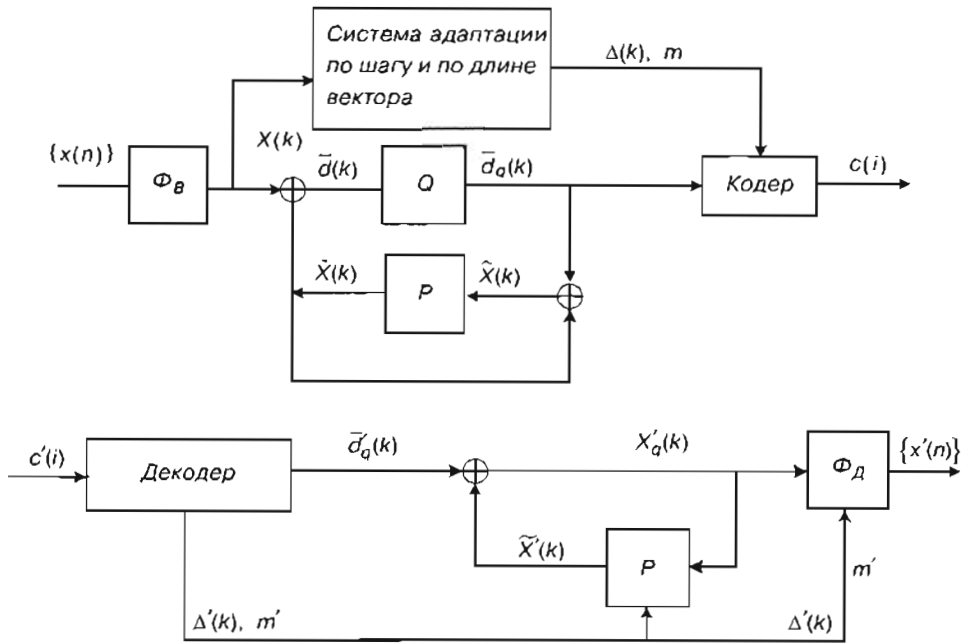


Рис. 1. Схема ВР-кодирования с адаптацией по шагу квантования и по длине вектора

ствляется для каждого i -го сигнала, по которому передается сигнал $x_i(k)$. При этом, согласно векторно-разностному уравнению (2), предсказанные значения $\{\hat{x}_i(k)\}$ формируются независимо друг от друга.

Для определения вектора $\hat{X}_m(k)$ предсказанных значений кодируемого вектора $X(k)$ можно записать векторно-матричное разностное уравнение

$$\hat{X}_m(k) = \sum_{l=1}^p D_m(l) \hat{X}(k-l), \quad (3)$$

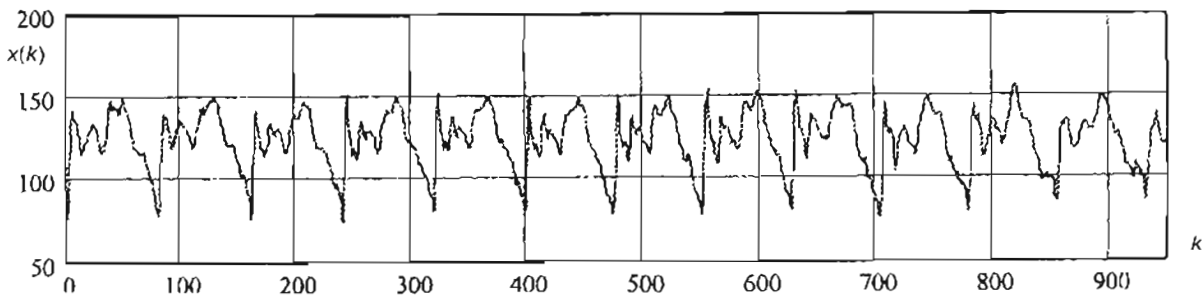
где $D_m(l) = \text{diag}\{\alpha_i(l)\}_{i=0}^{m-1}$ – диагональная матрица коэффициентов предсказания; $\hat{X}(k-l)$ – вектор квантованных значений кодируемого сигнала. Для $p = 1$ выражение может быть записано в виде

$$\hat{X}_m(k) = D_m(1) \hat{X}(k-1),$$

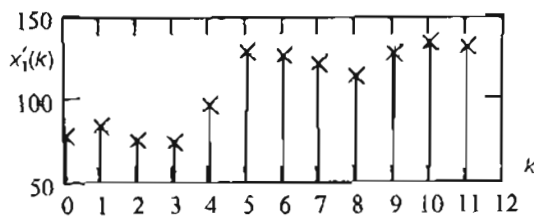
где $D_m(1) = I_m$ – единичная матрица.

Для i -го сигнала $\{x_i(k)\}$ являются справедливыми известные соотношения [6], определяющие каче-

а)



б)



в)

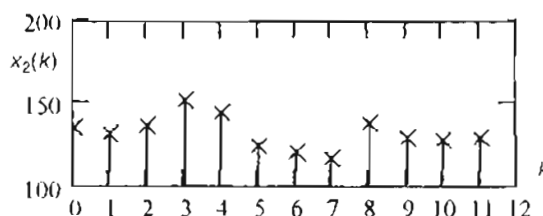


Рис. 2. Графики сигналов: а – $x(n)$, $n = \overline{0,900}$; б – $x_1(n)$, $n = \overline{0,13}$; в – $x_2(k)$, $k = \overline{0,12}$

ство кодирования. Например, отношение сигнал/шум (С/Ш) можно записать через отношение дисперсии погрешности $\sigma_{\hat{d}_i}^2$ предсказания к дисперсии погрешности квантования $\sigma_{e_i}^2$,

$$(C/Ш)_i = \sigma_{\hat{d}_i}^2 / \sigma_{e_i}^2.$$

Для коэффициентов усиления в i -м канале имеем

$$G_{p_i} = \sigma_{\hat{x}_i}^2 / \sigma_{\hat{d}_i}^2.$$

Таким образом, ВР-кодирование с независимой межканальной обработкой может быть осуществлено с использованием в каждом из каналов скалярно-разностных алгоритмов, в том числе алгоритмов с линейным предсказанием.

При этом для определения оптимальных коэффициентов предсказания, представленных матрицей

$$D_m^{(v)}(l) = \text{diag} \left\{ \alpha_v^{(i)}(l) \right\}_{i=1}^m,$$

можно, например, воспользоваться известным [6] соотношением, связывающим коэффициенты предсказания на v -м интервале времени с кратковременной автокорреляционной функцией

$$j = \overline{1, p} \quad R_v^{(i)}(j) = \sum_{l=1}^p \alpha_v^{(i)}(l) R_v^{(i)}(j-l), \quad (4)$$

которое может быть представлено в матричной форме

$$\bar{R}_p = [R_p] \bar{\alpha}_p, \quad (5)$$

где \bar{R}_p – вектор значений $R_v^{(i)}(j)$; $[R_p]$ – автокорреляционная матрица; $\bar{\alpha}_p$ – вектор коэффициентов предсказания $\alpha_v^{(i)}(l)$.

Главной отличительной особенностью ВР-кодирования является обновление на каждом периоде основного тона среднего значения приращения величины отсчетов в каждом канале либо среднего приращения величины отсчетов по всему периоду в целом. Это позволяет значительно расширить возможности ВР-кодирования в части повышения коэффициентов сжатия и помехоустойчивости с сохранением высокого качества восстановления речи.

Экспериментальные исследования ВР-кодирования с дельта-модуляцией показали их значительные преимущества по сравнению с дельта-модуляцией скалярно-разностных алгоритмов.

ВР-кодирование с межканально-зависимым линейным предсказанием

При межканально-зависимом ВР-кодировании речевых сигналов в качестве математической модели предсказателя может выступать векторно-матричное уравнение

$$\tilde{X}_{n_i}(k) = \sum_{l=1}^p A_m(l) \hat{X}(k-l), \quad (6)$$

где $A_m(l) = [\alpha_{ij}(l)]$, $i, j = \overline{1, m}$ – квадратная матрица коэффициентов предсказания.

Разностному уравнению (6) может быть сопоставлена структурная схема предсказателя (рис. 3), где входами являются компоненты $\hat{x}_i(k-l)$ вектора $\hat{X}(k-l)$, а выходами – компоненты $\tilde{x}_i(k)$ вектора предсказанных значений $\tilde{X}(k)$, определяемые в виде скалярных произведений $(\bar{\alpha}_i(l), \hat{X}(k-l))$ векторов $\bar{\alpha}_i(l)$ и $\hat{X}(k-l)$.

При подстановке (6) в уравнение (2) выражение для вектора погрешности предсказания принимает вид

$$\bar{d}(k) = X(k) - \sum_{l=1}^p A_m(l) \hat{X}(k-l) \quad (7)$$

Оптимальные коэффициенты предсказания, заданные в форме матрицы $A_m(l)$, можно определить из условия минимизации дисперсий $\sigma_{\bar{d}}^2$ элементов вектора $\bar{d}(l)$ посредством решения системы уравнений

$$\frac{\partial \sigma_{\bar{d}}^2}{\partial \alpha_{ij}} = 0,$$

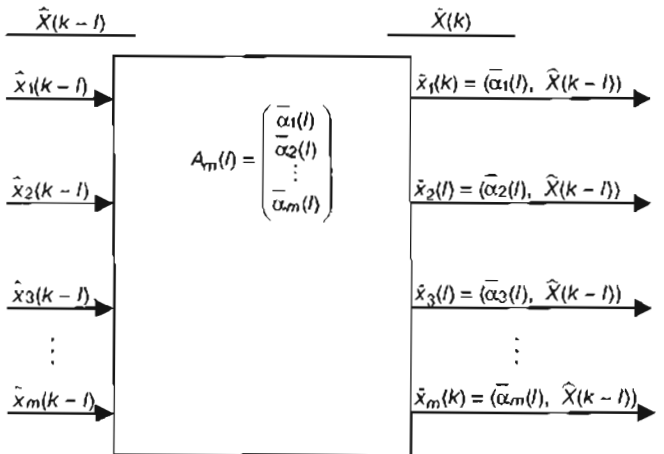
где $\sigma_{\bar{d}}^2 = E(\bar{d}^2(k)) = E((X(k) - \tilde{X}(k))^2)$, $E(\cdot)$ – оператор осреднения.

При этом предполагается, что среднее $\hat{m}_{d_i} = 0$.

Выполнив преобразования для каждого элемента $\tilde{d}_i(k)$, аналогичные скалярно-разностным алгоритмам кодирования, можно получить следующую векторно-матричную форму, связывающую коэффициенты предсказания с блочными корреляционными матрицами

$$R_{xx}(j) = \sum_{l=1}^p A_m(l) R_{xx}(j-l), \quad j = \overline{1, p} \quad (8)$$

От последовательностной векторно-матричной формы (8) можно перейти к блочно-матричной форме, через которую будет задана система линей-



■ Рис. 3. Схема ВР-предсказателя

ных уравнений, решаемых относительно оптимальных коэффициентов предсказания $\alpha_{ij}(l)$:

$$\bar{R}_p = R_p \bar{A}_p, \quad (9)$$

где \bar{R}_p – блочно-матричный вектор-столбец $\bar{R}_p = R'_{xx}(1), R'_{xx}(2), \dots, R'_{xx}(p)$; $\bar{A}(p)$ – блочно-матричный вектор-столбец $\bar{A}' = A'(1), A'(2), \dots, A'(p)$; \hat{R}_p – блочно-теплица корреляционная матрица [7]

$$\hat{R}_p = \begin{pmatrix} R_{xx}(0) & R_{xx}(1) & \dots & R_{xx}(p-1) \\ R_{xx}(1) & R_{xx}(0) & \dots & R_{xx}(p-2) \\ \dots & \dots & \dots & \dots \\ R_{xx}(p-1) & R_{xx}(p-2) & \dots & R_{xx}(0) \end{pmatrix},$$

где $R_{xx}(k) = \frac{1}{N} \sum_{n=0}^{N-k-1} (X(n+k) X'(n))$ – смещенная оценка многоканальной корреляционной матрицы

$$R_{xx} = \begin{pmatrix} r_{11}(k) & r_{12}(k) & \dots & r_{1m}(k) \\ r_{21}(k) & r_{22}(k) & \dots & r_{2m}(k) \\ \dots & \dots & \dots & \dots \\ r_{m1}(k) & r_{m2}(k) & \dots & r_{mn}(k) \end{pmatrix}$$

К блочно-матричной форме (9) можно применить быстрый алгоритм решения заданной через нее системы линейных уравнений. При этом сложность построения быстрых алгоритмов значительно зависит от величины параметра m – количества каналов (или длины вектора). В общем случае матрицы \hat{R}_p не являются теплицевыми, поэтому возможность применения быстрого алгоритма продемонстрируем на разложении Холецкого [6] для значений параметров $m = 2$ и $p = 2$. В этом случае блочно-матричная форма (10) принимает вид

$$\begin{pmatrix} R_{xx}(0) & R_{xx}(1) \\ R_{xx}(1) & R_{xx}(0) \end{pmatrix} \begin{pmatrix} A_2(1) \\ A_2(2) \end{pmatrix} = \begin{pmatrix} R_{xx}(1) \\ R_{xx}(2) \end{pmatrix}. \quad (10)$$

Раскрывая в уравнении (10) блоки-матрицы, получим

$$\begin{pmatrix} r_{11}(0) & r_{12}(0) & r_{11}(1) & r_{12}(1) \\ r_{21}(0) & r_{22}(0) & r_{21}(1) & r_{22}(1) \\ r_{11}(1) & r_{12}(1) & r_{11}(0) & r_{12}(0) \\ r_{21}(1) & r_{22}(1) & r_{21}(0) & r_{22}(0) \end{pmatrix} \begin{pmatrix} \alpha_{11}(1) & \alpha_{12}(1) \\ \alpha_{21}(1) & \alpha_{22}(1) \\ \alpha_{11}(2) & \alpha_{12}(2) \\ \alpha_{21}(2) & \alpha_{22}(2) \end{pmatrix} = \begin{pmatrix} r_{11}(1) & r_{12}(1) \\ r_{21}(1) & r_{22}(1) \\ r_{11}(2) & r_{12}(2) \\ r_{21}(2) & r_{22}(2) \end{pmatrix}. \quad (11)$$

Таким образом, от блочно-матричной формы (10) удалось перейти к двум обычным векторно-матричным формам представления систем линейных уравнений, для решения которых относительно коэффициентов предсказания $\alpha_{ij}(l)$ можно применить метод Холецкого в обычной векторно-матричной форме.

Для быстрого метода решения систем линейных уравнений с использованием разложения Холецко-

го симметричной матрицы количество умножений пропорционально третьей степени параметра предсказания – p^3 . Поэтому для практических применений параметр p нежелательно выбирать более трех или четырех. Для параметра m желательно иметь примерно такие же ограничения. При этом необходимо учитывать, что параметр m связан с величиной периода основного тона вокализованной речи и может рассматриваться как скользящее окно внутри периода основного тона.

Разностное кодирование с двумерным линейным предсказанием

При двумерном предсказании речевого сигнала $x(i, n)$ уравнения разностного кодирования могут быть записаны в следующем виде:

$$d(i, n) = x(i, n) - \bar{x}(i, n),$$

где $i = \overline{1, m}, n = \overline{0, p-1}$.

В этом случае в качестве математической модели предсказателя можно использовать двумерное разностное уравнение

$$\bar{x}(i, n) = \sum_{l=1}^{p_1-1} \sum_{k=1}^{p_2-1} \alpha(l, k) \bar{x}(i-1, n-k). \quad (12)$$

Несложно заметить, что модель (6) отличается от модели (12) лишь формой представления: модель (6) представлена в векторно-матричной форме, а модель (12) – в последовательностной форме.

Аналогичным образом отличаются корреляционные модели.

Для двумерного сигнала можно записать блочно-теплицевую автокорреляционную матрицу в виде

$$\hat{R}_{p,p_2} = \begin{pmatrix} R_{xx}(0) & R_{xx}(1) & \dots & R_{xx}(p_1-1) \\ R_{xx}(1) & R_{xx}(0) & \dots & R_{xx}(p_1-2) \\ \dots & \dots & \dots & \dots \\ R_{xx}(p_1-1) & R_{xx}(p_1-2) & \dots & \dots \end{pmatrix},$$

где

$$R_{xx}(k) = \begin{pmatrix} r_{xx}(k, 0) & r_{xx}(k, 1) & \dots & r_{xx}(k, p_2-1) \\ r_{xx}(k, 1) & r_{xx}(k, 0) & \dots & r_{xx}(k, p_2-2) \\ \dots & \dots & \dots & \dots \\ r_{xx}(k, p_2-1) & r_{xx}(k, p_2-2) & \dots & r_{xx}(k, 0) \end{pmatrix},$$

$r_{xx}(k, m)$ – коэффициенты двумерной корреляции.

Таким образом, математические методы разностного кодирования с двумерным линейным предсказанием являются эквивалентными математическим моделям многоканального векторно-разностного кодирования.

Быстрые алгоритмы вычисления оценок апериодических автокорреляционных функций

Методы скалярно- и векторно-разностного кодирования с линейным предсказанием базируются на

решении системы линейных уравнений, заданной через автокорреляционную матрицу, относительно коэффициентов предсказания. В свою очередь, автокорреляционная матрица строится на основе аперидической автокорреляционной функции (АКФ), заданной в последовательностной форме:

$$\hat{r}(k) = \frac{1}{N_1} \sum_{n=0}^{N_1-k-1} x(n) x(n+k), \quad (13)$$

где $k = \overline{0, N_1 - 1}$.

Выражение (13) характеризует смещенную оценку последовательности, которая может рассматриваться как стационарный в широком смысле случайный процесс. Согласно работе [7], использование смещенной оценки АКФ гарантирует положительную полуопределенность автокорреляционной матрицы, необходимую для разрешимости системы линейных уравнений. Заметим, что, пользуясь выражением (13), можно получить лишь первую половину значений АКФ. Вторая половина значений может быть получена за счет симметрии значений АКФ.

Быстрые алгоритмы вычисления оценки АКФ можно получить либо через векторно-матричные, либо через полиномиальные формы представления периодических АКФ, через которые в том числе можно выразить аперидические АКФ, заданные выражением (13).

Оценку периодической АКФ можно также задать в последовательностной форме

$$\hat{r}(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) x(n+k)_N, \quad (14)$$

где аргумент $n+k$ дискретной функции $x(n+k)$ вычисляется по $\text{mod } N$. Легко показать, что после дополнения в выражении (13) последовательности $\{x(n)\}_{n=0}^{N_1-1}$ нулями с помощью периодической АКФ можно вычислить оценки аперидической АКФ длины $2N_1 = N$. В векторно-матричной форме оценку периодической АКФ можно записать в виде

$$\hat{r}_N = \frac{1}{N} \hat{S}_N x_N, \quad (15)$$

где \hat{r}_N – вектор значений оценки АКФ; \hat{S}_N – матрица – правый циркулянт с образующим вектором-строкой $\hat{S}_N = x'_N = (x_0, \dots, x_{N-1})$, соответствующим последовательности $\{x(n)\}$. Для выражения (15) известен быстрый алгоритм вычисления АКФ через быстрое преобразование Фурье (БПФ):

$$\hat{r}_N = \frac{1}{N} (F_N^{(\Phi)})^{-1} D_N^* F_N^{(\Phi)} x_N, \quad (16)$$

где $F_N^{(\Phi)}$ – факторизованная форма матрицы преобразования Фурье в поле комплексных чисел, соответствующая тому или иному алгоритму БПФ; D_N^* – диагональная матрица, элементами которой являются компоненты вектора комплексных коэффициентов БПФ от вектора x_N , т. е.

$$D_N^* = \text{diag}\{d_0^*, d_1^*, \dots, d_{N-1}^*\},$$

$$d_N^* = d_0, d_2, \dots, d_{N-1}, d_N = F_N x_N.$$

Если ввести $\bar{P} = |\bar{X}_N|^2 = D_N^* F^{(\Phi)} x_N$ – вектор квадрата модуля коэффициентов БПФ от вектора x_N , то из выражения (16) можно получить известное соотношение для корреляционного метода оценки спектральной плотности мощности сигнала

$$\hat{P}_x(K) = \sum_{n=0}^{N-1} \hat{r}(m) \exp(-i2\pi km / N).$$

Поэтому предлагаемые далее быстрые алгоритмы вычислений оценок АКФ в равной степени могут быть пригодны также для цифрового спектрального анализа.

Алгоритмы вычисления АКФ на основе быстрых гиперкомплексных преобразований Фурье

В работах [7, 8] были предложены алгоритмы быстрых гиперкомплексных преобразований Фурье (ГПФ). Рассмотрим возможности применения данных алгоритмов для быстрого вычисления АКФ.

Матрица гиперкомплексных преобразований Фурье задается в виде

$$F_N^{(1)} = [j_1^{km}],$$

где $k, m = \overline{0, N-1}$, j_1 – мнимая единица, образующая мультипликативную циклическую группу из всех мнимых единиц. Данные ГПФ образованы в расширениях поля рациональных чисел, являющихся также полями.

Матрично-факторизованную форму представления алгоритмов БПФ по основанию два с прореживанием по частоте можно записать

$$F_N^{(\Gamma\Phi)} = \tilde{J}_N \text{diag} \left\{ \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \right\} \text{diag} \left\{ \begin{pmatrix} I_2 & I_2 \\ D_2 & -D_2 \end{pmatrix} \right\} \times \dots \times \begin{pmatrix} I_{N/2} & I_{N/2} \\ D_{N/2} & -D_{N/2} \end{pmatrix},$$

где $D_{N/2} = \text{diag}\{1, j_1, j_2, \dots, j_{N/2-1}\}$, $D_2 = \begin{pmatrix} 1 & 0 \\ 0 & j \end{pmatrix}$, $j = \sqrt{-1}$ – мнимая единица в поле комплексных чисел, j_k – k -я мнимая единица в гиперкомплексной алгебре с $(N/2 - 1)$ мнимыми единицами.

Блок-схема алгоритма вычисления N -точечной периодической АКФ через ГПФ представлена на рис. 4, где через $\hat{R}_{xx}(k)$ обозначена оценка АКФ в области ГПФ. Применяя к ней обратное ГПФ, получаем значения АКФ во временной области $r_{xx}(m)$.

Использование алгоритмов ГБПФ лучше всего продемонстрировать на примере. На рис. 5 и 6 приведены графы прямого и обратного ГБПФ, соот-

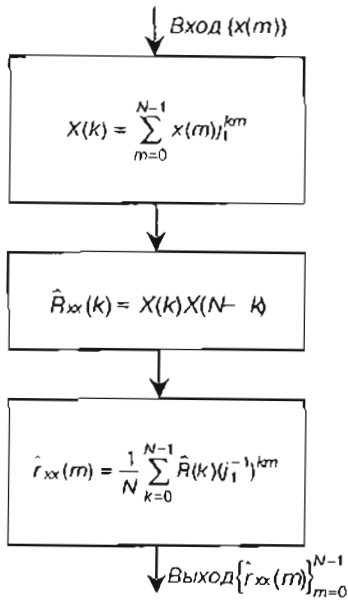


Рис. 4. Блок-схема алгоритма вычисления аperiodической АКФ через ГПФ

ветствующие вычислению аperiodической АКФ через восьмиточечную периодическую АКФ. Значения коэффициентов ГПФ представлены в гиперкомплексной четырехроторной алгебре – алгебре квантернионов $\{1, j, i, k\}$, $i = \sqrt{-1}$, $j = \sqrt{i}$, $k = \sqrt{-i}$.

Значения АКФ в области ГПФ:

$$\begin{aligned}
 R(1) = R(7) &= a + jb - kb, \\
 a &= x_0^2 + x_1^2 + x_2^2 + x_3^2, \\
 b &= x_0x_3 + x_1x_2 + x_0x_1 + x_2x_3 = \\
 &= x_0(x_1 - x_3) + x_2(x_1 + x_3); \\
 R(3) = R(5) &= a - jb + kb, \\
 R(2) = R(6) &= (x_0 - x_2)^2 + (x_1 - x_3)^2; \\
 R(4) &= (x_0 + x_2 - x_1 - x_3)^2; \\
 R(0) &= (x_0 + x_1 + x_2 + x_3)^2.
 \end{aligned}$$

Таким образом, алгоритм вычисления АКФ на основе ГБПФ для $N = 8$; количества арифметических операций в поле рациональных чисел; число умножений $M = 10$. число сложений $A = 18$.

Заметим, что алгоритм вычисления аperiodических АКФ через быстрое преобразование Хартли (БПХ) может быть получен посредством подстановки в выражение (16) матрицы

$$F_N = P_N H_N,$$

где $P_N = \frac{1}{N} F_N H_N$ – переходная матрица от матрицы H_N к матрице F_N .

В результате получим

$$\hat{R}_N = \frac{1}{N} H_N^{(\Phi)} [D_N^{(h)}] H_N^{(\Phi)} X_N = \frac{1}{N} \bar{J}_N H_N^{(\Phi)} D_N^{(h)} H_N^{(\Phi)} X_N,$$

где $[D_N^{(h)}] = P_N^{-1} D_N^{(h)} P_N$;

$$D_N^{(h)} = d_0^{(h)} \oplus 1/2 \left[\begin{matrix} \oplus_{k=1}^{N-1} (d_{N-k}^{(h)} + d_k^{(h)}) \\ \oplus_{k=1}^{N-1} (d_{N-k}^{(h)} - d_k^{(h)}) \end{matrix} \right] \bar{I}_{N-1},$$

$\bar{d}_N^{(h)} = H_N X_N$, $d_k^{(h)}$ – k -й компонент вектора $\bar{d}_N^{(h)}$, $\bar{J}_k = 1 \oplus \bar{I}_{N-1}$, \bar{I}_{N-1} – матрица инверсной перестановки

Выводы факторизованных форм $H_N^{(\Phi)}$ матриц H_N приведены в работе [9].

Аппроксимация нормированных аperiodических АКФ функциями среднего значения разностей

Функции среднего значения разностей (ФСР) было предложено [5] использовать для определения величины периода основного тона вокализированной речи. В отличие от модифицированных АКФ они не имеют операций умножения вещественных чисел. В работе [3] был предложен метод прореженных весовых функций, применение которых к ФСР позволяет снизить количество арифметических операций в несколько раз без заметного снижения точности величины периода основного тона.

Покажем, каким образом можно использовать ФСР для аппроксимации нормированных аperiodических АКФ, через которые строятся автокорреляционные матрицы.

В общем виде для оценки ФСР можно записать выражение

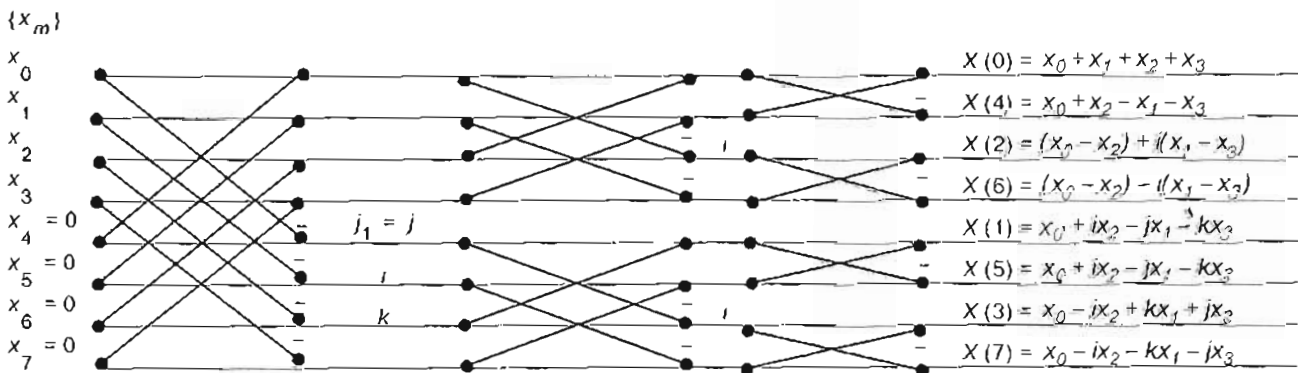


Рис. 5. Граф обратного ГБПФ

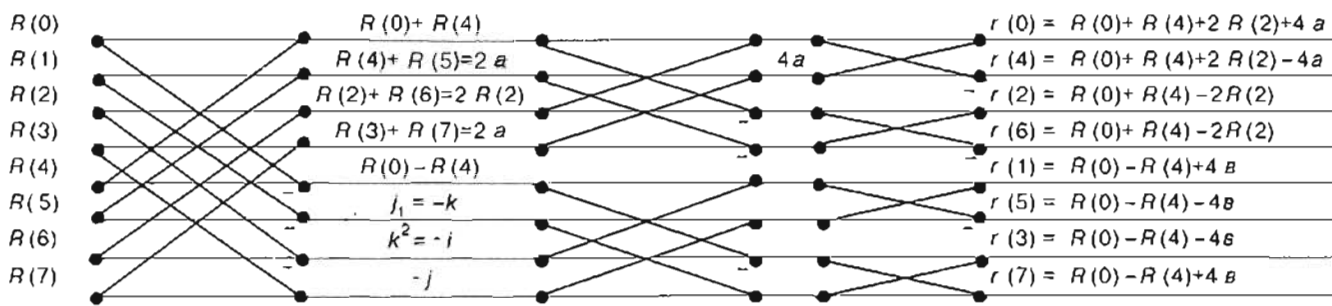


Рис. 6. Граф прямого ГБПФ

$$\hat{\gamma}(k) = \frac{1}{N} \sum_{m=0}^{N-1} |x(m)w_1(m) - x(m \pm k)w_2(m \pm k)|, \quad k = \overline{0, N_i - 1} \quad (17)$$

Применительно к определению величины периода основного тона параметр N_i много больше величины N .

Рассмотрим возможность применения оценки $\hat{\gamma}(k)$ для аппроксимации нормированных смещенных оценок аперiodических АКФ.

Для упрощения выкладок в дальнейшем будем использовать прямоугольные весовые функции с амплитудой, равной единице, так что выражение (17) приобретает вид

$$\hat{\gamma}(k) = \frac{1}{N} \sum_{m=0}^{N-1} |x(m) - x(m+k)|, \quad (18)$$

где $m, k = \overline{0, N-1}, m+k = \overline{(m+k)}_N$.

Функцию (18) по аналогии с АКФ можно рассматривать как периодическую, если параметр $(m+k)$ вычисляется по mod N .

Периодическая функция $\hat{\gamma}(k)$, так же, как и периодическая АКФ, обладает симметрией, т. е. $\hat{\gamma}(k) = \hat{\gamma}(N-k), k = \overline{1, N-1}$.

Введем дискретную функцию, образованную через дополнения $\tilde{\gamma}(k) = \gamma_{\max} - \hat{\gamma}(k)$. Полученная функция $\tilde{\gamma}(k)$ будет иметь максимальное значение при $k=0$ (при нулевом сдвиге окна, так же, как АКФ).

Если допустить, что с помощью N -точечной периодической ФСР вычисляется $N/2$ точечная аперiodическая ФСР, то исходная последовательность отсчетов цифрового сигнала $\{x_i\}_{i=0}^{N/2-1}$ должна быть дополнена $N/2$ нулями (так же, как АКФ). Тогда минимальное значение периодической ФСР равно $\tilde{\gamma}_{\min} = \hat{\gamma}(N/2) = 0$. Пронормировав значения $\hat{\gamma}(N/2)$, получим соотношения

$$\tilde{\gamma}_{\min} = \hat{\gamma}(N/2) = \hat{R}(N/2) = 0, \quad \tilde{\gamma}_{\max} = \hat{\gamma}(0) = \hat{R}_{\max} = \hat{R}(0).$$

Поведение функций $\tilde{\gamma}(k)$ и $\hat{R}(k)$ в промежуточных точках между значениями $\tilde{\gamma}_{\max}$ и $\tilde{\gamma}_{\min}$, можно оце-

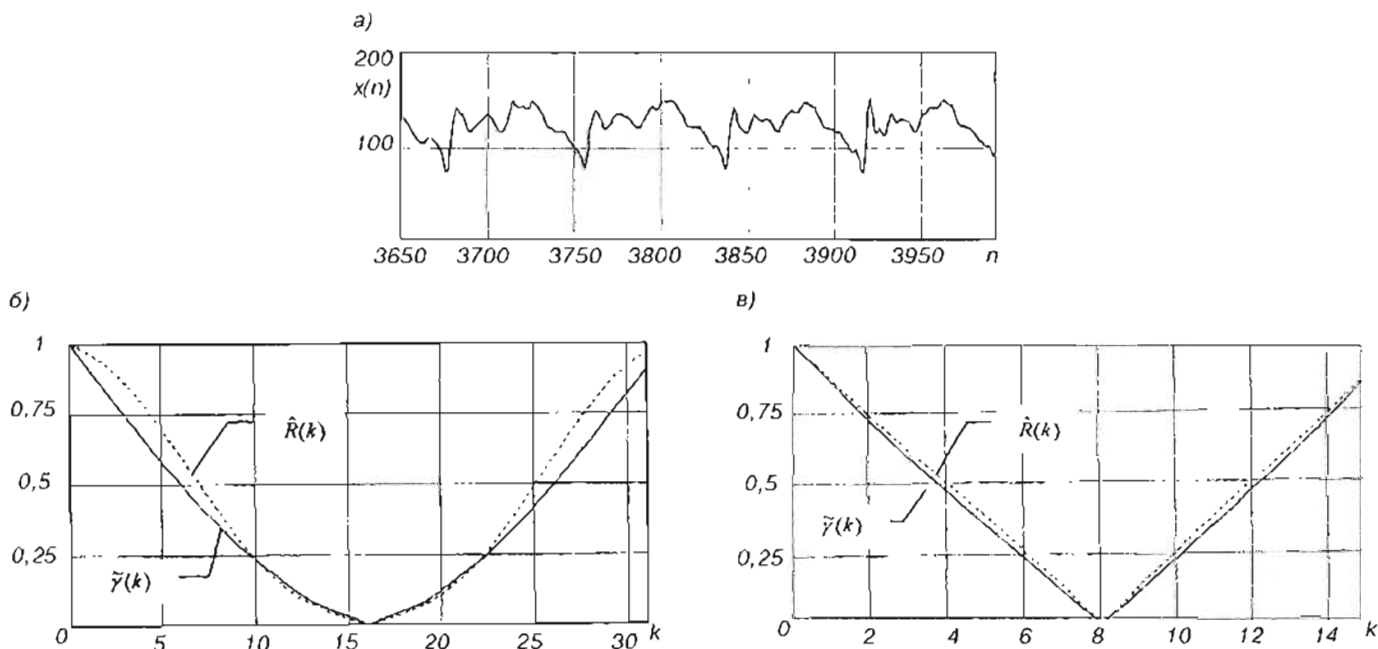


Рис. 7. Графики нормированных АКФ и $\tilde{\gamma}(k)$: а – фрагмент локализованного речевого сигнала; б – при $N = 16$; в – при $N = 32$;

■ **Оценки количества арифметических операций быстрых алгоритмов**

N- длина дополненных N/2 нулями последовательностей $\{x_i\}_{i=0}^{N-1}$	Количество нетривиальных умножений			Количество сложений-вычитаний		
	АКФ через БПХ	ФСР	ГБПФ	АКФ через БПХ	ФСР	ГБПФ
4	4	-		12	4	
8	12	-	10	44	16	18
16	48	-	24	128	64	39
32	152	-	68	348	256	110
64	427	-	220	884	1024	385
128	1096	-	780	2146	4096	1441
256	2696	-	1924	5042	8384	4608

нить с помощью функции $\Delta(k)$, отражающей погрешность отклонения

$$\Delta(k) = \frac{\hat{R}(k)}{\hat{R}_{\max}} - \frac{\tilde{\gamma}(k)}{\tilde{\gamma}_{\max}}$$

где

$$\hat{R}_{\max} = R(0) = \frac{1}{N} \sum_{i=0}^{N-1} x_i^2,$$

$$\tilde{\gamma}_{\max} = \tilde{\gamma}(0) = \frac{1}{N} \sum_{i=0}^{N-1} |x_i|.$$

Для практического применения интерес представляет возможность использования дополненных нормированных ФСР для вычисления N/2 оптимальных коэффициентов предсказания из системы линейных уравнений, заданных через теплицеву матрицу, образующей строкой которой может быть использованы N/2 значений ФСР вместо N/2 значений АКФ.

В таблице представлены сравнительные оценки количества арифметических операций для вычисления аперiodических АКФ через БПХ, ГБПФ и через N-точечные периодические ФСР. По таблице можно заключить, что для длин периодических АКФ и ФСР от N = 4 до N = 64 в алгоритмах вычисления АКФ через БПХ общее количество арифметических операций сложений и умножений больше количества сложений-вычитаний, необходимых для вычислений ФСР, которое может быть отражено выражением $A_{\text{ФСР}} = N^2 / 4$.

Литература

1. Брауде-Золотарев Ю. Сжатие речи. – Компьютерра. – 1998. – № 15 (293). – С. 28–31.
2. Spanias A. Speech codig: A Tutorial Review ASU — T.R.C. Technical report. T.R.C. – С.Р. – ASP – 9201. – July 1992. – P. 21–23.
3. Гагарин К. Ю. Быстрые алгоритмы векторно-разностного кодирования речевых сигналов. Сб. научн. тр. СПбГТУ, сер. ВТАРЭ. – № 457. – СПб., 1995. – С. 84–89.
4. Гагарин Ю. И., Раххал М. Математические модели предсказателей в схемах векторно-разностного кодирования речевых сигналов на основе многоканальных линейных систем. Сб. научн. тр. – сер. ВТАРЭ. – № 487. – СПбГПУ, 2002. – С. 80–81.
5. Рабинер Л. В., Шафер Р. В. Цифровая обработка речевых сигналов. – М.: Радио и связь, 1981. – 495 с.

В заключение следует отметить, что опубликованные в данной работе результаты были получены при финансовой поддержке Минобразования РФ НИР-грант ТО2-03.2-2731

На рис. 7 приведены графики функций $\hat{R}(k)$ и $\tilde{\gamma}(k)$, из которых можно заключить, что характер изменения значений $\hat{R}(k)$ и $\tilde{\gamma}(k)$ является одинаковым, хотя в отдельных случаях наблюдается отклонение значений $\hat{R}(k)$ и $\tilde{\gamma}(k)$ до 30 %. [10]

Выводы

1. Полученные математические модели ВР-кодирования речевых сигналов во временной области позволили использовать для нахождения оптимальных коэффициентов многоканального (двумерного) линейного предсказания быстрые алгоритмы решения систем линейных уравнений одномерного предсказания.

2. Для вычисления оценок АКФ предложены быстрые алгоритмы через гиперкомплексные преобразования Фурье и через дополнения функций среднего значения разностей. Приведенные сравнительные данные по количеству арифметических операций позволяют заключить, что для длин АКФ, не превышающих 128 точек, предлагаемые алгоритмы являются наиболее предпочтительными. Кроме того, все арифметические операции в быстрых алгоритмах выполняются в поле рациональных чисел.

6. Марпл С. Л. Цифровой спектральный анализ и его приложения. – М., Мир, 1990. – 584 с.
7. Гагарин Ю. И., Гагарин К. Ю. Гиперкомплексные быстрые преобразования Фурье в расширениях поля рациональных чисел // Труды СПбГТУ, серия ВТАРЭ. – 1998. – № 472. – С. 77–80.
8. Гагарин К. Ю. Быстрые гиперкомплексные преобразования Фурье в расширениях поля рациональных чисел с длиной преобразования, факторизованной взаимно простыми множителями // Труды СПбГТУ, серия ВТАРЭ. – 2000. – № 480. – С. 89–92.
9. Гагарин Ю. И. Математические модели и алгоритмы быстрых ортогональных преобразований. – СПбГТУ, 1999 – 400 с.
10. Раххал М. Методы и алгоритмы векторно-разностного кодирования цифровых аудиосигналов. – СПбГПУ, кандидатская диссертация, 2004. – 126 с.

УДК 621.396

ОБОСНОВАНИЕ МАСШТАБА ЦИФРОВЫХ КАРТ МЕСТНОСТИ, ИСПОЛЬЗУЕМЫХ ПРИ РАСЧЕТЕ НАПРЯЖЕННОСТИ ПОЛЯ РАДИОСИГНАЛОВ

И. А. Зикратов,

канд. техн. наук, доцент

К. В. Степаненко,

адъюнкт

Военный университет противовоздушной обороны (филиал, г. Санкт-Петербург)

Предложен подход, основанный на теории случайных полей, позволяющий обосновать полноту представления данных о подстилающей поверхности типа «шероховатый рельеф» в цифровых картах местности, используемых в специализированных геоинформационных системах проектирования радиолиний. Приведены примеры корреляционной связи результатов расчета поля рассеяния с параметрами цифровой модели местности.

The approach based on the theory of casual field is proposed. It allows to prove completeness of data presentation about a spreading surface such as « rough relief » on the digital landscape maps used in specialized geoinformation systems of radiolines designing. The examples of correlation connection of calculation results of dispersion field with parameters of digital landscape model are performed.

Интенсивный рост количества радиотехнических средств (РТС) сбора, обработки и передачи информации с использованием радиоволн различных частотных диапазонов приводит к необходимости их рационального размещения на ограниченных площадях. Основу этой задачи составляет расчет дальности действия радио и радиолокационных средств, который осуществляют с применением различных методик [1].

При определении дальности действия РТС наибольшую трудность вызывает учет влияния неровной подстилающей поверхности на величину комплексной амплитуды напряженности электромагнитного поля в точках наблюдения, находящихся под малыми углами места. Известно [2], что исследование процессов рассеивания радиоволн на земной поверхности, отличающейся многообразием форм, привело к необходимости разработки ряда математических (электродинамических) моделей подстилающей поверхности (ПП) и соответствующих им методов теории дифракции и распространения радиоволн. Повышение степени достоверности результатов расчета можно осуществлять, во-первых, путем использования точного или асимптотического математического аппарата теории дифракции и распространения радиоволн, анализ и рассмотрение которого выходит за рамки данной статьи, и, во-вторых,

за счет обоснованной степени детализации данных о свойствах ПП, оказывающих существенное влияние на рассеяние радиоволн. К основным таким свойствам относятся геометрия (рельеф) земной поверхности и величина комплексной диэлектрической проницаемости, которая определяет радиофизические свойства ПП.

Возможность для реализации указанных путей повышения точности прогнозирования напряженности поля радиосигналов РТС появилась с широким внедрением геоинформационных систем (ГИС), основанных на использовании данных о земной поверхности, представленных в цифровых картах местности (ЦКМ), и объектно-ориентированных методах их обработки, которые привели к попыткам автоматизации расчета дальности действия разработчиками радиолиний [3, 4].

В геоинформационных системах моделирование ПП осуществляется на основе цифровой картографической информации (ЦКИ), которая может формироваться по материалам аэрофотосъемки, геодезической съемки, данным GPS, топографических карт и т. д. [4] и ориентирована в основном на описание геометрии земной поверхности. В зависимости от картографической технологии изготовления ЦКМ, цифровая картографическая информация отличается масштабом и точностью представления

данных о геометрии ПП. Данные о радиофизических свойствах ПП чаще всего представляют собой сведения о характере покрытия (лес, тип водоема и т. п.), без описания их радиофизических свойств, что существенно затрудняет учет влияния комплексной диэлектрической проницаемости на величину коэффициента отражения радиоволн.

В статье предлагается подход, позволяющий получить рекомендации по выбору такого масштаба цифровой картографической информации о ПП типа «шероховатый рельеф», использование которого обеспечит снижение погрешности при расчете дальности РТС, обусловленной неполнотой исходных данных. Под поверхностью типа «шероховатый рельеф» понимается геометрия участка местности, представляющего собой совокупность статистически однородных неровностей, характерные размеры которых превышают длину волны электромагнитного колебания (крупномасштабные неровности).

Приняты два ограничения в постановке и решении задачи:

1. Рассматривается скалярное волновое поле. Очевидно, что применительно к случаю электромагнитных колебаний необходимо учитывать поляризацию излученных и рассеянных волн. Особенности решения задачи для «электромагнитного» случая авторами исследуются и будут описаны в дальнейшем.

2. В настоящей работе не рассматривается влияние мелкомасштабных неровностей, размеры которых сопоставимы с длиной волны, так как существующие в настоящее время ЦКМ не обладают данными об их характерных размерах. Известно, что влияние таких неровностей проявляется в появлении диффузной составляющей в отраженном поле. В практических расчетах влияние подобных шероховатостей учитывают введением соответствующего множителя [3].

С учетом стохастической изменчивости свойств земной поверхности и влияния на процессы рассеяния радиоволн ряда неконтролируемых факторов

(учет которых в ЦКМ не представляется возможным), оценку пригодности цифрового картографического материала целесообразно получить, на взгляд авторов, с использованием статистических методов расчета полей рассеяния. Наиболее известным среди них является метод Кирхгофа (касательной плоскости) [5]. В этом случае в качестве показателя результата прогнозирования пространственных возможностей РТС можно говорить о мере корреляционной связи рассчитанного и истинного значений комплексной амплитуды напряженности электромагнитного поля в заданной точке наблюдения.

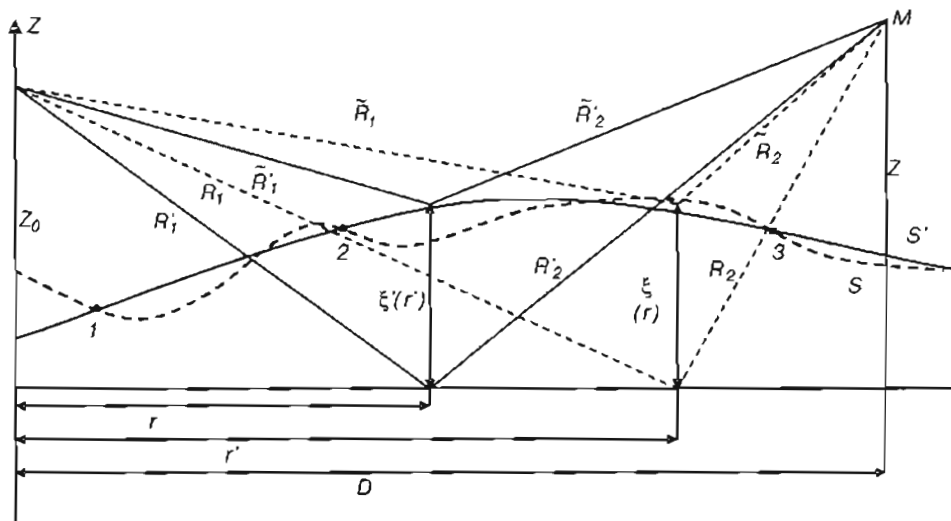
Метод Кирхгофа используют при рассмотрении рассеяния волн в том случае, когда характерные размеры неровностей на поверхности значительно превышают длину волны дифрагирующего поля и выполняется условие [5]

$$\sin(\psi) \gg \frac{1}{\sqrt{ka}}, \quad (1)$$

где $k = \frac{2\pi}{\lambda}$ – волновое число; a – характерный радиус кривизны неровностей; ψ – угол скольжения вектора падающей волны.

Пусть рассматриваемая реальная поверхность S представлена в ЦКМ с некоторой, присущей используемому масштабу картографического материала, точностью. По метрическим данным ЦКМ, в которой содержится информация о высотах рельефа местности, построена модель S' реальной поверхности S . При построении модели могут использоваться различные методы, предложенные, например, в работах [6–8].

Определим, насколько рассчитанное значение комплексной амплитуды поля $U(\mathbf{R})$ в произвольной точке наблюдения M с координатами \mathbf{R} , рассеянного на поверхности S' , будет соответствовать значению комплексной амплитуды поля в этой же точке наблюдения $U(\mathbf{R})$ при рассеянии на «эталонной»



■ Рис. 1. К постановке задачи

поверхности S . Эта задача сводится к отысканию связи между взаимной корреляцией неровностей эталонной поверхности S и ее модели S' и пространственной взаимной корреляции полей в выбранной точке наблюдения при их рассеянии на соответствующих поверхностях (рис. 1).

В радиолокации и значительной части линий связи используются длины волн, для которых в среднепересеченной местности выполняется неравенство (1) Тогда поле точечного источника в точке наблюдения M с координатами \mathbf{R} в приближении метода Кирхгофа можно представить в виде

$$U(\mathbf{R}) = U_0(\mathbf{R}) + \frac{1}{4\pi} \int_S V(r) \frac{\partial}{\partial n} \left[\frac{e^{ik(\hat{R}_1 + \hat{R}_2)}}{\hat{R}_1 \hat{R}_2} \right] dr, \quad (2)$$

где $U_0(\mathbf{R})$ – поле прямой волны; $V(r)$ – коэффициент отражения в точке r поверхности S ; \hat{R}_1 и \hat{R}_2 – расстояния от источника и точки наблюдения до текущей точки поверхности S , по которой производится интегрирование; n – нормаль к текущей точке поверхности.

Второе слагаемое в равенстве (2) описывает поле рассеяния на поверхности S .

Разложим \hat{R}_1 и \hat{R}_2 через вектор r , лежащий в плоскости $z = 0$, и значение высоты поверхности в этой точке $\zeta(r)$ при условии выполнения неравенств

$$k\hat{R}_1 \gg 1; k\hat{R}_2 \gg 1; \quad (3)$$

$$\sin^2 \psi \frac{k\sigma^2}{R_1} \ll 1; \sin^2 \chi \frac{k\sigma^2}{R_2} \ll 1, \quad (4)$$

где ψ и χ – углы скольжения падающей и отраженной волн; σ^2 – среднеквадратичная высота неровностей;

$$R_1 = \sqrt{r^2 + z_0^2}, R_2 = \sqrt{(D-r)^2 + z^2}.$$

Неравенства (3) справедливы при нахождении точек излучения и наблюдения в волновой зоне. Неравенства (4) выполняются при среднеквадратической высоте характерных неровностей, значительно меньшей размера первой зоны Френеля.

Замена производной по нормали в уравнении (2) равенством $\frac{\partial}{\partial n} = (\mathbf{n}\nabla)$ приводит выражение (2) к виду

$$U(\mathbf{R}) = \frac{1}{4\pi} \times \int_{S_0} \frac{\exp\{i[k(R_1 + R_2) - q_z \zeta(r)]\}}{R_1 R_2} V(r) \frac{(\mathbf{n}\mathbf{q})}{n_z} dr. \quad (5)$$

Здесь и далее под $U(\mathbf{R})$ понимается не полное поле в точке наблюдения, а поле рассеяния. Использование равенства $dS = \frac{dx dy}{n_z}$, где n_z – проекция вектора нормали на ось Z , позволяет перейти от интегрирования по поверхности S к интегрированию по

ее проекции S_0 на плоскость $z = 0$. При этом вектор излучения $\mathbf{q} = -k\nabla(R_1 + R_2)$ и все величины под знаком интеграла отнесены к поверхности S_0 .

Для принятых условий задачи поле $U(\mathbf{R})$ по аналогии с полем рассеяния на поверхности S описывается выражением

$$U(\mathbf{R}) = \frac{1}{4\pi} \times \int_{S_0} \frac{\exp\{i[k(R_1' + R_2') - q_z' \zeta'(r')]\}}{R_1' R_2'} V(r') \frac{(\mathbf{n}'\mathbf{q}')}{n_z'} dr'. \quad (6)$$

Интегрирование в выражении (6) также производится по поверхности S_0 , а физический смысл величин со штрихом понятен из рис. 1.

Учитывая, что вектор нормали к текущей точке поверхности S имеет поперечную и продольную компоненты

$$\mathbf{n} = (n_z, \mathbf{n}_\perp) = \left(\frac{1}{\sqrt{1+\gamma^2}}, \frac{-\gamma}{\sqrt{1+\gamma^2}} \right).$$

множитель в равенстве (5) с помощью несложных преобразований приводится к виду

$$\frac{(\mathbf{n}\mathbf{q})}{n_z} = \frac{q_z - \gamma \mathbf{q}_\perp}{\sqrt{1+\gamma^2}} \sqrt{1+\gamma^2} = q_z - \gamma \mathbf{q}_\perp, \quad (7)$$

где $\gamma = \nabla \zeta(r)$; ∇ – оператор дифференцирования. При больших значениях параметра Релея $(q_z \sigma)^2 \gg 1$, когда наибольший вклад в интеграл (5) вносят лишь точки стационарной фазы, можно полагать $\gamma = -\frac{\mathbf{q}_\perp}{q_z}$. Тогда из равенства (7) получаем окончательный результат:

$$\frac{(\mathbf{n}\mathbf{q})}{n_z} = \frac{q^2}{q_z}. \quad (8)$$

Следуя аналогичным рассуждениям, для сомножителя в подынтегральном выражении (6) получим

$$\frac{(\mathbf{n}'\mathbf{q}')}{n_z'} = \frac{q'^2}{q_z'}. \quad (9)$$

Как известно, взаимная корреляционная функция случайного поля определяется равенством

$$\overline{\tilde{U}(\mathbf{R}) \tilde{U}^*(\mathbf{R})} = \langle U(\mathbf{R}) U(\mathbf{R}) \rangle - \langle U(\mathbf{R}) \rangle \langle U(\mathbf{R}) \rangle, \quad (10)$$

где $\tilde{U}(\mathbf{R})$ и $\tilde{U}^*(\mathbf{R})$ – флуктуационные составляющие полей рассеяния $U(\mathbf{R})$ и $U(\mathbf{R})$. Тогда, после подстановки (8) и (9) соответственно в интегралы (5) и (6) и использования формулы (10), выражение для взаимной корреляционной функции полей рассеяния примет вид:

$$\overline{\tilde{U}(\mathbf{R}) \tilde{U}^*(\mathbf{R})} = \frac{1}{16\pi^2} \iint_{S_0} \frac{V(r) V(r') q^2 q'^2}{R_1 R_2 R_1' R_2' q_z q_z'} e^{ik(R_1 + R_2 - R_1' - R_2')} \times W(q_z, -q_z', \mathbf{r} - \mathbf{r}') dr dr'. \quad (11)$$

Здесь

$$W(q_z, -q'_z, \mathbf{r} - \mathbf{r}') = f_2(q_z, -q'_z, \mathbf{r} - \mathbf{r}') - f_1(q_z) f_1(-q'_z), \quad (12)$$

где $f_2(q_z, -q'_z, \mathbf{r} - \mathbf{r}') = e^{-i(q_z \xi - q'_z \xi')}$ и $f_1(q_z) f_1(-q'_z) = e^{-i q_z \xi} \cdot e^{-i q'_z \xi'}$ – соответственно двумерная и произведение одномерных характеристических функций высот неровностей поверхности и ее модели. Формула (11) аналогична выражению для пространственной корреляционной функции флуктуаций поля рассеяния на статистически неровной поверхности, представленному в работе [5], однако физический смысл переменных находящихся под знаком интеграла, иной. Отличие обусловлено постановкой задачи, в которой рассматриваются результаты рассеяния волн на разных поверхностях, а именно, на эталонной ПП и ее модели, неровности которых взаимно коррелированы, применительно к одной точке наблюдения.

Введем в формулу (11) новую переменную интегрирования $\xi = \mathbf{r} - \mathbf{r}'$ и разложим показатель экспоненты в ряд по степеням ξ , ограничившись линейными членами.

$$\begin{aligned} R_1 + R_2 - R'_1 - R'_2 &= \\ &= \sqrt{r^2 + z_0^2} + \sqrt{(D - r)^2 + z^2} - \\ &- \sqrt{(r - \xi)^2 + z_0^2} - \sqrt{[D - (r - \xi)]^2 + z^2} = \\ &= \sqrt{(D - r)^2 + z^2} - \sqrt{D^2 - 2Dr + r^2 + z^2} + \\ &+ \left[\frac{r}{\sqrt{r^2 + z_0^2}} - \frac{\sqrt{D^2 - 2Dr + r^2 + z^2} (D - r)}{(D - r)^2 + z^2} \right] \xi \end{aligned}$$

откуда для показателя экспоненты окончательно получим:

$$k(R_1 + R_2 - R'_1 - R'_2) = k \left[\frac{r}{R_1} - \frac{(D - r)}{R_2} \right] \xi = -\mathbf{q}_\perp \xi, \quad (13)$$

Такое разложение будет справедливо при выполнении неравенств

$$\sin^2 \psi \frac{k \xi_x^2}{R} \ll 1; \quad \frac{k \xi_y}{R} \ll 1, \quad (14)$$

где $R \equiv R_1 \equiv R_2$, а ξ_x и ξ_y размеры области, существенной для интегрирования по $d\xi$. Эти размеры обусловлены расстояниями, на которых существенно убывает функция $W(q_z, q'_z, \xi)$.

В результате выражение для взаимной корреляционной функции полей рассеяния в одной и той же точке наблюдения \mathbf{R} примет вид:

$$\begin{aligned} \overline{\tilde{U}(\mathbf{R}) \tilde{U}^*(\mathbf{R})} &= \frac{1}{16\pi^2} \frac{V(\mathbf{r})V(\mathbf{r}')q^2q'^2}{R_1 R_2 R'_1 R'_2 q_z q'_z} \times \\ &\times \int_{S_0} e^{-i\mathbf{q}_\perp \xi} W(q_z, q'_z, \xi) d\xi \end{aligned} \quad (15)$$

Так как локальный коэффициент отражения Френеля $V(\mathbf{r})$ является медленно меняющейся функцией расстояния, то множитель $\frac{V(\mathbf{r})V(\mathbf{r}')q^2q'^2}{R_1 R_2 R'_1 R'_2 q_z q'_z}$ в выражении (15) вынесен за знак интеграла.

Используя полученные выражения, можно оценить требования к полноте цифровой картографической информации. Из выражения (15) следует, что связь параметров поля, рассеянного на реальной поверхности, с результатами расчета этих параметров в заданной точке наблюдения находится в зависимости от функции $W(q_z, q'_z, \xi)$, описывающей корреляцию неровностей эталонной поверхности S и ее модели S' , и условий наблюдения. Условия наблюдения (длина волны, координаты точек излучения и наблюдения) характеризуются вектором излучения \mathbf{q} .

Оценив расстояние ξ , на котором существенно убывает функция $W(q_z, q'_z, \xi)$, а следовательно, и $\tilde{U}(\mathbf{R}) \tilde{U}^*(\mathbf{R})$, можно обосновать минимально допустимую погрешность представления поверхности S' . Эта погрешность зависит от выбранной степени детализации исходных данных, зависящей, в свою очередь, от масштаба картографического материала ЦКМ.

Решить эту задачу можно, только зная конкретный вид закона распределения случайных неровностей. Вид закона распределения высот неровностей или их углов наклона устанавливают, либо проводя соответствующий статистический анализ топографических карт конкретного участка местности [9], либо «подбирая» искусственно сконструированные функции автокорреляции [3].

Считая, для примера, распределение высот неровностей поверхностей S и S' гауссовым, с нулевыми средними и среднеквадратическими отклонениями σ и σ' , запишем характеристические функции в виде

$$f_2(q_z, -q'_z, \mathbf{r} - \mathbf{r}') = e^{q_z q'_z \sigma \sigma' w_{SS'}(\xi) - \frac{1}{2}(q_z^2 \sigma^2 + q_z'^2 \sigma'^2)}, \quad (16)$$

$$f_1(q_z) = e^{-\frac{q_z^2 \sigma^2}{2}}; \quad f_1(q'_z) = e^{-\frac{q_z'^2 \sigma'^2}{2}}, \quad (17)$$

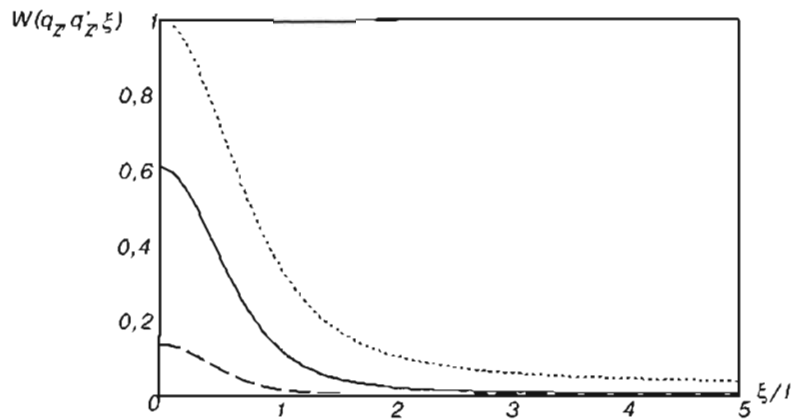
где $w_{SS'}(\xi)$ – взаимная корреляционная функция поверхностей S и S' .

Процесс построения поверхности S' по данным ЦКИ можно интерпретировать процедурой текущего сглаживания случайной поверхности S , параметром которой является масштаб расстояний Δ :

$$S'(x) = \frac{1}{2\Delta} \int_{x-\Delta}^{x+\Delta} S(u) du.$$

В этом случае взаимную корреляционную функцию случайных стационарных поверхностей S и S' можно найти из известного выражения [10]

$$w_{SS'}(\xi) = \int_0^\infty h(u) w_S(\xi - u) du, \quad (18)$$



■ Рис. 2. Зависимость корреляционной связи величины поля рассеяния с результатами ее расчета от параметров подстилающей поверхности и условий наблюдения:

- $\delta_{\rho\rho'} = 0 \%$;
- $\delta_{\rho\rho'} = 33 \%$;
- $\delta_{\rho\rho'} = 50 \%$

где $w_S(\xi - u)$ – корреляционная функция исходной поверхности; $h(u)$ – прямоугольная импульсная характеристика линейной системы, осуществляющей текущее сглаживание:

$$h(u) = \begin{cases} 1/2\Delta, & |u| < \Delta \\ 0, & |u| > \Delta \end{cases} \quad (19)$$

Пусть поверхность S имеет гауссову корреляцию неровностей:

$$w_S(\xi) = e^{-\frac{\xi^2}{l^2}}, \quad (20)$$

где l – радиус пространственной корреляции неровностей поверхности (средний размер неровностей). Подставляя (19) и (20) в (18) и учитывая, что масштаб расстояний Δ имеет смысл введенной ранее переменной ξ , получим выражение для взаимной корреляционной функции поверхностей S и S' :

$$w_{SS'}(\xi) = \frac{l\sqrt{\pi}}{2\xi} \operatorname{erf}\left(\frac{\xi}{l}\right). \quad (21)$$

Используя представление функции ошибок в виде:

$$\operatorname{erf} x = \frac{2}{\sqrt{\pi}} \int_0^x e^{-x^2} dx = \frac{2}{\sqrt{\pi}} \left(x - \frac{x^3}{3} \pm \dots \right)$$

и подставляя его в (21) и (16), получим оценку масштаба ξ , при котором существует корреляционная связь между рассеянным полем поверхности S со статистическими параметрами σ , l и результатами его расчета, зависящими от параметра σ' модели поверхности S' при заданных условиях радиолокационного наблюдения q_z , q'_z :

$$\xi \cong \pm \frac{\sqrt{3}}{\rho\rho'} \sqrt{\rho\rho'(\rho\rho' - 1)} l, \quad (22)$$

где $\rho = q_z\sigma$; $\rho' = q'_z\sigma'$.

На рис. 2 представлены графики функции $W(q_z, q'_z, \xi)$ при различных значениях статистических параметров поверхностей S и S' и условиях радиолокационного наблюдения.

На рисунке величиной $\delta_{\rho\rho'}$ обозначена выраженная в процентном соотношении разность величин $\rho = q_z\sigma$ и $\rho' = q'_z\sigma'$.

Из анализа представленных зависимостей можно сделать следующие рекомендации по выбору масштаба ЦКМ, имеющие четкую физическую интерпретацию.

Выбираемый масштаб расстояний ξ должен иметь порядок среднего размера неровностей, находящихся в пределах зоны Френеля. Уменьшение масштаба ($\xi \gg l$) приводит к возрастанию погрешности вычислений. Это связано с тем, что существенное влияние на величину взаимной корреляции полей $U(\mathbf{R})$ и $U'(\mathbf{R})$ оказывает значение погрешности $\delta_{\rho\rho'}$ определения вектора излучения \mathbf{q} , от которого зависят фазовые соотношения между прямой и отраженными волнами и, следовательно, интерференционная картина в точке наблюдения. Следовательно, стремление повысить степень достоверности определения дальности РТС под малыми углами места точки наблюдения однозначно приводит к необходимости повышения точности определения координат «зеркальных точек» (в приближении Кирхгофа), от которых зависит величина z -компоненты вектора излучения q_z , что достигается путем использования в пределах первой зоны Френеля картографической информации заведомо более крупного масштаба ($\xi \leq l$), а при отсутствии таковой – дополнительной топографической обработкой позиций.

Например, получившие наиболее широкое распространение ЦКМ, изготовленные по технологии оцифровки бумажных топографических карт, в настоящее время имеют наиболее крупный масштаб 1:100 000. Отсюда следует вывод, что данные таких ЦКМ способны обеспечить высокую степень достоверности расчета дальности действия РТС, рабо-

тающих в диапазоне коротких волн ($\lambda = 100$ м) и на более длинных волнах. Для более высоких частот расчет, основанный на картографическом материале указанного масштаба, не обеспечит построение интерференционной картины в пространстве, и может быть сведен к определению участков видимости на трассе распространения радиоволн посредством элементарных геометрических выкладок.

При больших расстояниях до точек наблюдения и излучения, когда слагаемые (17) в выражении можно считать равными, выражение (16) упрощается. В этом случае подынтегральное выражение в формуле (15) принимает смысл преобразования Фурье от характеристической функции высот неровностей, которое представляет собой энергетический спектр случайной поверхности. Следовательно, вычисляя энергетический спектр случайной поверхности, можно получать количественную оценку взаимной корреляции параметров поля рассеяния и их оценок, получаемых при расчете.

Таким образом, предложенный подход позволяет обосновать требования к масштабу и полноте представления исходных данных ЦКМ, необходимых для расчета дальности действия РТС при наличии на ПП крупномасштабных неровностей. Полученные оценки пригодности ЦКИ являются вероятностными и могут использоваться на этапе выбора масштаба ЦКМ, который зависит от способа формирования ЦКИ, с учетом точности задач, решаемых в ГИС.

Литература

1. **Радиоэлектронные системы.** Основы теории и построения. Справочник / Ширман Я. Д., Лосев Ю. И., Минервин Н. Н. и др. / Под ред. Я. Д. Ширмана – М.: ЗАО «МАК-ВИС», 1998. – 828 с.
2. **Справочник по радиолокации.** Под ред. М. Скольника. – Нью-Йорк, 1970 / Пер. с англ.; под общей ред. К. Н. Трофимова. – В 4 т. – Т. 1. Основы радиолокации. – М.: Сов. радио, 1976. – 456 с.
3. **Безлюдников О. Л., Деревянко С. А., Севостьянов С. В.** и др. Автоматизация анализа рельефа местности при расчете напряженности поля радиосигналов // Радиотехника. – 2001. – № 9. – с. 86–88.
4. **Цветков В. Я.** Геоинформационные системы и технологии. – М.: Финансы и статистика, 1998. – 288 с.
5. **Басс Ф. Г., Фукс И. М.** Рассеяние волн на статистически неровной поверхности. – М.: Наука, 1972. – 424 с.
6. **Семенов О. И., Васильев В. П.** Основы автоматизации проектирования поверхностей с использованием базисных сплайнов. – Мн.: Наука и техника, 1987. – 167 с.
7. **Ногами М.** Методика и программное обеспечение для построения цифровой модели рельефа по карте с горизонталями / Пер. с яп. в реферативном журнале «Картография», 1992. – Тидзу. Март. – 1991. – Т. 29. – № 3. – с. 20–26.
8. **Самотонин Д. Н., Биденко С. И.** Сплайн-интерполяция влияния рельефа местности при моделировании распространения радиоволн с использованием векторных гео данных // Информационно-управляющие системы. – 2004. – № 3. – с. 22–30.
9. **Hyre Y. S., Moore R. K.** Theoretical Scattering Coefficients for Near-vertical Incidence from Contour Maps // J. Res. Natl. Bur. Std. – Vol. 65D. – N 5. – September-October. – 1961. – p. 427–432.
10. **Тихонов В. И.** Статистическая радиотехника. – 2-е изд., перераб. и доп. – М.: Радио и связь, 1982. – 624 с.

УДК 681.3.06

STATE MACHINE — РАСШИРЕНИЕ ЯЗЫКА JAVA ДЛЯ ЭФФЕКТИВНОЙ РЕАЛИЗАЦИИ АВТОМАТОВ

Н. Н. Шамгунов,

аспирант

Г. А. Корнеев,

аспирант

А. А. Шалыто,

доктор техн. наук, профессор

Санкт-Петербургский государственный университет информационных технологий, механики и оптики

В данной статье предлагается новый язык объектно-ориентированного программирования State Machine, являющийся расширением языка программирования Java. В язык вводится ряд синтаксических конструкций, позволяющих программировать с использованием понятий автомат, состояние и событие. Для обработки новых синтаксических конструкций разработан препроцессор, преобразующий код на языке State Machine в код на языке Java. При этом новые синтаксические конструкции преобразуются в код на языке Java в соответствии с предложенным ранее авторами паттерном State Machine.

This paper presents a new Object-oriented programming language State Machine which is an extension of the Java language. This language presents a number of syntactic constructions that provides an ability to create programs with using notion of automaton, state and event. Author developed a preprocessor that transforms State Machine code to Java code. This preprocessor transforms new syntactic constructions to Java code according to State Machine pattern which was presented by these authors earlier.

Введение

В программировании часто возникает потребность в объектах, изменяющих свое поведение в зависимости от состояния. Обычно поведение таких объектов описывается при помощи конечных автоматов. Существуют различные паттерны проектирования для реализации указанных объектов, приведенные, например, в работах [1, 2]. В большинстве из этих паттернов или автоматы реализуются неэффективно, или сильно затруднено повторное использование компонентов автомата. Эти недостатки устранены в предложенном авторами паттерне State Machine [3].

В последнее время имеет место тенденция создания языков, ориентированных на предметную область [4]. В данном случае такой областью является автоматное программирование.

Одним из способов создания предметно-ориентированных языков является расширение существующих, например, за счет введения в них автоматных конструкций [5, 6].

В работе [7] был предложен язык State (расширяющий язык C# [8]), который предназначен для реализации объектов с изменяющимся поведением.

Однако конструкции, вводимые в этом языке, невозможно реализовать эффективно, поскольку в нем вызов метода объекта влечет за собой вычисление набора предикатов.

Зарекомендовавшим себя способом расширения языков программирования является встраивание в них поддержки паттернов проектирования. Так, в язык C# встроено паттерн Observer [1], широко используемый, например, для реализации графического интерфейса пользователя.

В данной работе предлагается язык программирования State Machine, расширяющий язык Java [9] и основанный на одноименном паттерне. В качестве основного языка был выбран язык программирования Java, так как для него существуют современные инструменты создания компиляторов, не только представленные в документации, но и описанные в монографии [11].

Особенности языка State Machine

В предлагаемом языке, как и в паттерне State Machine, основной идеей является описание объектов, варьирующих свое поведение, в виде автоматов. В предложенном в работе [3] подходе разде-

ляются классы, реализующие логику переходов (контексты), и классы состояний. Переходы инициируются состояниями путем уведомления контекста о наступлении событий. При этом в зависимости от события и текущего состояния контекст устанавливает следующее состояние в соответствии с графом переходов.

Отметим, что если в паттерне *State* [1] следующее состояние указывается текущим, то в паттерне *State Machine* это выполняется путем уведомления класса контекста о наступлении события.

Логика переходов задается в терминах состояний и событий. При этом в языке *State Machine* полностью скрывается природа событий. Для пользователя они представляют собой сущности, принадлежащие классу состояния и участвующие в формировании таблицы переходов. Для описания логики переходов на этапе проектирования используются специальные графы переходов, предложенные в работе [3]. Эти графы состоят только из состояний и переходов, помеченных событиями.

По сравнению с языком *Java* в язык *State Machine* введены дополнительные конструкции, позволяющие описывать объекты с варьирующимся поведением в терминах автоматного программирования, определенных в работе [3]: *автоматов, состояний и событий*. Для описания автоматов и состояний в язык введены ключевые слова *automaton* и *state* соответственно, а для событий — ключевое слово *events*.

Отметим, что в предлагаемом языке, так же как и в паттерне *State Machine*, события являются основным способом воздействия объекта состояния на контекст. В указанном паттерне программист должен самостоятельно создавать объекты, представляющие события в виде статических переменных класса *Event*, в то время как в языке *State Machine* события введены как часть описания состояний. Это сделано для того, чтобы подчеркнуть их важность.

В паттерне *State Machine* реализация интерфейса автомата в контексте делегирует вызовы методов интерфейса текущему экземпляру состояния, причем делегирование реализуется вручную. В программе на языке *State Machine* делегировать вызовы не требуется, так как препроцессор автоматически сгенерирует соответствующий код. Для этого используется технология *Reflection* [12]. Поэтому важно, чтобы препроцессор и генерируемый им код были реализованы на одном языке программирования. В частности, если бы язык расширял язык *C#* (как язык *State*), то и сам препроцессор необходимо было бы написать на языке *C#*.

Так же как в паттерне *State Machine*, в предлагаемом языке состояние может делегировать дальнейшее выполнение действия автомату. При этом для ссылки на автомат используется ключевое слово *automaton* (как и при описании автоматов).

Делегирование действия автомату требуется, например, при восстановлении после ошибки (соответствующий пример рассмотрен в разделе «По-

вторное использование»). В этом случае состояние, обрабатывающее ошибку, осуществляет действия по восстановлению и, в случае успеха, передает управление новому состоянию автомата.

Описания автоматов и состояний на языке *State Machine* помещаются в файлы с расширением *.sm*. Авторами разработан препроцессор, преобразующий код, написанный на предлагаемом языке, в код на языке *Java* (в файлы с расширением *.java*). При этом новые синтаксические конструкции преобразуются в соответствии с паттерном *State Machine*. Препроцессор генерирует код, содержащий параметры типа *(genetics)* [10], что позволяет осуществлять проверку типов во время компиляции. Полученный код компилируется при помощи *Java*-компилятора, поддерживающего параметры типа

Пример использования языка *State Machine*

Рассмотрим особенности новых синтаксических конструкций языка *State Machine* на примере проектирования и реализации класса *Connection*, описанного в работе [3], и приведем его краткое описание.

Описание примера. Требуется спроектировать класс *Connection*, представляющий сетевое соединение, имеющее два управляющих состояния: *соединено* и *разъединено*. Переход между ними происходит или при возникновении ошибки посредством вызовов методов *установить соединение (connect)* и *разорвать соединение (disconnect)*. В состоянии *соединено* может производиться получение (метод *receive*) и отправка (метод *send*) данных. В случае возникновения ошибки при передаче данных генерируется исключительная ситуация (*IOException*) и сетевое соединение переходит в состояние *разъединено*, в котором прием и отправка данных невозможны. При попытке осуществить передачу данных в этом состоянии объект также генерирует исключительную ситуацию.

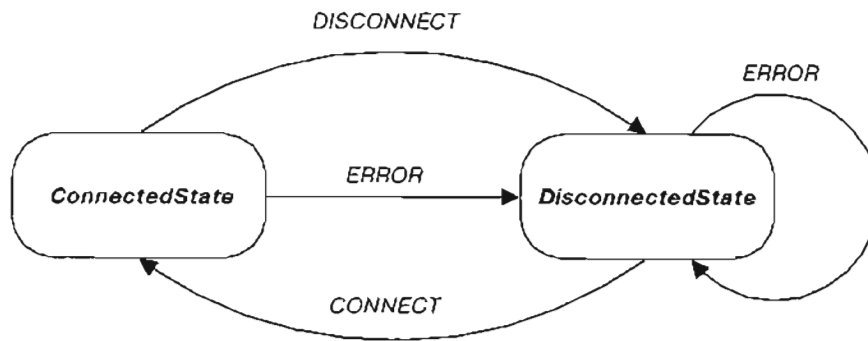
Интерфейс, который требуется реализовать в классе *Connection*, выглядит следующим образом.

```
package connection;

import java.io.IOException;

public interface IConnection {
    public void connect() throws
        IOException;
    public void disconnect() throws
        IOException;
    public int receive() throws
        IOException;
    public void send(int value) throws
        IOException;
}
```

В работе [3] для реализации состояний *соединено* и *разъединено* предложено использовать классы *ConnectedState* и *DisconnectedState* соответственно. Состояния уведомляют контекст о событиях. Класс *ConnectedState* — о событиях *DISCONNECT* и



■ Рис. 1. Граф переходов для класса Connection

ERROR, а класс DisconnectedState — о событиях CONNECT и ERROR.

Для рассматриваемого примера на рис. 1 представлен граф переходов вида, используемого в предлагаемом подходе [3].

Описание состояний. Для описания состояния используется ключевое слово state. Приведем код состояния ConnectedState на языке State Machine:

```

package connection;
import java.io.IOException;

public state ConnectedState implements
    IConnection events DISCONNECT,
    ERROR {
    protected final Socket socket;

    public ConnectedState(Socket socket) {
        this.socket = socket;
    }

    public void connect() throws
        IOException {
    }

    public void disconnect() throws
        IOException {
        try {
            socket.disconnect();
        } finally {
            castEvent(DISCONNECT);
        }
    }

    public int receive() throws IOException
    {
        try {
            return socket.receive();
        } catch (IOException e) {
            castEvent(ERROR);
            throw e;
        }
    }

    public void send(int value) throws
        IOException {
        try {
            socket.send(value);
        } catch (IOException e) {
            castEvent(ERROR);
            throw e;
        }
    }
}
  
```

При описании состояния указан интерфейс автомата (IConnection) и список событий, которые это состояние может сгенерировать (ERROR,

DISCONNECT). Так же как в паттерне State Machine, контекст уведомляется о наступлении события вызовом метода castEvent. За исключением этого, состояния описываются аналогично классу на языке Java.

В предлагаемом языке состояние может реализовывать несколько интерфейсов. При этом первый из реализуемых состоянием интерфейсов будет считаться интерфейсом автомата.

Для реализации автомата Connection необходимо также описать состояние DisconnectedState:

```

package connection;
import java.io.IOException;

public state DisconnectedState implements
    IConnection events CONNECT,
    ERROR {
    protected final Socket socket;

    public DisconnectedState(Socket
        socket) {
        this.socket = socket;
    }

    public void connect() throws
        IOException {
        try {
            socket.connect();
            castEvent(CONNECT);
        } catch (IOException e) {
            castEvent(ERROR);
            throw e;
        }
    }

    public void disconnect() throws
        IOException {
    }

    public int receive() throws
        IOException {
        throw new IOException("Connection
            is closed");
    }

    public void send(int value) throws
        IOException {
        throw new IOException("Connection
            is closed");
    }
}
  
```

Описание автомата. В языке State Machine автомат предназначен для определения набора состояний и переходов.

Для описания автомата применяется ключевое слово `automaton`. Приведем код на предлагаемом языке для автомата `Connection`, реализующий граф переходов (см. рис. 1)

```
package connection;

public automaton Connection implements
    IConnection {
    state DisconnectedState
    disconnected(CONNECT ->
    connected, ERROR ->
    disconnected);
    state ConnectedState connected(ERROR ->
    disconnected, DISCONNECT ->
    disconnected);
    public Connection(Socket socket) {
    disconnected @= new
    DisconnectedState(socket);
    connected @= new
    ConnectedState(socket);
    }
}
```

Обратим внимание, что класс автомата должен реализовывать ровно один интерфейс, который и считается интерфейсом автомата. В данном примере — это интерфейс `IConnection`.

Состояния (`connected` и `disconnected` классов `ConnectedState` и `DisconnectedState` соответственно) описываются при помощи ключевого слова `state`. Первое из состояний, описанных в автомате, является стартовым. В данном примере это состояние `disconnected`.

Переходы по событиям описываются в круглых скобках, после имени состояния. Для одного состояния переходы разделяются запятыми. Например, для состояния `connected` переходами являются `DISCONNECT -> disconnected` и `ERROR -> disconnected`. Первый из них означает, что при поступлении события `DISCONNECT` в состоянии `connected` автомат переходит в состояние `disconnected`.

В конструкторе `public Connection(Socket socket)` производится создание объектов состояний. Отметим, что состояния, входящие в автомат, должны реализовать интерфейс автомата. Инициализация объектов состояний производится при помощи нового оператора `@=`, специально введенного для этой цели в язык *State Machine*. Таким образом, оператор `connected @= new ConnectedState(socket)` означает инициализацию состояния `connected` новым объектом класса `ConnectedState`.

За исключением этого, автомат описывается аналогично классу на языке *Java*.

Отметим, что состояния автомата перечисляются, но не определяются в нем. Таким образом, одни и те же состояния могут использоваться для реализации различных автоматов.

Компиляция примера. Для генерации *Java*-кода из файлов с расширением `.sm` необходимо выполнить команду `java ru.ifmo.is.sml.Main <имя файла1> [<имя файла2>, <имя файлаN>]`.

В результате будет сформирован одноименный файл с расширением `.java`. Отметим, что для гене-

рации класса `Connection` необходимо предварительно скомпилировать *Java*-компилятором интерфейс `IConnection`. Это требуется для генерации реализации этого интерфейса в соответствующем классе.

Для полной компиляции данного примера необходимо выполнить следующую последовательность команд:

```
rem Компиляция интерфейса автомата
javac IConnection.java

rem Преобразование состояний
java ru.ifmo.is.sml.Main ConnectedState.sm
    DisconnectedState.sm

rem Компиляция классов состояний
javac ConnectedState.java
    DisconnectedState.java

rem Преобразование автомата Connection
java ru.ifmo.is.sml.Main Connection.sm

rem Компиляция автомата
javac Connection.java
```

В результате будут сформированы соответствующие *Java*-файлы, которые будут скомпилированы *Java*-компилятором `javac`.

Отметим, что для компиляции и работы полученных классов требуется только класс `AutomatonBase`, определенный в пакете `ru.ifmo.is.sml.runtime`.

Грамматика описания автоматов и состояний

Как отмечено выше, язык программирования *State Machine* основан на языке *Java*, в который вводятся синтаксические конструкции для поддержки программирования в терминах *автомат* и *состояние*.

В данном разделе приводятся грамматики в расширенной форме Бэкуса–Наура [13] для описания этих конструкций.

Грамматика описания состояния:

```
state_decl ::=modifiers
state type extends_decl? implements_decl
events? (balanced)
extends_decl ::=extends type
implements_decl ::=implements type (, type)*
type ::=id (, id)^
events ::=events id (, id)^
balanced ::=«сбалансированная по
скобкам последовательность»
modifiers ::= (abstract | final |
strictfp | public)^
```

Здесь и далее терминальные символы выделены полужирным шрифтом, а нетерминальные — курсивным. Для краткости не раскрывается определение нетерминала `balanced`. Он соответствует сбалансированной относительно использования круглых и фигурных скобок последовательности терминальных и нетерминальных символов [14].

Состояние должно реализовывать не менее одного интерфейса. При этом первый из них считается интерфейсом автомата.

В коде состояния возможно делегирование методов текущему состоянию автомата. Для этого используется ключевое слово `automaton`, которое имеет тип интерфейса автомата.

Отметим, что в данной версии языка состояния не могут содержать параметры типа.

Грамматика описания автомата:

```

automaton_decl ::= modifiers automaton type
                implements_decl (state_var_decl+
                balanced )
state_var_decl ::= state type id {
                event_mapping (, event_mapping)*
                };
event_mapping ::= id (, id)* -> id
    
```

Отметим, что интерфейс автомата должен совпадать у автомата и всех состояний, которые он использует. Это семантическое правило, поэтому оно не может быть выражено грамматикой.

Для инициализации состояний в конструкторе автомата применяется оператор `@=`. Слева от него указывается имя состояния, а справа – объект, реализующий это состояние. Тип указанного объекта должен в точности совпадать с типом, указанным при описании автомата.

В конструкторе все состояния автомата должны быть проинициализированы, при этом каждое – не более одного раза (как если бы они были обыкновенными переменными, описанными с модификатором `final`).

Использование оператора `@=` вне конструктора является ошибкой.

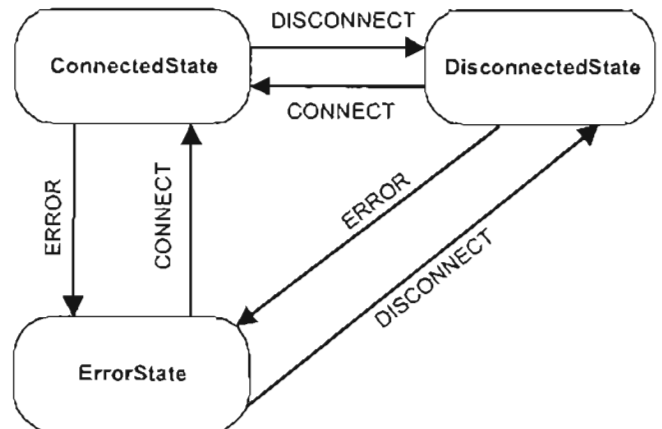
Повторное использование

Одним из преимуществ объектно-ориентированного программирования является возможность повторного использования кода. Эта возможность также поддерживается и в предлагаемом языке.

Допустимые способы повторного использования. В объектно-ориентированном программировании на языке *Java* интерфейс объекта представляет собой некоторый контракт [18]. При этом возможно наследование класса, основанное только на его контракте. Наследование от автомата как класса допустимо, но для расширения его поведения в наследнике необходимо изменять набор состояний и функцию переходов. Таким образом, пользователь не может воспринимать базовый автомат как черный ящик, поскольку доступ к реализации функции переходов базового автомата нарушает инкапсуляцию. Поэтому наследование автомата от класса или другого автомата в языке *State Machine* запрещено.

При этом предлагаемый язык допускает повторное использование классов состояний. Так же как и в паттерне *State Machine*, это может быть сделано двумя способами.

- 1) наследование состояний;
- 2) использование классов состояний в нескольких автоматах



■ Рис. 2. Граф переходов класса `ResumableConnection`

В первом способе создаются наследники состояний, реализующие более широкий интерфейс по сравнению с базовым состоянием. Из полученных состояний конструируется новый автомат. Во втором способе одни и те же классы состояний используются для конструирования различных автоматов. В обоих случаях определяется новый автомат со своим набором состояний и переходов в нем. Также возможна комбинация этих подходов.

Ниже оба способа повторного использования классов состояний будут рассмотрены на примерах.

Описание примеров. Следуя работе [3], рассмотрим две модификации автомата `Connection`.

Автомат `PushBackConnection` предоставляет возможность возврата данных в объект соединения и их последующего считывания. Интерфейс этого автомата – `IPushBackConnection` расширяет интерфейс `IConnection`:

```

package push_back_connection;

import connection.IConnection;
import java.io.IOException;

public interface IPushBackConnection extends
    IConnection {
    void pushBack(int value) throws
        IOException;
}
    
```

Автомат `ResumableConnection` реализует следующую логику поведения класса, представляющего сетевое соединение. В случае возникновения ошибки при передаче данных автомат закрывает канал связи и генерирует исключительную ситуацию. При очередном вызове метода передачи данных производится попытка восстановить соединение.

Для описания такого поведения требуется ввести новое состояние – *ошибка*, переход в которое означает, что канал передачи данных закрыт из-за ошибки. Полученный граф переходов используемого в предлагаемом подходе вида изображен на рис. 2.

Наследование состояний. Для реализации автомата `PushBackConnection` необходимо описать

два новых состояния `PushBackConnectedState` и `PushBackDisconnectedState`, унаследованных от состояний `ConnectedState` и `DisconnectedState` соответственно. Это позволяет обеспечить повторное использование кода состояний.

Приведем код состояния `PushBackConnectedState`:

```
package push_back_connection;

import connection.*;
import java.util.Stack;
import java.io.IOException;

public state PushBackConnectedState extends
    ConnectedState implements
    IPushBackConnection {
    private final Stack<Integer> stack =
        new Stack<Integer>();

    public PushBackConnectedState(Socket
        socket) {
        super(socket);
    }

    public int receive() throws IOException
    {
        return stack.empty() ?
            super.receive() : stack.pop();
    }

    public void pushBack(int value) {
        stack.push(value);
    }
}
```

Состояние `PushBackDisconnectedState` реализуется аналогично:

```
package push_back_connection;

import connection.*;
import java.io.IOException;

public state PushBackDisconnectedState
    extends DisconnectedState
    implements IPushBackConnection {
    public PushBackDisconnectedState(Socket
        socket) {
        super(socket);
    }

    public void pushBack(int value) throws
        IOException {
        throw new IOException("Connection is
            closed (pushBack)");
    }
}
```

Автомат `PushBackConnection` не наследует автомат `Connection`. Повторное использование кода достигается за счет применения наследников состояний `ConnectedState` и `DisconnectedState`.

```
package push_back_connection;

import connection.Socket;

public automaton PushBackConnection
    implements IPushBackConnection
    states
        PushBackConnectedState connected {
            ERROR -> disconnected,
            DISCONNECT -> disconnected
        },
        PushBackDisconnectedState
        disconnected {
            CONNECT -> connected,
```

```
            ERROR -> disconnected
        }
    }

    public PushBackConnection(Socket
        socket) {
        connected @= new
            PushBackConnectedState(socket);
        disconnected @= new
            PushBackDisconnectedState(socket);
    }
}
```

Использование одного состояния в различных автоматах. Для реализации автомата `ResumableConnection` необходимо дополнительно реализовать состояние `ErrorState`, определяющее поведение в состоянии *ошибка*. В автомате также будут использованы состояния `ConnectedState` и `DisconnectedState`, разработанные для автомата `Connection`.

Приведем код состояния `ErrorState`:

```
package resumable_connection;

import connection.*;
import java.io.IOException;

public state ErrorState implements
    IConnection events CONNECT,
    DISCONNECT {
    protected final Socket socket;

    public ErrorState(Socket socket) {
        this.socket = socket;
    }

    public void connect() throws
        IOException {
        socket.connect();
        castEvent(CONNECT);
    }

    public void disconnect() throws
        IOException {
        castEvent(DISCONNECT);
    }

    public int receive() throws IOException
    {
        connect();
        return automaton.receive();
    }

    public void send(int value) throws
        IOException {
        connect();
        automaton.send(value);
    }
}
```

Теперь можно определить автомат `ResumableConnection`:

```
package resumable_connection;

import connection.*;

public automaton ResumableConnection
    implements IConnection {
    state DisconnectedState
        disconnected(CONNECT ->
            connected, ERROR -> error);
    state ConnectedState
        connected(DISCONNECT ->
            disconnected, ERROR -> error);
    state ErrorState error(DISCONNECT ->
        disconnected, CONNECT ->
        connected);
}
```

```
private ResumableConnection() {
    Socket socket = new Socket();

    connected @= new
        ConnectedState(socket);
    disconnected @= new
        DisconnectedState(socket);
    error @= new ErrorState(socket);
}
}
```

Из приведенных примеров следует, что состояния могут быть использованы повторно.

Реализация препроцессора

Для реализации препроцессора были использованы инструменты создания компиляторов *JLex* и *Cup*, описанные в работе [11], которые распространяются по лицензии *open source license* [15]. Первый из них предназначен для построения лексических анализаторов, а второй — синтаксических [14].

В результате работы препроцессора производится преобразование переданных ему в качестве параметров файлов с расширением *.sm*, содержащих код автоматов и состояний на языке *State Machine*, в файлы с расширением *.java*. В процессе преобразования теряется исходное форматирование программы и комментарии. Однако это не является недостатком, поскольку получаемый код — промежуточный и не предназначен для редактирования вручную.

Препроцессор (в открытых кодах) можно скачать по адресу <http://is.ifmo.ru>, раздел *Статьи*. Для работы препроцессора необходимо также установить инструменты создания компиляторов *JLex* и *Cup*, доступные по адресу [16].

Генерация Java-классов по описанию состояний. Каждое состояние, описанное на языке *State Machine*, преобразуется в одноименный класс на языке *Java*. При этом все методы и их реализация переходят в генерируемый класс.

В случае, если, одно состояние расширяет другое, то генерируемый класс будет расширять соответствующий ему *Java*-класс. В противном случае генерируемый класс будет расширять класс *AutomatonBase.StateBase*, входящий в пакет *ru.ifmo.is.sml.runtime*.

Указанный класс является базовым для всех классов состояний. В нем определено поле *automaton*, позволяющее вызывать методы автомата, в котором содержится данный экземпляр состояния. Отметим, что в языке *State Machine* *automaton* является ключевым словом. Таким образом, конфликтов с полями, определенными пользователем, не возникает.

В классе *StateBase* также реализован метод *sendEvent*, который состояния вызывают для уведомления автомата о наступлении события. Для реализации событий используется класс *AutomatonBase.StateBase.Event*.

Рассмотрим код, сгенерированный препроцессором для состояния *ConnectedState*. Отметим, что здесь и ниже форматирование сгенерированного кода и вставка комментариев выполнены вручную:

```
package connection;

import java.io.IOException;

public class ConnectedState<AI extends
    IConnection>
    // Базовый класс, для всех состояний
    extends ru.ifmo.is.language.Automaton
    Base.StateBase<AI>
    implements IConnection {
    // События преобразуются в набор
    // статических переменных
    public final static Event DISCONNECT =
        new Event(ConnectedState.class,
            "DISCONNECT", 0);
    public final static Event ERROR = new
        Event(ConnectedState.class,
            "ERROR", 1);

    // В остальном — без изменений
    protected final Socket socket;

    public ConnectedState(Socket socket) {
        this.socket = socket;
    }

    public void connect() throws
        IOException {}

    public void disconnect() throws
        IOException {
        try {
            socket.disconnect();
        } finally {
            castEvent(DISCONNECT);
        }
    }

    public int receive() throws IOException
    {
        try {
            return socket.receive();
        } catch (IOException e) {
            castEvent(ERROR);
            throw e;
        }
    }

    public void send(int value) throws
        IOException {
        try {
            socket.send(value);
        } catch (IOException e) {
            castEvent(ERROR);
            throw e;
        }
    }
}
```

Обратим внимание, что в этом коде, так же как в паттерне *State Machine*, для представления событий используются статические переменные. При создании экземпляров событий (класса *Event*) первыми двумя параметрами конструктора являются ссылка на класс, определивший событие и имя события, которые могут быть использованы для автоматического протоколирования [17]. Третий параметр необходим для эффективной реализации графов переходов. Он представляет собой порядковый номер события в состоянии, отсчитываемый от нуля. При этом учитываются события, определенные в базовом состоянии, которым присваиваются меньшие номера.

Генерация Java-классов по описанию автоматов. Автомат в языке *State Machine* также преобразуется в одноименный класс, унаследованный от

класса AutomatonBase, определенного в пакете ru.ifmo.is.sml.runtime.

Препроцессор автоматически генерирует методы, реализующие интерфейс автомата. В каждом из них вызывается соответствующий метод текущего состояния. Такой метод гарантированно присутствует в состоянии, поскольку автомат и состояние реализуют интерфейс автомата.

Для автомата Connection препроцессор генерирует следующий Java-класс:

```
package connection;

import java.io.IOException;
import ru.ifmo.is.sml.runtime.AutomatonBase;

public class Connection
    extends AutomatonBase<IConnection>
    implements IConnection
{
    private final static int[][]
        $TRANSITION_TABLE
        = new int[][] {
        {
            /* CONNECT */
            0, /* connected */
            /* ERROR */
            1, /* disconnected */
        }, {
            /* DISCONNECT */
            1, /* disconnected */
            /* ERROR */
            1, /* disconnected */
        }
    };

    public Connection(Socket socket) {
        super(new IConnection[2/*число состояний*/]);
        {
            DisconnectedState<IConnection>
            $state = new DisconnectedState
            <IConnection>(socket);
            state(0/*disconnected*/, $state,
            $state, this,
            $TRANSITION_TABLE[
            0/*disconnected*/]);
        }
        {
            ConnectedState<IConnection>
            $state = new
            ConnectedState<IConnection>(socket);
            state(1/*connected*/, $state,
            $state, this,
            $TRANSITION_TABLE[1/*connected*/
            ]);
        }
    }

    // Делегирование методов интерфейса
    автомата
    public void connect() throws
    IOException { state.connect(); }
    public void disconnect() throws
    IOException {
        state.disconnect(); }
    public int receive() throws IOException
    { return state.receive(); }
    public void send(int value) throws
    IOException { state.send(value); }
}
}
```

Отметим, что использование имени state для поля, хранящего текущее состояние, не приводит к неоднозначности, так как в предлагаемом языке оно является ключевым словом. По этой же причине

метод, связывающий состояние с автоматом, также называется state. Интересной особенностью является необходимость дважды передавать в этот метод ссылку на состояние (state), а также ссылку на сам автомат (this). Это связано с невозможностью на языке Java определить класс, унаследованный от своего параметра типа.

На основе переходов, заданных в описании автомата, препроцессор строит таблицу переходов, хранящуюся в статическом поле \$TRANSITION_TABLE. Таблица представляет собой массив массивов целых чисел. В приведенном коде таблица переходов является матрицей 2x2. Однако, в общем случае, она матрицей не является, поскольку состояния могут порождать разное количество событий.

Строки таблицы переходов передаются объектам состояний при помощи вызова метода init, входящего в класс AutomatonBase. Это позволяет представить функцию уведомления о событии (castEvent) в компактном виде.

При такой реализации для осуществления каждого перехода требуется фиксированное время, существенно меньшее, чем в реализации, предложенной в работе [3]. Таким образом, базовый класс для всех автоматов выглядит следующим образом.

```
package ru.ifmo.is.sml.runtime;

public abstract class AutomatonBase<AI> {
    private final AI[] states;
    protected AI state;

    public AutomatonBase(AI[] states) {
        this.states = states;
    }

    protected void state(int index,
        StateBase<AI> state, AI stateI,
        AI automaton, int[] transitions)
    {
        states[index] = stateI;
        state.base = this;
        state.automaton = automaton;
        state.transitions = transitions;
        if (index == 0) this.state =
            states[index];
    }

    public static abstract class
        StateBase<AI> {
        protected AI automaton;
        private AutomatonBase<AI> base;
        private int[] transitions;

        protected void castEvent(Event
            event) {
            base.state =
            base.states[transitions[event.index]];
        }

        public final static class Event {
            private final Class state;
            private final String name;
            private final int index;

            public Event(Class state, String
                name, int index) {
                this.state = state;
                this.name = name;
                this.index = index;
            }
        }
    }
}
```


Обратим внимание, что класс AutomatonBase содержит класс StateBase как вложенный.

Выводы

Предложенный язык программирования State Machine обладает следующими достоинствами.

1. Позволяет писать программы в терминах автоматного программирования.
2. Непосредственно поддерживает одноименный паттерн.

3. Повышает компактность и облегчает восприятие кода, по сравнению с реализацией паттерна State Machine непосредственно на языке Java.

4. Обеспечивает более быстрое выполнение переходов по сравнению с реализацией паттерна State Machine, приведенного в работе [3].

Предложенный синтаксис позволяет распознавать паттерн в коде, в том числе и автоматически, что может быть использовано при построении диаграмм и документации.

Литература

1. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns. MA: Addison-Wesley Professional, 2001. – 395 P. (Гамма Э., Хелм Р., Джонсон Р., Влассидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб.: Питер, 2001. – 368 с.)
2. Adamczyk P. The Anthology of the Finite State Machine Design Patterns (<http://jerry.cs.uiuc.edu/~plop/plop2003/Papers/Adamczyk-State-Machine.pdf>).
3. Шамгунов Н. Н., Корнеев Г. А., Шалыто А. А. Паттерн State Machine для объектно-ориентированного проектирования автоматов // Информационно-управляющие системы. – 2004. – № 5. – С. 13–25.
4. Эндрю Х., Дэвид Т. Программист-прагматик. – М.: Лори, 2004. – 288 с.
5. Язык прикладного программирования STL 1.0. Руководство пользователя (v1.0.13b) (http://imt-automation.ifmo.ru/pdfs/stlguide_1_0_13b.pdf).
6. Ваганов С. А. FloraWare — ускорить разработку приложений (<http://www.softcraft.ru/paradigm/oop/flora/index.shtml>).
7. Шамгунов Н. Н., Шалыто А. А. Язык автоматного программирования с компиляцией в Microsoft CLR // Microsoft Research Academic Days in St. Petersburg, 2004.
8. <http://msdn.microsoft.com/vcsharp/team/language/default.aspx> C# Language.
9. Gosling J., Joy B., Steele G., Bracha G. The Java Language Specification (http://java.sun.com/docs/books/jls/second_edition/html/j.title.doc.html).
10. Generics (<http://java.sun.com/j2se/1.5.0/docs/guide/language/generics.html>).
11. Appel A. W. Modern Compiler Implementation in Java. – NY, Cambridge, 1998. – 512 с.
12. Green A. Trail: The Reflection API (<http://java.sun.com/docs/books/tutorial/reflect/>).
13. Naur P. et al. Revised Report on the Algorithmic Language ALGOL 60 // Communications of the ACM. – 1960. – Vol. 3. – N 5. – p. 299–314.
14. Aho A., Sethi R., Ullman J. Compilers: Principles, Techniques and Tools. MA: Addison-Wesley, 1985, 500 p. (Ахо А., Сети Р., Ульман Дж. Компиляторы: принципы, технологии и инструменты. М.: Вильямс, 2001. — 768 с).
15. Open Source License (<http://www.opensource.org/licenses/historical.php>).
16. <http://www.cs.princeton.edu/~appel/modern/java/> Modern Compiler Implementation in Java.
17. Шалыто А. А., Туккель Н. И. SWITCH-технология – автоматный подход к созданию программного обеспечения «реактивных» систем // Программирование. – 2001. – № 5. – С. 45–62. (<http://is.ifmo.ru>, раздел «Статьи»).
18. Object-Oriented Programming Concepts (<http://java.sun.com/docs/books/tutorial/java/concepts/>).

УДК 621.391.251

МНОГОПороговое декодирование кодов с низкой плотностью проверок на четность

А. В. Белоголовый,

канд. техн. наук, ассистент

Е. А. Крук,

доктор техн. наук, профессор

Санкт-Петербургский государственный университет аэрокосмического приборостроения

В статье предлагается новый алгоритм декодирования. Новый алгоритм принадлежит к классу быстрых алгоритмов, но благодаря использованию порогов при декодировании качество работы приближается к результатам медленных декодеров. В работе приводится описание алгоритма и сравнение его с традиционными и быстрыми декодерами на основе моделирования.

In paper the new decoding algorithm is proposed. The new algorithm belongs to fast decoding algorithms class, but due to the usage of threshold in decoding the performance becomes closer to slow decoders results. In paper there are shown the description of algorithm and the comparison with traditional and fast decoders based on simulations.

Коды с низкой плотностью проверок на четность

Коды с низкой плотностью проверок на четность были впервые предложены Р. Галлагером в 1963 году как высокоскоростные коды с малой вероятностью ошибки для передачи по каналам с шумом [1, 2].

Коды с низкой плотностью проверок на четность – это линейные коды, определяемые проверочной матрицей, содержащей в основном нулевые элементы и относительно небольшое количество ненулевых. Для случая двоичных кодов с низкой плотностью проверок на четность проверочная матрица состоит из небольшого числа единиц и нулей на всех остальных позициях матрицы. Пример проверочной матрицы с низкой плотностью проверок на четность:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Для всего класса кодов с низкой плотностью проверок на четность [1–3] существуют декодеры, основывающиеся только на свойствах разреженности проверочной матрицы и не использующие никакие свойства самого пространства кодовых слов. Эти алгоритмы декодирования имеют низкую сложность и позволяют достичь малой вероятности ошибки при передаче в канале с аддитивным нормально распределенным (гауссовским) шумом [1–7].

Традиционное декодирование кодов с низкой плотностью проверок на четность

Рассмотрим итеративное декодирование кодов с низкой плотностью проверок на четность в двоичном симметричном канале и канале с аддитивным белым гауссовским шумом.

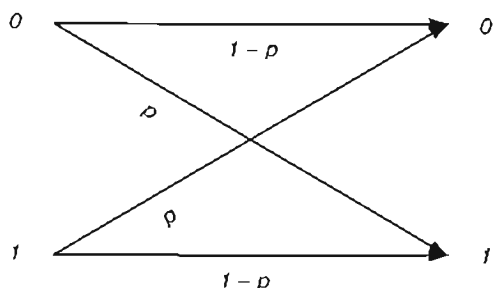
Симметричные каналы с двоичным входом. Симметричным каналом с двоичным входом будем называть дискретный по времени канал со следующими свойствами:

входной алфавит X состоит из двух символов, обозначаемых 0 и 1;

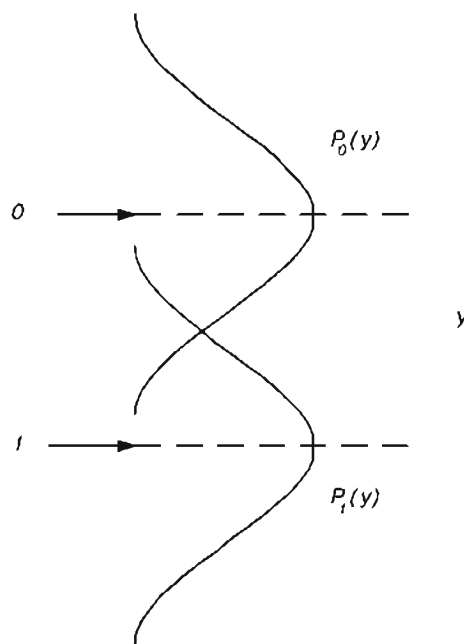
выходной алфавит Y может быть представлен как дискретное либо непрерывное вещественное множество чисел;

выход y в заданный отсчет времени зависит только от одного входного символа x ;

а)



б)



■ Рис. 1. Симметричные каналы с двоичным входом: а – двоичный симметричный канал; б – канал с аддитивным гауссовским шумом

для всех выходов y выполняется свойство симметрии: $P_0(y) = P_1(-y)$. Здесь под $P_x(y)$ понимается плотность распределения вероятности, если y – непрерывный выход, и условная вероятность, если Y – дискретное множество.

На рис. 1 приведены примеры симметричных каналов с двоичным входом.

Под вектором ошибки для кодового слова $c = \{c_0, \dots, c_{N-1}\}$ длины N будем понимать вектор $e = \{e_0, \dots, e_{N-1}\}$ такой же длины, соответствующие элементы которого равны разностям между вектором $y = \{y_0, \dots, y_{N-1}\}$, принятым из канала, и передаваемым кодовым словом: $e_i = y_i - c_i, 0 \leq i \leq N$. Эта запись равносильна представлению принятого вектора как суммы кодового слова и ошибки: $y_i = c_i + e_i, 0 \leq i \leq N$.

Декодирование ошибок канала. Декодирование – это процедура поиска и исправления ошибки, наложенной каналом на кодовое слово, по принятому из канала вектору или собственно поиск кодового слова по вектору, принятому из канала.

Декодирование по максимуму правдоподобия кода C обозначает нахождение по заданному принятому вектору y такого кодового слова $c \in C$, которое максимизирует вероятность того, что передавалось слово c при условии принятия вектора $y: p(c/y) \rightarrow \max$ [51–54, 56, 60]. Задача декодирования по максимуму правдоподобия является NP-полной.

Для оценки качества работы различных декодеров используется оценка вероятности ошибки декодирования на информационный бит, вычисляемая как отношение количества ошибочных информационных бит после декодирования к общему количеству переданных информационных бит:

$$BER = \frac{N_{\text{ошибок}}}{N_{\text{всех бит}}}$$

Итеративные схемы декодирования кодов с низкой плотностью проверок на четность не являются декодерами по максимуму правдоподобия, но позволяют получить разумный баланс по сложности и вероятности ошибки декодирования по сравнению с декодированием по максимуму правдоподобия. Итеративное декодирование подразумевает, что нахождение кодового слова будет производиться не за один проход, а за несколько, с последовательным уточнением результата на каждом шаге [1–3, 7].

«Жесткое» декодирование – это схема декодирования для двоичного симметричного канала при небольшом количестве ошибок в канале. «Жесткое» декодирование инвертированием битов – самая простая по сложности схема декодирования кодов с низкой плотностью проверок на четность [1, 2].

Как уже говорилось, под проверкой понимается любая строка $h = \{h_0, \dots, h_{N-1}\}$ из проверочной матрицы кода с низкой плотностью проверок на четность. Будем говорить, что проверка для некоего вектора $y = \{y_0, \dots, y_{N-1}\}$ выполняется тогда, когда скалярное произведение вектора y на проверку дает ноль. Будем говорить, что элемент y_i принятого вектора y участвует в проверке $h = \{h_0, \dots, h_{N-1}\}$ тогда, когда соответствующий элемент проверки h_i не равен нулю.

Одна итерация «жесткого» декодирования инвертированием битов производится следующим образом.

1. Для принятого вектора вычисляются все проверки.

2. Если некоторый бит принятого вектора участвовал более чем в половине невыполнившихся проверок, бит инвертируется.

3. После такого анализа всех символов принятого вектора вектор проверяется на принадлежность коду. Если вектор является кодовым словом, декодирование заканчивается, в противном случае выполняется следующая итерация алгоритма.

Такая процедура декодирования применима для кодов с низкой плотностью проверок на четность потому, что большинство проверок в таком случае будут содержать одну ошибку или не будут содержать ошибок вообще, и тогда нарушение большого количества проверок для символа принятого слова будет обозначать наличие в нем ошибки.

Сложность одной итерации «жесткого» декодирования инвертированием бит является линейной, количество итераций декодирования обычно выбирается около $\log_2(N)$, где N – длина кодового слова.

Декодирование по вероятностям является «мягким» декодированием, т. е. декодированием на основе вектора, состоящего не из дискретных значений (0 и 1), а из вещественных величин, полученных на выходе канала [1–3, 6, 7], путем пересчета вероятностей (англ. – «belief propagation decoding»).

Будем называть множеством $N(m)$ множество символов принятого вектора, которые входят в m -ю проверку: $N(m) = \{n : H_{m,n} = 1\}$, где H – проверочная матрица кода с низкой плотностью проверок на четность. Будем называть множеством $M(n)$ множество проверок, в которых участвует n -й символ принятого вектора: $M(n) = \{m : H_{m,n} = 1\}$.

На основе принятого из канала вектора формируются два (для двоичного случая) вектора вероятностей того, что в принятом векторе на данной позиции находился заданный символ:

$$f^0 = (f_0^0, \dots, f_{N-1}^0), \quad f_i^0 = P(y_i = 0)$$

и $f^1 = (f_0^1, \dots, f_{N-1}^1), \quad f_i^1 = P(y_i = 1),$

где y_i – элементы принятого вектора. Каждому ненулевому элементу проверочной матрицы кода с низкой плотностью проверок на четность приписывается две величины: $q_{i,j}^x$ и $r_{i,j}^x$. Величина $q_{i,j}^x$ является вероятностью того, что j -й символ принятого вектора имеет значение x по информации, полученной из всех проверок, кроме i -й. Величина $r_{i,j}^x$ является вероятностью того, что проверка i выполняется, если j -й символ принятого вектора равен x , а все остальные символы проверок имеют распределение вероятностей, заданное величинами $\{q_{i,j}^x : j \in N(i) \setminus j\}$.

Перед началом работы алгоритму требуется инициализация, заключающаяся в том, что значения $q_{i,j}^0$ и $q_{i,j}^1$ принимаются равными f_j^0 и f_j^1 соответственно, а далее алгоритм работает по принципу пересчета вероятностей символов принятого вектора (belief propagation), используя для пересчета вероятностей правило Байеса для апостериорной вероятности события. Одна итерация алгоритма представляет собой следующую последовательность действий.

1. Для всех проверок вычисляются величины

$$\Delta r_{i,j} = \prod_{j' \in N(i) \setminus j} (q_{i,j'}^0 - q_{i,j'}^1) \quad \text{и пересчитываются вероятности } r_{i,j}^x = \frac{1 + (-1)^x \cdot \Delta r_{i,j}}{2} \quad \text{для } x = \{0, 1\}.$$

2. Для всех символов принятого вектора пере-

считываются вероятности $q_{i,j}^x = \alpha_{i,j} \cdot f_j^x \cdot \prod_{i' \in M(j) \setminus i} r_{i',j}^x$ для $x = \{0, 1\}$, где $\alpha_{i,j}$ – нормирующие коэффициенты, обеспечивающие равенство $q_{i,j}^0 + q_{i,j}^1 = 1$.

3. Формируются векторы псевдоапостериорной вероятности q_j^0 и q_j^1 следующим образом:

$$q_j^x = \alpha_j \cdot f_j^x \cdot \prod_{i \in M(j)} r_{i,j}^x, \quad \text{где } \alpha_j \text{ – нормирующие коэффициенты, обеспечивающие равенство } q_j^0 + q_j^1 = 1.$$

4. Формируется вектор решения c^* по следующему правилу: $c_j^* = \begin{cases} 1 & \text{при } q_j^1 > 1/2 \\ 0 & \text{иначе} \end{cases}$. Если вектор c^*

является кодовым словом, декодирование заканчивается, в противном случае выполняется следующая итерация алгоритма.

Сложность данного алгоритма выше, чем сложность «жесткого» декодирования инвертированием битов, однако качество декодирования повышается за счет использования дополнительной информации на выходе канала. Однако качество работы такого алгоритма зависит от инициализации: чем точнее она произведена, тем точнее будет конечный результат. Для канала с гауссовским шумом инициализация может быть произведена при помощи информации о дисперсии шума в канале. Для других распределений шума в канале или при неизвестных характеристиках шума точная инициализация алгоритма может являться сложной задачей.

Быстрое декодирование LDPC

Несмотря на то, что декодирование пересчетом вероятностей является эффективным методом для каналов с непрерывным выходом, тот факт, что сложность его значительно выше, чем сложность «жесткого» декодирования, оставляет место для поиска более быстрых алгоритмов декодирования, обладающих приемлемым качеством.

Среди известных алгоритмов быстрого декодирования кодов с низкой плотностью проверок на четность для каналов с непрерывным выходом наиболее известен алгоритм «min-sum», являющийся упрощением декодера «belief propagation», а также алгоритм «UMP» (Uniformly Most Powerful [7]), который рассматривается ниже.

Пусть из канала был принят вещественный («мягкий») вектор $y = (y_0, \dots, y_{N-1})$. Будем считать надежностью i -го символа вектора y величину r_i , характеризующую удаление принятого значения y_i от

некоего порогового значения thr , при котором все значения переданных символов равновероятны: $\text{thr} : P(1/y_j) = P(0/y_j)$, $r_j = |y_j - \text{thr}|$. Для случая двоичной модуляции, при которой все биты кодового слова отображаются во множество $\{-1, +1\}$ и передаются по симметричному каналу с двоичным входом, надежность принятого символа y будет, таким образом, равна абсолютной величине принятого значения: $r_j = |y_j|$.

Быстрое декодирование по надежностям является собой баланс между «жестким» декодированием и декодированием по вероятностям [1–3]. При декодировании используется вектор «жестких» решений $x = \{x_0, \dots, x_{N-1}\}$ (вектор, состоящий из 0 и 1 и представляющий собой набор дискретных решений) и вектор надежностей $r = \{r_0, \dots, r_{N-1}\}$ (вектор, состоящий из вещественных величин из интервала $[0, +\infty)$). По аналогии с вероятностным декодированием каждому ненулевому символу проверочной матрицы приписываются два числа: $x_{i,j}$ и $r_{i,j}$, представляющие собой «жесткое» решение относительно символа i , полученное при помощи всех проверок, кроме i -й, и надежность символов i -й проверки без учета j -го символа соответственно. Декодирование также является итеративным, одна итерация представляет собой следующую последовательность действий.

1. Для всех проверок и всех символов проверок вычисляется надежность проверок относительно символа j , представляющая собой минимальное значение надежностей всех символов, входящих в проверку, кроме самого символа j :

$$\hat{r}_{i,j} = \min_{j' \in N(i) \setminus j} r_{i,j'}$$

2. Для всех символов принятого вектора и всех проверок производится пересчет надежностей символов проверок: если проверка, в которую входил некоторый символ, не выполняется, то надежность $r_{i,j}$ уменьшается на величину надежности проверки $\hat{r}_{i,j}$, в противном случае надежность $r_{i,j}$ увеличивается на $\hat{r}_{i,j}$; учитываются все проверки, кроме i -й. Если после применения всех проверок для какого-то символа j величина $r_{i,j}$ стала меньше нуля, то обновленная надежность $r_{i,j}$ берется по модулю, а жесткое решение $x_{i,j}$ инвертируется

3. Аналогично пересчитываются надежности принятого вектора: если i -я проверка, в которую входит j -й символ, нарушается, то из надежности r_j вычитается надежность i -й проверки, иначе надежность символа увеличивается на надежность проверки. Если после применения всех проверок для какого-то символа j величина r_j стала меньше нуля, то обновленная надежность $r_{i,j}$ берется по модулю, а жесткое решение $x_{i,j}$ инвертируется.

4. Если «жесткое» решение x является кодовым словом, декодирование заканчивается, иначе декодер начинает следующую итерацию.

Сложность декодера UMP (быстрого декодирования по надежностям) значительно ниже, чем

сложность декодера, пересчитывающего вероятности, за счет того, что пересчет надежностей выполняется по упрощенной схеме (схеме «взвешенного» мажоритарного голосования, в качестве «весов» используется надежность проверок), а также за счет возможности использования исключительно целочисленных операций сложения и сложения по модулю два. Общую сложность одной итерации можно оценить как сумму сложностей $M(2J + \log 2J - 2)$ сравнений, $2NJ$ сложений по модулю два и $2N(J + 4)(J - 1)$ сложений, где N – длина кодового слова, M – количество информационных бит в кодовом слове, J – количество единиц в столбце проверочной матрицы кода с низкой плотностью проверок на четность.

Также к достоинствам быстрого декодера по надежностям можно отнести то, что декодеру не требуется знать характеристики шума в канале (дисперсию и т. д.), а следовательно, такой декодер может работать в любом симметричном канале с двоичным входом. Далее в работе будет показано, как это свойство быстрого декодера по надежностям может быть использовано при кодовом квантовании изображений.

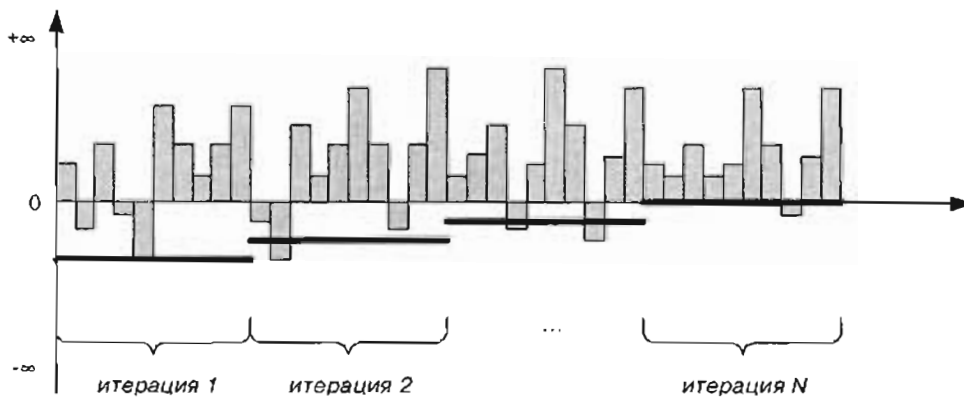
Недостатком быстрого декодера по надежностям является оценка вероятности ошибки декодирования, которая для канала с аддитивным гауссовским шумом оказывается на 0,5 дБ хуже, чем вероятность ошибки декодирования вероятностного декодера. Ниже будет описана модификация алгоритма, позволяющая добиться приближения вероятности ошибки декодирования к вероятности, получаемой при вероятностном декодировании.

Многopороговое декодирование

Быстрый декодер по надежностям, рассмотренный выше, принимает решения относительно символов принятого вектора на основе пересчета надежностей и инвертирования символов «жесткого» решения, причем инвертирование символов производится всегда, когда соответствующая надежность после пересчета становится меньше нуля. Ноль в данном случае является порогом инвертирования символа, причем на всех итерациях этот порог остается неизменным, несмотря на то, что на первых итерациях декодирования надежности символов битов еще плохо определены, и инвертирование таких символов может привести как к исправлению ошибок, так и, наоборот, к внесению новых и последующему их размножению.

Основная идея многopорогового декодирования по надежностям состоит в том, чтобы изменять значения порогов инвертирования символов от одной итерации к другой следующим образом (рис. 2):

на первых итерациях порог инвертирования символов выбирается так, чтобы количество инвертированных символов было минимальным (вплоть до инвертирования только одного символа на первой итерации);



■ Рис. 2. Пороги для многопорогового декодирования: ■ — пересчитанные надежности; — — пороги

на последующих итерациях пороги инвертирования постепенно повышаются.

Общая схема работы многопорогового декодера по надежностям изображена на рис. 3:

1. Инициализация декодера и вычисление надежностей символов проверок производится так же, как и в алгоритме быстрого декодирования по надежностям: для всех элементов принятого вектора Y_i вычисляется жесткое решение X_i и надежность R_i , надежность является абсолютным значением Y_i , а также для всех $m \in M(n)$ присваиваются $Y_{m,n} = R_n$ и $X_{m,n} = X_n$.

2. Для каждой итерации i вычисляется пороговое значение THR_i .

3. Производится вычисление новых значений надежностей проверок и символов, так же, как и в алгоритме быстрого декодирования по надежностям: для всех n и $m \in M(n)$ вычисляются суммы

$$S_{m,n} = X_n \oplus \sum_{n' \in N(m) \setminus n} X_{m,n'}$$

и значения $Y_{m,n} \min = \min\{Y_{m,n'}\}$.

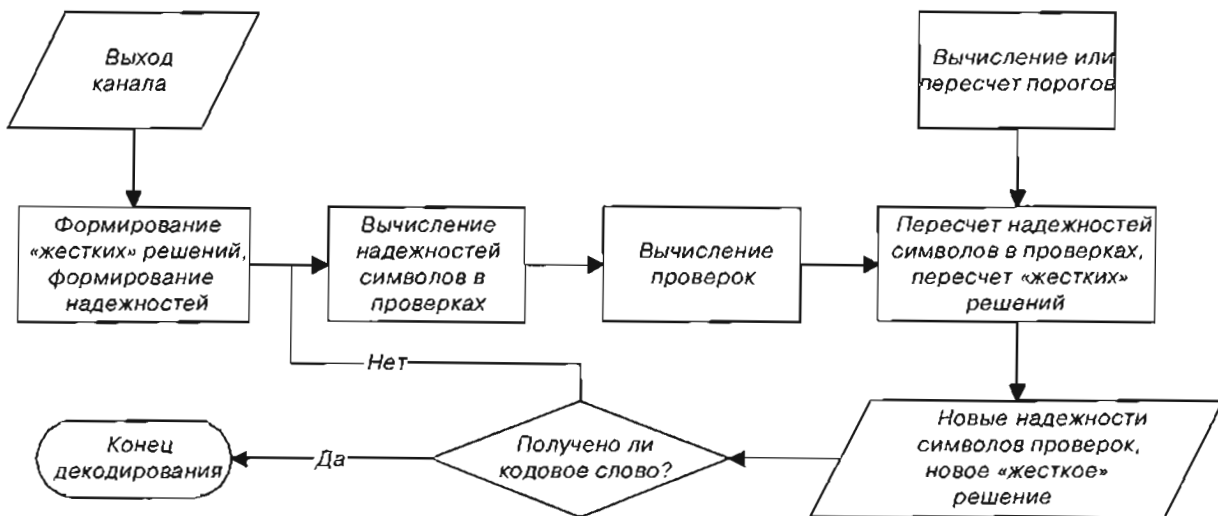
4. Для всех n и всех $m \in N(n)$ вычисляются

$$Z_{m,n} = R_n + \sum_{m' \in M(n) \setminus m} (-1)^{S_{m',n}} Y_{m',n} \min.$$

5. Изменение текущих значений символов «жесткого» решения надежностей проверок и символов на новые значения происходит только для тех символов, для которых новое значение надежности оказывается меньше порога THR_i ; новое значение надежности принимается равным абсолютному значению пересчитанной надежности. Вводится «зона неопределенности»: для всех остальных символов, если значение надежности оказалось меньше нуля, надежность принимается равной нулю: для всех n вычисляются

$$Y_{m,n} = \begin{cases} Z_{m,n}, & Z_{m,n} > 0 \\ -Z_{m,n}, & Z_{m,n} < THR_i; \\ 0 & \end{cases}$$

$$X_{m,n} = \begin{cases} X_{m,n}, & Z_{m,n} > THR_i, \\ 1 \oplus X_{m,n}, & Z_{m,n} < THR_i, \end{cases}$$



■ Рис. 3. Схема работы многопорогового декодера по надежностям

$$Z_n = R_n \oplus \sum_{m \in M(n)} (-1)^{S_{m,l}} Y_{m,n} \text{ min};$$

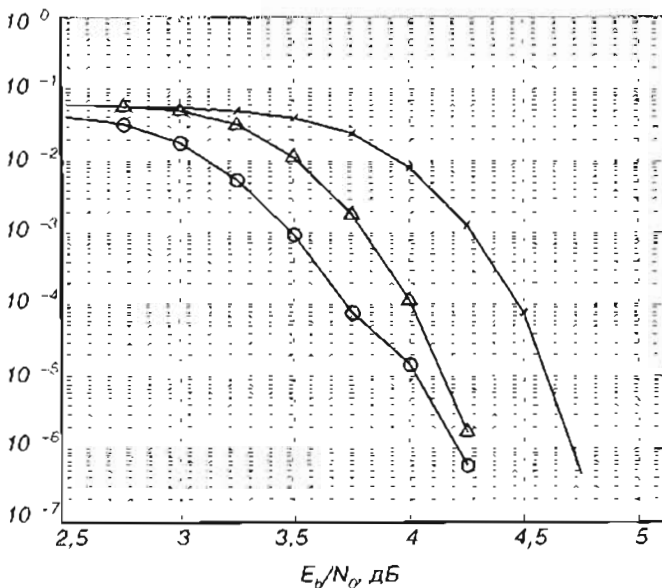
$$C_n = \begin{cases} X_n, Z_n > 0 \\ 1 \oplus X_n, Z_n < 0. \end{cases}$$

6. Декодирование итеративно продолжается до тех пор, пока вектор C не станет кодовым словом или не будет выполнено заданное число итераций.

При многопороговом декодировании, если на первой итерации была исправлена хотя бы одна ошибка, декодирование на последующих итерациях становится значительно проще и общее качество декодирования улучшается [8]. По-прежнему для работы декодеру не требуется информация о шуме в канале, достаточно лишь задать надежности.

Декодер, работающий по многопороговой схеме, позволяет получить вероятность ошибки декодирования на 0,1–0,4 дБ лучшую, чем обеспечивает быстрый декодер по надежности «UMP», практически приближаясь к вероятности ошибки, получаемой при вероятностном декодировании кодов с низкой плотностью проверок на четность. На рис. 4 приведены кривые вероятностей ошибок декодирования (BER) двоичных кодов в канале с аддитивным гауссовским шумом, имеющим математическое ожидание, равное нулю, и энергию шума N_0 при использовании трех различных декодеров: декодера по вероятностям, быстрого декодера по надежности и многопорогового декодера.

Помимо независимости от характеристик канала многопороговый декодер обладает свойством де-



■ Рис. 4. Вероятность ошибки при декодировании (2048, 1649) – кода с низкой плотностью проверок на четность при помощи различных декодеров для различных отношений сигнал/шум в канале: ○ – вероятностный декодер; ● – быстрый декодер; △ – многопороговый декодер

кодеров кодов с низкой плотностью проверок на четность, а именно – универсальность и применимость для любой конструкции таких кодов.

Критерии качества и сложности декодирования

Так как декодирование LDPC является итеративным, то в общем случае сложность декодирования можно представить как произведение сложности одной итерации на общее количество итераций:

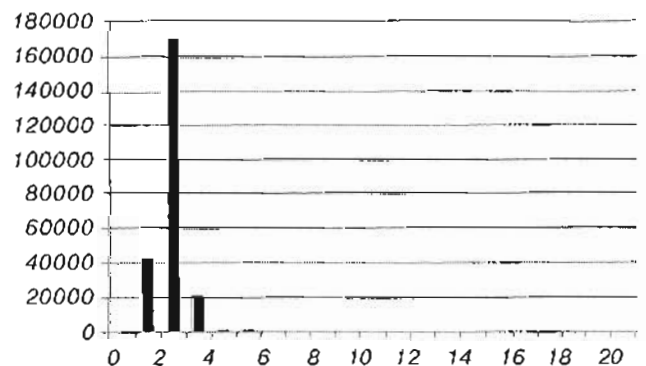
$$C_{complete} = C_{one\ iteration} \cdot N_{iterations}$$

Сложность одной итерации декодирования пропорциональна числу единиц в проверочной матрице [1–3, 7]. Таким образом, чтобы анализировать всю сложность декодирования, необходимо учитывать не только сложность одной итерации декодера, но и количество итераций, требуемых для обеспечения заданного качества декодирования.

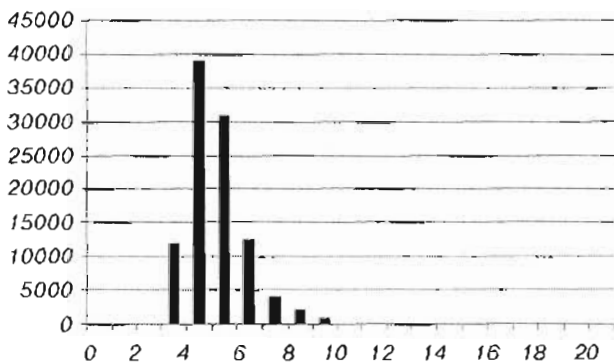
Будем понимать под скоростью сходимости зависимость вероятности ошибки декодирования от количества итераций, за которое данная вероятность ошибки декодирования была получена.

На рис. 5–7 приводятся гистограммы сходимости многопорогового декодера для различных конструкций LDPC кодов из расчета максимального количества итераций, равного 20. На осях гистограмм отложено количество итераций и количество попыток, в которых декодер закончил работу за данное число итераций. Во всех случаях конструкции кодов анализировались на отношении сигнал/шум, обеспечивавшем вероятность ошибки 10^{-5} .

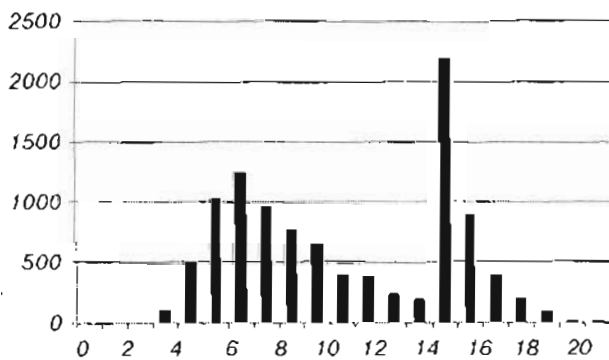
Из гистограмм видно, что требуемое количество итераций не является постоянной величиной для различных конструкций кодов, и наименьшее количество итераций требуется для декодирования Евклидово-геометрических кодов (EG-LDPC). Следовательно, общая сложность декодирования для этой конструкции будет минимальна, несмотря на большее количество единиц в проверочной матрице EG-LDPC кодов.



■ Рис. 5. Число итераций, затраченных на декодирование LDPC кодов, основанных на Евклидовых геометриях [4]



■ Рис. 6. Число итераций, затраченных на декодирование LDPC кодов, основанных на кодах Рида-Соломона с двумя информационными символами [5]



■ Рис. 7. Число итераций, затраченных на декодирование LDPC кодов, построенных по принципу density evolution [6]

Заключение

В данной работе предложен алгоритм многопорогового декодирования для кодов с низкой плотностью проверок на четность. Являясь представителем класса «быстрых» декодеров, алгоритм по

сложности равен алгоритму «UMP», рассмотренному в работе, и требует сложности декодирования в 10 раз меньшей, нежели сложность традиционных алгоритмов. При этом многопороговый алгоритм декодирования позволяет в гауссовском канале получить вероятность ошибки декодирования всего на 0,1–0,2 дБ хуже, чем при использовании локально-оптимальных алгоритмов, а по сравнению с быстрыми алгоритмами декодирования новый алгоритм выигрывает 0,3–0,4 дБ по вероятности ошибки.

Литература

1. Gallager R. G. Low-density parity-check codes. PhD thesis. – 1963. – 90 p.
2. Gallager R. G. Low-density parity-check codes // IEEE Trans. on Inform. Theory. – Vol. IT-8. – Jan, 1968. – P. 21–28.
3. MacKay D. J. C., Neal R. M. Near Shannon limit performance of low density parity check codes // IEE Electronics Letters. – Vol. 32. – N 18. – 29 Aug. 1996. – P. 1645–1655.
4. Djurdjevic I., Xu J., Abdel-Ghaffar K., Lin S. A class of low-density parity-check codes constructed based on Reed-Solomon codes with two information symbols // IEEE Communications Letters. – Vol. 7. – N 7. – July 2003. – P. 317–319.
5. Kou Y., Lin S., Fosserier M. P. C. Low-density parity-check codes based on finite geometries: A rediscovery and new results // IEEE Transactions on Information Theory. – Vol. IT-47. – Nov. – 2001. – P. 2711–2736.
6. Richardson T. J., Shokrollahi M. A., Urbanke R. L. Design of capacity approaching irregular low-density parity-check codes // IEEE Trans. Information Theory. – Vol. 47. – Feb. 2001. – P. 619–637.
7. Fosserier M. P. C., Mihaljevic M., Imai H. Reduced complexity iterative decoding of low-density parity check codes based on belief propagation // IEEE Transactions on Communications. – 47(5). – May 1999. – P. 673–680.
8. Колесник В. Д., Мирончиков Е. Т. Декодирование циклических кодов. – М.: Связь, 1968. – 252 с.

УДК 621.391.251

К ВОПРОСУ О ПОСТРОЕНИИ LDPC – КОДОВ НА ОСНОВЕ ЕВКЛИДОВЫХ ГЕОМЕТРИЙ

А. А. Овчинников,

канд. техн. наук, ассистент

Санкт-Петербургский государственный университет аэрокосмического приборостроения

В статье рассматриваются коды с малой плотностью проверок на четность, основанные на конечных Евклидовых геометриях. Для ряда таких кодов получены оценки их минимального расстояния и спектра, предложены процедуры укорочения Евклидово-геометрических конструкций и оценки расстояния полученных кодов. Приведены результаты моделирования рассматриваемых кодов в канале с аддитивным белым гауссовским шумом (АБГШ).

In this paper we consider the low-density parity-check codes based on finite Euclidean geometries. For the number of such codes the estimations of their minimal distance and spectrum are obtained, the shortening procedures and the estimations of distance for such codes are suggested. The simulation results in channel with AWGN are presented.

Введение

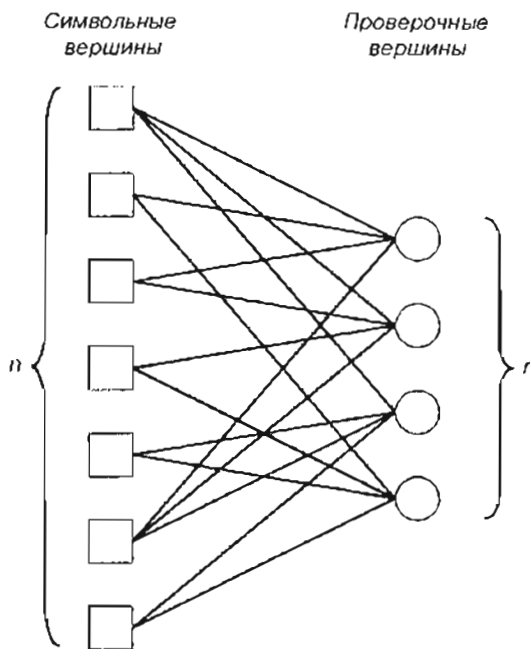
Коды с малой плотностью проверок на четность (LDPC-коды) были впервые предложены Р. Галлагером [1, 2] и позднее исследовались в работах [3–6]. Несмотря на то, что в течение долгого времени LDPC-коды были практически исключены из рассмотрения, в последние годы наблюдается увеличение количества исследований в этой области. Это связано с тем, что, обладая плохим минимальным расстоянием, коды с малой плотностью, тем не менее, обеспечивают высокую степень исправления ошибок при весьма малой сложности их декодирования. Было показано, что с ростом длины некоторые LDPC-коды могут превосходить турбо-коды и приближаться к пропускной способности канала с АБГШ [7]. Вместе с тем, многие предложенные конструкции LDPC-кодов являются циклическими или квазициклическими, что позволяет производить не только быстрое декодирование, но и эффективные процедуры кодирования. Кроме того, даже для LDPC-кодов, не обладающих свойством циклическости, были предложены эффективные процедуры кодирования [8].

Код с малой плотностью проверок на четность задается своей проверочной матрицей H , обладающей свойством разреженности, т. е. ее строки и столбцы содержат мало ненулевых позиций по сравнению с размерностью матрицы. Наравне с традиционным заданием кода как нулевого пространства проверочной матрицы, LDPC-коды ча-

сто задаются с помощью графа, для которого матрица H является матрицей смежности (так называемого графа Таннера). Это двудольный граф, вершины которого делятся на два множества: 1) l символьных вершин, соответствующих столбцам; 2) r проверочных вершин, соответствующих строкам проверочной матрицы. Ребра, соединяющие вершины графа, соответствуют ненулевым позициям в матрице H . Пример такого графа приведен на рис. 1.

LDPC-коды, у которых строки и столбцы содержат одинаковое число единиц, принято называть регулярными кодами, в то время как коды с неравным числом единиц называются нерегулярными. Как правило, построение хороших нерегулярных кодов использует вероятностные методы, анализ таких кодов производится в асимптотике, тогда как регулярные конструкции основаны на объектах (например, комбинаторных) с известными свойствами и могут анализироваться с учетом свойств этих объектов.

Настоящая статья рассматривает построение LDPC-кодов, основанных на Евклидовых геометриях. Евклидово-геометрические коды известны довольно давно [9–11], однако в качестве кодов с малой плотностью они стали рассматриваться только в последние годы [12]. Мы анализируем свойства LDPC-кодов, основанных на Евклидовой геометрии, и предлагаем методы построения новых кодов, используя свойства Евклидовых геометрий и проведенный анализ.



■ Рис. 1. Граф Таннера LDPC-кода

Конструкции и декодирование

Как и для всякого линейного (n, k) -кода, одной из оценок качества LDPC-кода является вероятность ошибочного декодирования, которая обычно характеризуется долей ошибочных бит в декодированном сообщении (BER), при заданных длине кода n и скорости $R = k/n$.

Одним из главных параметров, влияющих на вероятность ошибочного декодирования, является кодовое минимальное расстояние d_0 . В случае LDPC-кодов, однако, часто минимальное расстояние кода мало, и низкая вероятность ошибки достигается за счет хороших спектральных свойств кода (небольшого количества слов малого веса).

Р. Галлагером были предложены алгоритмы декодирования LDPC-кодов как для дискретных (bit-flipping decoding), так и для полунепрерывных (belief propagation decoding) каналов. Общим свойством LDPC-декодеров является то, что они представляют собой итеративные процедуры, оперирующие не с блоками, а с отдельными символами принятого сообщения. Параметром декодера является максимальное число итераций, после которого декодер принимает решение о передававшемся слове. На практике часто бывает достаточно небольшого числа итераций, чтобы правильно декодировать принятое слово. В работе М. С. Пинскера и В. В. Зяблова [3] показано, что сложность декодирования LDPC-кода составляет порядка $n \log n$. В работах [12, 13] рассмотрены ускоренные процедуры декодирования LDPC-кодов, дающие незначительное увеличение вероятности ошибки.

Работа декодера LDPC-кода ухудшается, если в графе Таннера соответствующего LDPC-кода присутствуют циклы небольшой длины. Как правило,

циклы длины 6 не оказывают существенного влияния на качество декодирования, поэтому существующие на сегодня конструкции должны обеспечивать отсутствие циклов длины 4. Для этого достаточно, чтобы любые два столбца проверочной матрицы LDPC-кода не имели более одной общей ненулевой позиции.

В работах [6, 14] проведен асимптотический анализ декодера Галлагера «belief propagation» для некоторых каналов связи. Показано, что при использовании этого декодера существует некий порог, такой, что при определенном уровне помех в канале (превышающем этот порог) вероятность ошибки декодирования не стремится к нулю с ростом числа итераций. Величина этого порога зависит от распределения весов строк и столбцов проверочной матрицы LDPC-кода, и эти веса можно оптимизировать с помощью предлагаемой в работах [6, 14] процедуры «density evolution». Коды с распределениями, полученными с помощью этой процедуры, дают выигрыш на низких отношениях сигнал-шум (в канале с АБГШ), однако, как правило, обладают так называемым эффектом «error-floor», т. е. более медленным уменьшением вероятности ошибки при увеличении отношения сигнал-шум.

В последние годы было предложено много конструкций LDPC-кодов. Некоторые из них основывались на свойствах известных комбинаторных объектов – разностных множеств, блок-схем, геометрий [15, 16], другие – на различного рода вероятностных методах [7, 14]. В последнем параграфе данной работы приведены сравнительные результаты моделирования некоторых из них.

Конечные Евклидовы геометрии

Опишем вкратце Евклидовы геометрии [9, 10]. Приведенные здесь соотношения будут использоваться нами в дальнейшем для оценки параметров получаемых LDPC-кодов.

Евклидовой геометрией EG называется совокупность объектов – точек и прямых, удовлетворяющих следующим аксиомам:

1. Через любые две точки можно провести прямую, причем только одну;
2. Для любой прямой L и любой точки p , не лежащей на L , можно провести прямую, проходящую через p и не пересекающую L (т. е. прямую, параллельную прямой L).
3. Существуют три точки, не лежащие на одной прямой.

Следует отметить, что приведенный набор аксиом не является единственным, с помощью которого задается Евклидова геометрия, однако другие наборы аксиом могут быть сведены к этим трем, и наоборот.

Одной из наиболее часто используемых и практически важных форм задания Евклидовых геометрий является описание их с помощью конечных по-

лей. Евклидова геометрия $EG(m, q)$, где $q = p^s$, p – простое, задается с помощью конечного поля $GF(q^m)$ [расширения поля $GF(q)$] следующим образом: точками Евклидовой геометрии являются элементы поля $\alpha^j \in GF(q^m)$, $j = -\infty, 0, 1, \dots, q^m - 2$, α – примитивный элемент поля $GF(q^m)$. Заметим, что в множество точек Евклидовой геометрии входит и нулевая точка – нулевой элемент $\alpha^{-\infty}$ поля $GF(q^m)$. Тогда линия, проходящая через нулевую точку и некоторую ненулевую точку α^j , задается уравнением

$$L(0, \alpha^j) = \{\beta\alpha^j\} = \{\beta\alpha^j : \beta \in GF(q), \alpha^j \in GF(q^m), \alpha^j \neq 0\}, \quad (1)$$

т. е. проходит через точки $\alpha^{-\infty} = 0$ и $\alpha^j \neq 0$, и содержит элементы поля, получаемые из α^j умножением на все элементы β (включая нулевой) подполя $GF(q)$. Если некоторые элементы α^j и α^l линейно независимы, то есть α^l не лежит на прямой $L(0, \alpha^j)$, то, в соответствии с аксиомой 2, можно провести линию, параллельную линии $L(0, \alpha^j)$ и проходящую через точку α^l :

$$L(\alpha^l, \alpha^j) = \{\alpha^l + \beta\alpha^j\} = \{\alpha^l + \beta\alpha^j : \beta \in GF(q)\}. \quad (2)$$

Так как элемент β в уравнениях (1), (2) принимает $q = p^s$ различных значений, каждая прямая в Евклидовой геометрии содержит

$$\rho = q = p^s \quad (3)$$

точек. Всего существует

$$|L| = q^{m-1}(q^m - 1)/(q - 1) \quad (4)$$

линий в $EG(m, q)$. Каждая линия имеет $q^{m-1} - 1$ параллельных, через каждую точку проходит

$$\gamma = (q^m - 1)/(q - 1) \quad (5)$$

прямых (или, другими словами, в каждой точке пересекается γ прямых).

Евклидова геометрия $EG(2, q)$ называется плоскостью. Точки плоскости могут быть получены как линейные комбинации трех точек $\alpha^j, \alpha^l, \alpha^k$, не лежащих на одной прямой:

$$\{\alpha^j + \mu\alpha^l + \eta\alpha^k\}, \mu, \eta \in GF(q). \quad (6)$$

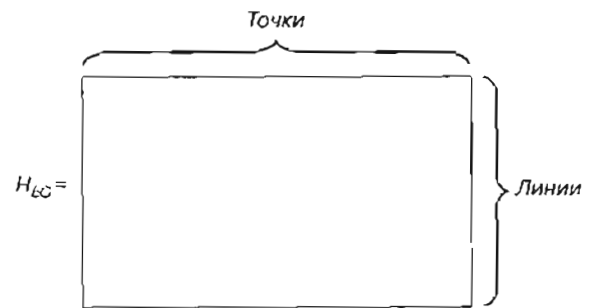
Из соотношений (4) и (6) следует, что плоскость содержит q^2 точек и $q(q + 1)$ прямых.

Теперь рассмотрим способы задания кодов, основанных на конечных Евклидовых геометриях.

Коды EG-LDPC

Евклидово-геометрические коды строятся как система инцидентий геометрии $EG(m, q)$ [9, 11, 17]. Так как число единиц в проверочной матрице Евклидово-геометрического кода мало по сравнению с размерами матрицы, такой код можно рассматривать как LDPC-код.

LDPC-код, основанный на Евклидовой геометрии, с проверочной матрицей H_{EG} , строится следующим образом: строки проверочной матрицы



■ Рис. 2. Проверочная матрица EG-кода

соответствуют линиям Евклидовой геометрии, столбцы – ненулевым точкам в $EG(m, p^s)$. Элементы матрицы H_{EG} определяются из векторов инцидентий линий Евклидовой геометрии (рис. 2):

$$H_{EG}(i, j) = \begin{cases} 1, & \text{если точка } j \text{ лежит на прямой } i \\ 0, & \text{в противном случае.} \end{cases} \quad (7)$$

Из уравнений (1)–(5) следует, что проверочная матрица H_{EG} имеет

$$n = q^m \quad (8)$$

столбцов и

$$r = q^{m-1}(q^m - 1)/(q - 1) \quad (9)$$

строк. Каждый столбец матрицы содержит

$$\gamma = (q^m - 1)/(q - 1) \quad (10)$$

единиц, каждая строка содержит

$$\rho = p^s \quad (11)$$

единиц.

Обычно рассматриваются Евклидово-геометрические коды при $p = 2$, не содержащие нулевой точки [12, 18]. Такие коды иногда называются EG-кодами типа 0, они являются циклическими [9, 11], их параметры

$$n = 2^{ms} - 1; \quad (12)$$

$$r = (2^{(m-1)s} - 1)(2^{ms} - 1)/(2^s - 1). \quad (13)$$

Число информационных символов таких кодов оценено в работе [19].

Наряду с заданием проверочной матрицы, как показано на рис. 2, можно рассматривать EG-LDPC код с матрицей, транспонированной к матрице (7). Тогда ее строки соответствуют точкам геометрии, а столбцы – линиям. В обоих случаях задания матрицы свойства геометрии и (8)–(11) обеспечивают выполнение следующих свойств проверочной матрицы.

1. Каждая строка содержит ρ единиц [следует из (11)].
2. Каждый столбец содержит γ единиц [следует из (10)].
3. Любые два столбца имеют не более чем одну общую ненулевую позицию (так как через две точки можно провести только одну прямую).

■ Таблица 1. Параметры некоторых EG-LDPC кодов (параметр d_0 соответствует нижней оценке минимального расстояния)

n	k	A	d_0	γ	ρ	$EG(m, q)$	Тип
16	7	0,4375	6	5	4	(2,4)	
64	37	0,5781	10	9	8	(2,8)	
256	175	0,6836	18	17	16	(2,16)	
1024	781	0,7627	34	33	32	(2,32)	
64	13	0,2031	22	21	4	(3,4)	
512	139	0,2715	74	73	8	(3,8)	
20	11	0,55	5	4	5	(2,4)	EG
72	45	0,625	9	8	9	(2,8)	EG
272	191	0,7022	17	16	17	(2,16)	EG
1056	813	0,7699	33	32	33	(2,32)	EG
336	285	0,8482	5	4	21	(3,4)	EG
4672	4299	0,9202	9	8	73	(3,8)	EG
90	9	0,1	9	8	73	(2,9)	EG
756	27	0,0357	9	8	73	(2,27)	EG
7371	6642	0,9011	10	9	91	(3,9)	EG

4. Любые две строки имеют не более одной общей ненулевой позиции (так как две прямые пересекаются не более чем в одной точке).

Свойства 3 и 4 означают, что граф Таннера как для кода с проверочной матрицей H_{EG} , так и для кода с проверочной матрицей H_{EG}^T не имеет циклов длины 4.

Параметры некоторых EG-LDPC кодов (включающих в себя точку 0) приведены в табл. 1. Параметры кодов с проверочной матрицей H_{EG}^T помечены как EG^T . Результаты моделирования EG-LDPC кодов в канале с АБГШ приведены в последнем параграфе.

Так как столбцы проверочной матрицы (7) имеют не более одной общей ненулевой позиции, любые γ столбцов матрицы линейно независимы и, значит, не могут образовать нулевой синдром. Тогда минимальное расстояние кода с проверочной матрицей (7) оценивается как

$$d_0 \geq \gamma + 1. \quad (14)$$

Далее мы получим более точные оценки минимального расстояния EG-кодов.

Укорочение EG-LDPC кодов

В исследованиях [12, 20] приведены некоторые методы укорочения EG-LDPC кодов. С помощью моделирования показано, что вероятность ошибки при укорочении может уменьшаться, однако не приводится аналитических обоснований для выбора того или иного метода укорочения.

Рассмотрим методы укорочения LDPC-кодов и проведем анализ минимального расстояния некоторых Евклидово-геометрических кодов.

Рассмотрим Евклидово-геометрические коды, проверочная матрица которых является транспонированной к матрице (7):

$$H_{EG}(i, j) = \begin{cases} 1, & \text{если точка } i \text{ лежит на прямой } j \\ 0, & \text{в противном случае.} \end{cases} \quad (15)$$

Рассмотрим Евклидово-геометрическое пространство, т. е. геометрии $EG(3, q)$, $q = p^s$. Такие коды имеют длину

$$n = q^2(q^3 - 1)/(q - 1), \quad (16)$$

их проверочная матрица H состоит из

$$r_H = q^3 \quad (17)$$

строк. Заметим, что H не обязательно имеет полный ранг, поэтому r_H может использоваться только как верхняя оценка числа проверочных символов.

Рассмотрим прямую в такой Евклидовой геометрии. Прямая содержит q точек. Через каждую точку, не лежащую на прямой, можно провести единственную прямую, параллельную данной. Каждая такая прямая также содержит q точек. Так как всего в геометрии q^3 точек, всего существует q^2 прямых, параллельных друг другу. Назовем построенное таким образом множество прямых параллельным классом. Геометрия содержит $q^2(q^3 - 1)/(q - 1)$ прямых, которые могут быть разбиты на $(q^3 - 1)/(q - 1)$ параллельных классов по q^2 прямых в каждом. Укорачивая проверочную матрицу EG-кода на столбцы, соответствующие параллельным классам, мы будем получать код, в котором число единиц в строках и столбцах остается равным, так как каждая точка геометрии присутствует в параллельном классе ровно

1000	0010	0001	0010	0010
1000	1000	1000	1000	1000
0100	0010	0100	0100	1000
0100	0001	0001	1000	0100
0010	0001	0010	0010	1000
0100	1000	0010	0001	0010
1000	0001	0100	0001	0001
0001	0010	0010	1000	0001
0001	0100	0001	0001	1000
0100	0100	1000	0010	0001
0001	0001	1000	0100	0010
1000	0100	0010	0100	0100
0010	0010	1000	0001	0100
0010	1000	0001	0100	0001
0001	1000	0100	0010	0100
0010	0100	0100	1000	0010

Рис. 3. Разбиение на параллельные классы для плоскости EG(2,4)

один раз. При этом расстояние кода не ухудшилось, а число единиц в строке стало меньшим, что может улучшить работу итеративного декодера.

Описанный метод укорочения позволяет оптимизировать параметры кодов для требуемых длин, выбирая параметры геометрий и величину укорочения с учетом получающихся весов строк.

Попробуем оценить, как укорочение на параллельные классы влияет на дистанционные характеристики кода. Для этого рассмотрим Евклидову плоскость, т. е. геометрию EG(2, q). Плоскость содержит q(q+1) прямых, которые могут быть разбиты на q+1 параллельных классов по q прямым в каждом. Каждая точка плоскости присутствует в параллельном классе ровно один раз. Пример разбиения плоскости на параллельные классы P₁, ..., P₅ приведен на рис. 3 для плоскости EG(2, 2²).

Рассмотрим отдельно случаи p=2 и p≠2. Пусть p≠2. Далее, пусть

$$W(x, y) = \sum_{i=0}^n A_i x^{n-i} y^i \quad (18)$$

– весовая функция кода [10], где A_i – число слов веса i в коде; x – число нулей; y – число единиц. Тогда справедлива следующая теорема.

Теорема 1. Если проверочная матрица (15) Евклидово-геометрического кода при m=2, q=p^s, p≠2 имеет полный ранг, тогда коэффициент A_i весовой функции (18) вычисляется как

$$A_i = \begin{cases} C_{q+1}^{i/q}, & i:2q \\ 0, & \text{в противном случае.} \end{cases} \quad (19)$$

Доказательство. Если ранг матрицы (15) является полным, т. е. равен q², то число информационных символов кода равно

$$k = n - r = q(q+1) - q^2 = q. \quad (20)$$

Разобьем множество прямых плоскости на q+1 параллельных классов по q прямым в каждом. Заметим, что при p≠2 число q всегда нечетно, тогда как число q+1 всегда четно. Так как все точки присутствуют в параллельном классе ровно один раз, сумма всех столбцов проверочной матрицы, соответствующих параллельному классу, дает столбец из всех единиц. Тогда очевидно, что сумма всех столбцов четного числа параллельных классов даст нулевой столбец, т. е. задаст кодовое слово. Заметим, что, так как q+1 четно, максимальное четное число классов равно q+1, что соответствует кодовому слову из всех единиц. Четное число 2i параллельных классов можно выбрать из Евклидова пространства C_{q+1}²ⁱ способами. Таким образом, общее количество кодовых слов, образованных параллельными классами, равно

$$C_{q+1}^0 + C_{q+1}^2 + C_{q+1}^4 + C_{q+1}^6 + \dots + C_{q+1}^{q+1} = \sum_{i=0}^{(q+1)/2} C_{q+1}^{2i} \quad (21)$$

Как следствие биномиальной теоремы [21], имеем следующие тождества:

$$\sum_{i=0}^n C_n^i = 2^n; \quad (22)$$

$$\sum_{i=0}^n (-1)^i C_n^i = 0. \quad (23)$$

Пусть n четно. Тогда из тождества (23) имеем

$$\sum_{i=0}^{n/2} C_n^{2i} = \sum_{i=0}^{n/2-1} C_n^{2i+1},$$

т. е. суммы четных и нечетных членов ряда C_nⁱ равны. Тогда с учетом тождества (22)

$$\sum_{i=0}^{n/2} C_n^{2i} = 2^{n-1}. \quad (24)$$

Из равенств (21) и (24) получаем

$$\sum_{i=0}^{(q+1)/2} C_{q+1}^{2i} = 2^q. \quad (25)$$

Но из (20) следует, что в (25) учтены все кодовые слова. Отсюда следует утверждение теоремы.

Таким образом, пользуясь теоремой 1, можно указать минимальное расстояние для рассматриваемых кодов.

Следствие 1. Минимальное расстояние кода из теоремы 1 равно

$$d_0 = 2q. \quad (26)$$

Для каждого конкретного кода ранг проверочной матрицы может быть вычислен экспериментально. Проведенные тесты показывают, что для рассматривавшихся параметров геометрий при m=2, p≠2

■ Таблица 2. Точное минимальное расстояние некоторых EG-LDPC кодов

n	k	A	\tilde{d}_0	d_0	γ	p	EG(m, ϕ)	Тип
20	11	0,55	5	5	4	5	(2,4)	EG
72	45	0,625	9	9	8	9	(2,8)	EG
272	191	0,7022	17	17	16	17	(2,16)	EG
1056	813	0,7699	33	33	32	33	(2,32)	EG
336	285	0,8482	5	5	4	21	(3,4)	EG
4672	4299	0,9202	9	9	8	73	(3,8)	
90	9	0,1	9	18	8	73	(2,9)	EG
756	27	0,0357	9	54	8	73	(2,27)	EG
7371	6642	0,9011	10	18	9	91	(3,9)	EG

ранг проверочной матрицы действительно является полным.

В табл. 2 приведены параметры некоторых LDPC-кодов, с их минимальными расстояниями, полученными на основе результатов теоремы 1. Здесь \tilde{d}_0 означает оценку минимального расстояния, d_0 – точное минимальное расстояние.

Теперь рассмотрим случай $p = 2$. Для поля характеристики 2 соотношение (20) не выполняется, т. е. проверочная матрица (15) содержит линейно зависимые строки. Таким образом, слова, образованные параллельными классами, не являются всеми кодовыми словами. В этом случае можно сформулировать следующие утверждения.

Теорема 2. Если код с проверочной матрицей (15) при $m = 2$, $p = 2$ имеет минимальное расстояние $q + 1$, тогда для слова веса $q + 1$ никакие две из линий, соответствующих ненулевым позициям этого слова, не лежат в одном параллельном классе.

Доказательство. Минимальное расстояние кода определяется минимальной линейной комбинацией столбцов проверочной матрицы, дающей в результате нулевой столбец-синдром. В случае рассматриваемых кодов столбец проверочной матрицы соответствует линии в Евклидовой геометрии. Каждая позиция синдрома соответствует точке Евклидовой плоскости и является суммой позиций столбцов, вошедших в линейную комбинацию. Таким образом, чтобы позиция синдрома равнялась нулю, необходимо, чтобы через соответствующую точку либо не проходили прямые из данной линейной комбинации, либо число прямых, проходящих через точку, было четным.

Следовательно, чтобы построить кодовое слово, нужно найти множество прямых L , такое, что через каждую точку геометрии либо не проходят прямые из L , либо в этой точке пересекается четное число прямых из L .

Тогда нахождению слова минимального веса соответствует нахождение множества L_0 минимальной мощности. Попробуем построить такое множество минимальной мощности.

Допустим, что на некотором шаге k уже сформировано множество прямых $L_0^{(k)}$. Обозначим P^+ те точки из $L_0^{(k)}$, через которые проходит нечетное число прямых, P^- – те точки, через которые проходит четное число прямых. Чтобы число прямых в L_0 было минимальным, необходимо, чтобы каждая следующая добавляемая прямая проходила через как можно большее число точек P^+ , не проходила через точки P^- и добавляла как можно меньшее число новых точек, которые на следующем шаге увеличат множество P^+ . Построение L_0 закончится тогда, когда множество точек P^+ станет пустым.

Это эквивалентно тому, что на шаге k новая прямая должна пересекать как можно большее число прямых, уже содержащихся в $L_0^{(k-1)}$, причем пересечение должно идти по точкам P^+ .

Рассмотрим прямую $l^{(0)}$ Евклидовой плоскости. Эта прямая содержит q точек и принадлежит какому-то классу параллельности. Следующая проведенная прямая $l^{(1)}$ может либо не пересечь данную, если она принадлежит тому же классу параллельности, что и $l^{(0)}$, либо пересечь в одной точке, если $l^{(1)}$ не параллельна $l^{(0)}$. Следующая проводимая прямая $l^{(2)}$ может пересечь либо одну прямую из уже выбранных, если она параллельна одной из них, либо пересечь обе, в противном случае.

Пример построения множества L_0 для случая $q = 4$ приведен на рис. 4. Точки множества P^+ обведены на этом рисунке кругами (кроме заключительной фигуры).

Таким образом, выбирая каждый раз прямую из еще не использовавшегося класса параллельности, мы будем пересекать все прямые, выбранные на предыдущих шагах. Добавление прямой к множеству $L_0^{(k-1)}$, в котором уже содержатся k прямых, добавляет $q - k$ новых точек. Всего в $L_0^{(k)}$ содержится

$$N_k = |L_0^{(k)}| = \sum_{i=0}^{k-1} (q - i), \quad k \geq 0 \quad (27)$$

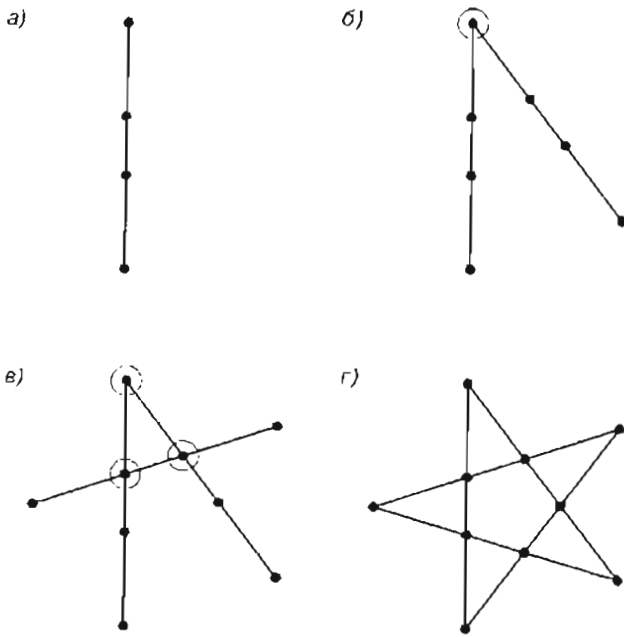


Рис. 4. Графическая интерпретация построения вектора минимального веса для EG(2,4): а – начальная линия; б – добавление второй линии; в – добавление третьей линии; г – замкнутая фигура

точек. Оценим число N_k^* точек типа P^* на k -м шаге. На нулевом шаге $L_0^{(0)}$ содержит единственную прямую, и все $N_0 = q$ точек являются P^* -точками. На первом шаге добавляемая прямая пересекает уже выбранную в одной точке, таким образом, в $L_0^{(1)}$ содержится одна точка типа P^{**} и $N_1 - 1$ точек типа P^* . На втором шаге новая прямая пересекает две существующие еще в двух точках, и таким образом, в $L_0^{(2)}$ содержится уже три точки типа P^{**} и $N_2 - 3$ точек типа P^* . В общем случае

$$N_k^* = N_k - \sum_{i=0}^{k-1} i, \quad k \geq 0 \quad (28)$$

или, с учетом (27),

$$N_k^* = \sum_{i=0}^{k-1} (q - i) - \sum_{i=0}^{k-1} i = \sum_{i=0}^{k-1} (q - 2i). \quad (29)$$

Теперь найдем минимальное число линий, которые нужно провести, чтобы получить кодовое слово, т. е. найдем $|L_0|$. Для этого нужно просто определить номер шага k , при котором (29) обратится в ноль:

$$|L_0| = \{k : N_k^* = 0\}, \quad k \geq 0. \quad (30)$$

Применив формулу суммы арифметической прогрессии, из соотношений (29) и (30) получим условие остановки построения L_0 :

$$\sum_{i=0}^{k-1} (q - 2i) = \frac{(q + q - 2k + 2)k}{2} = (q - k + 1)k = 0. \quad (31)$$

При условии $k > 0$ выполнение условия (31) возможно только при $k = q$. Таким образом, при

указанном методе построения требуется $q + 1$ прямых из разных классов параллельности, чтобы построить L_0 , а в Евклидовой плоскости содержится как раз $q + 1$ классов параллельности. Тогда из (30) и (31) имеем

$$|L_0| = q + 1,$$

что и завершает доказательство.

Однако в отношении теоремы 2 можно сформулировать следующее утверждение.

Замечание 1. Минимальное расстояние кода из теоремы 2 равно $q + 1$ только в том случае, если можно провести прямые указанным способом, т. е. только через точки типа P^* . Это утверждение не является доказанным, однако эксперименты показывают, что для полей характеристики 2 это действительно так. Таким образом, возможно, теорема 2 задает точное минимальное расстояние.

Следствие 2. Укорочение кода (15) при $m = 2$, $p = 2$ и с учетом замечания 1 на прямые, содержащие любой параллельный класс, приводит к коду с минимальным расстоянием

$$d_0 \geq q + 2. \quad (32)$$

Доказательство. Как было показано в доказательстве теоремы 2, для построения L_0 требуется $q + 1$ прямых из разных классов параллельности, которых в Евклидовой плоскости ровно $q + 1$. Таким образом, любое ненулевое кодовое слово минимального веса имеет $q + 1$ ненулевых позиций, по одной в каждом классе параллельности. Удаление любого класса параллельности приведет к тому, что в коде не останется слов веса $q + 1$, откуда следует утверждение следствия.

Сравнение конструкций в канале с АБГШ

Здесь мы приводим результаты моделирования для кодов, полученных с помощью укорочения на

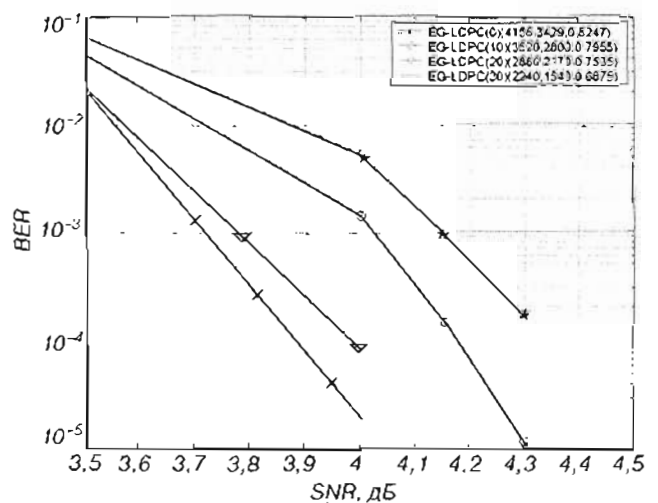


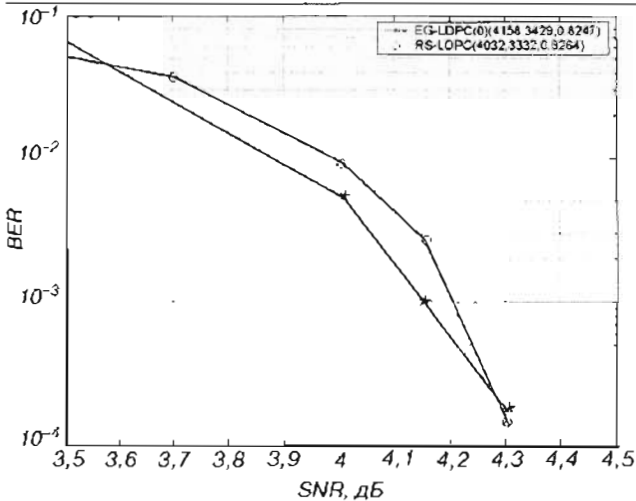
Рис. 5. Евклидово-геометрический код из EG(2, 2ⁱ) и его укорочения:

- EG-LDPC(0)(4158, 3429, 0.8247);
- EG-LDPC(10)(3520, 2800, 0.7955);
- EG-LDPC(20)(2880, 2170, 0.7535);
- EG-LDPC(30)(2240, 1540, 0.6875)

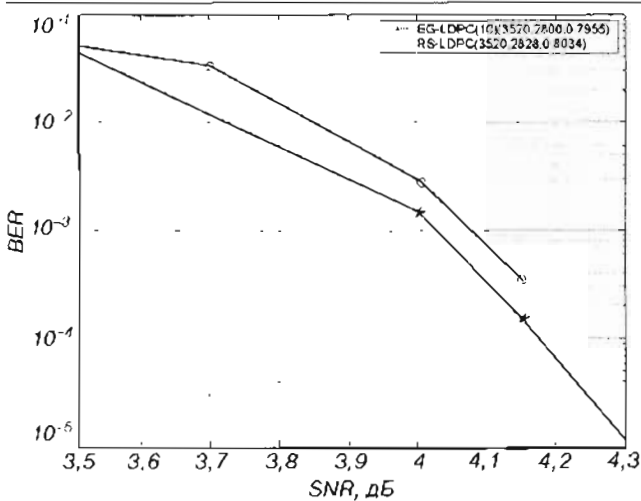
параллельные классы, описанного выше. Моделирование проводилось в канале с аддитивным белым гауссовским шумом (АБГШ), двоичной фазовой модуляцией, для декодирования использовался ускоренный декодер, описанный в работе [13], с ограничением максимального числа итераций 10.

В качестве исходного кода рассматривался Евклидово-геометрический код с проверочной матрицей (15), полученной с помощью конечной геометрии EG (2, 2⁶). Укоротим этот код на 10, 20 и 30 параллельных классов, что приведет к кодам с разными длинами и скоростями. Результаты их моделирования в канале с АБГШ показаны на рис. 5.

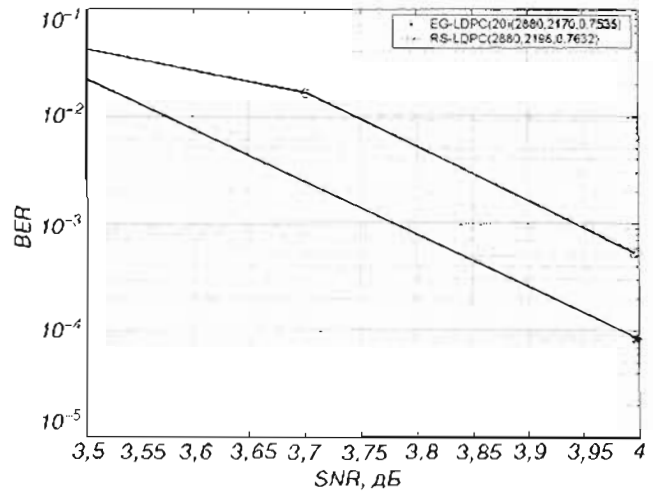
Для сравнения корректирующей способности полученных кодов с другими кодами рассмотрим конструкцию RS-LDPC, основанную на укороченных кодах Рида-Соломона [22]. На рис. 6–9 приведены результаты моделирования укороченных Евклидово-геометрических кодов и кодов RS-LDPC. Как видно



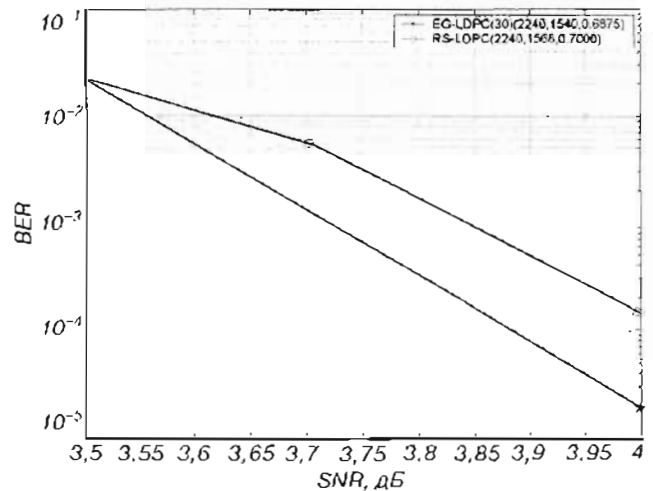
■ Рис. 6. Евклидово-геометрический код из EG (2, 2⁶) и код RS-LDPC (6, 63, 35):
 * EG-LDPC(0)(4158, 3429, 0.8247);
 ○ RS-LDPC(4032, 3332, 0.8264)



■ Рис. 7. Укорочение кода EG на 10 параллельных классов и код RS-LDPC (6, 55, 35):
 * EG-LDPC(10)(3520, 2800, 0.7955);
 ○ RS-LDPC(3520, 2828, 0.8034)



■ Рис. 8. Укорочение кода EG на 20 параллельных классов и код RS-LDPC (6, 45, 35):
 * EG-LDPC(20)(2880, 2170, 0.7535);
 ○ RS-LDPC(2880, 2198, 0.7632)



■ Рис. 9. Укорочение кода EG на 30 параллельных классов и код RS-LDPC (6, 35, 35):
 * EG-LDPC(30)(2240, 1540, 0.6875);
 ○ RS-LDPC(2240, 1568, 0.7000)

из графиков, исходный Евклидово-геометрический код и его укорочения дают выигрыш по сравнению с кодом RS-LDPC при сравнимых длинах и кодовых скоростях. Таким образом, предложенный метод построения укороченных Евклидово-геометрических кодов дает способ получения новых эффективных кодов для различных скоростей и кодовых длин.

Заключение

В данной статье рассмотрены свойства Евклидово-геометрических кодов. Для ряда EG-кодов получено описание их спектра, найдены более точные оценки минимального расстояния некоторых EG-кодов и их укорочений. Предложен метод построения LDPC-кодов, основанный на укорочении на параллельные классы. Данный метод позволяет более гибко задавать такие параметры кода, как длина и скорость. Проведенное моделирование в канале с АБГШ подтверждает эффективность кодов, полученных таким способом.

Литература

1. **Gallager R. G.** Low-density parity check codes // IRE Transactions information theory. – Jan. 1962.
2. **Gallager R. G.** Low density parity check codes. – Cambridge, MA: MIT Press, 1963.
3. **Зяблов В. В., Пинскер М. С.** Оценка сложности исправления ошибок низкоплотными кодами Галлагера // Проблемы передачи информации. – Vol. XI – N 1, 1975.
4. **MacKay D., Neal R. M.** Near shannon limit performance of low-density parity-check codes // IEEE Transactions on Information Theory. – Vol. 47. – Feb. 2001.
5. **MacKay D.** Good error correcting codes based on very sparse matrices // IEEE Transactions on information theory. – Vol. 45. – Mar. 1999.
6. **Richardson T. J., Urbanke R. L.** The capacity of low-density parity-check codes under message-passing decoding // IEEE transactions on information theory. – Vol. 47. – Feb. 2001.
7. **Forney G., Richardson T. J., Urbanke R. L., Chung S. Y.** On the design of low-density parity-check codes within 0.0045 db of the shannon limit // IEEE communications letters. – Vol. 5. – Feb. 2001.
8. **Richardson T. J., Urbanke R. L.** Efficient encoding of low-density parity-check codes // IEEE transactions on information theory. – Vol. 47. – Feb. 2001.
9. **Питерсон У., Уэлдон Э.** Коды, исправляющие ошибки. – М.: Мир, 1976.
10. **Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А.** Теория кодов, исправляющих ошибки. – М.: Связь, 1979.
11. **Блейхут Р.** Теория и практика кодов, контролирующихся ошибки. – М.: Мир, 1986.
12. **Kou Y., Lin S., Fossorier P. C.** Low-density parity-check codes based on finite geometries: A rediscovery and new results // IEEE transactions on information theory. – Vol. 47. – Nov. 2001.
13. **Fossorier M. P. C., Mihaljevic M., Imai H.** Reduced complexity iterative decoding of low-density parity-check codes based on belief propagation // IEEE transactions on communications. – Vol. 47. – May 1999.
14. **Richardson T. J., Urbanke R. L., Shokrollahi M.** Design of capacity-approaching irregular low-density parity-check codes // IEEE transactions on information theory. – Vol. 47. – Feb. 2001.
15. **Johnson S. J., Weller S. R.** Regular low-density parity-check codes from combinatorial designs // In Proc. IEEE Information Theory Workshop (Cairns, Australia). – Sept. 2001.
16. **Johnson S. J., Weller S. R.** Codes for iterative decoding from partial geometries // ISIT2002, submitted.
17. **Касами Т., Токура Н., Ивадари Е., Инагаки Я.** Теория кодирования. – М.: Мир, 1978.
18. **Lin S.** Shortened finite geometry codes // IEEE transactions on information theory. – Sept. 1972. – P. 692.
19. **Lin S.** On the number of information symbols in polynomial codes // IEEE transactions on information theory. – Vol. 18. – Nov. 1972. – P. 785–794.
20. **Lin S.** Shortened finite geometry codes // IEEE transactions on information theory. – Vol. 18. – Sept. 1972. – P. 692–696.
21. **Грэхем Р., Кнут Д., Паташник О.** Конкретная математика. Основание информатики. – М.: Мир, 1998.
22. **Djurdjevic I., Xu J., Abdel-Ghaffar K., Lin S.** A class of low-density parity-check codes constructed based on reed-solomon codes with two information symbols // IEEE communications letters. – Vol. 7. – July 2003.

УДК 20.53.23; 49.31.00

АДАПТИВНАЯ ПЕРЕДАЧА В МНОГОПОЛЬЗОВАТЕЛЬСКИХ МНОГОЧАСТОТНЫХ СИСТЕМАХ ВЕЩАНИЯ

П. В. Трифонов,
аспирант

Санкт-Петербургский государственный политехнический университет

Рассматривается задача оптимизации схемы передачи в цифровых многопользовательских системах вещания. Показано, что путем реализации управляемого разделения канала может быть достигнуто существенное снижение мощности передатчика, требуемой для достижения заданных параметров системы. Рассматриваются различные варианты реализации этого подхода. Исследуется влияние стохастических свойств радиоканала на эффективность метода.

Optimization of data transmission scheme in digital multi-user multi-carrier broadcast systems is considered. It is shown that optimization of channel sharing can provide considerable reduction of transmitter power required for achieving the target system parameters. Different approaches for this are considered. Impact of stochastic channel variations on the performance of the suggested method is analyzed.

Введение

В настоящее время наблюдается бурное развитие систем мобильной цифровой радиосвязи. Большинство подобных систем характеризуется необходимостью обеспечения высокоскоростной передачи данных от базовой станции к пользовательским терминалам и сравнительно низкой скорости передачи в обратном направлении. Так как базовые станции цифровых радиосетей в большинстве случаев представляют собой достаточно высокопроизводительные вычислительные устройства, возникает возможность реализации адаптивных алгоритмов, позволяющих подстроить используемую схему передачи под текущее состояние радиоканала и требования пользователей, что дает возможность существенно снизить мощность радиосигнала, требуемую для достижения заданного качества обслуживания пользователей.

Многоканальная система вещания

Современные цифровые радиосистемы, как правило, характеризуются достаточно широкой полосой сигнала. Следствием этого является сильная межсимвольная интерференция. Один из широко распространенных способов ее преодоления – многочастотная передача, состоящая в том, что исходный широкополосный канал разбивается на большое число узкополосных подканалов, свобод-

ных от межсимвольной интерференции. Как правило, несущие для этих подканалов выбираются ортогональными, что позволяет при выполнении определенных условий избежать межканальной интерференции. Такой подход получил название ортогонального частотного мультиплексирования (orthogonal frequency division multiplexing – OFDM) [1].

Рассмотрим систему вещания на основе OFDM. Несложно показать, что сигнал, принятый каждым из пользователей подобной системы, равен

$$r_{ki}^{(j)} = \mu_{ki}^{(j)} s_i^{(j)} + \eta_{ki}^{(j)}, \quad k = 1, \dots, K, \quad i = 1, \dots, N,$$

где $r_{ki}^{(j)}$ – сигнал, принятый k -м пользователем по i -му подканалу в момент времени j ; $s_i^{(j)}$ – сигнал, переданный базовой станцией; $\mu_{ki}^{(j)}$ – передаточный коэффициент, характеризующий состояние i -го подканала k -го пользователя в момент времени j ; $\eta_{ki}^{(j)}$ – отсчеты аддитивного белого гауссовского шума с дисперсией σ^2 .

Подканалы удобно характеризовать отношением канал–шум, определяемым как $\xi_{ki}^{(j)} = \frac{|\mu_{ki}^{(j)}|^2}{\sigma^2}$. В дальнейшем, если состояние канала предполагается неизменным, индекс j будет опускаться. Задачей базовой станции является отображение данных, предназначенных каждому из пользователей, на передаваемые сигналы $s_i^{(j)}$, а задачей каждого пользователя является выделение предназначен-

ной ему информации из принятых сигналов $r_{kl}^{(j)}$. Основной задачей является построение такого отображения, для которого существовала бы эффективная процедура разделения, т. е. обеспечение множественного доступа. Среди наиболее распространенных методов множественного доступа можно выделить временное, частотное, пространственное и кодовое разделение. В данной работе рассматривается кодовое разделение. Причины этого будут пояснены ниже. При этом существуют различные способы отображения наложенного сигнала на подканалы [2]. В общем случае, подканалы могут быть объединены в группы и разделение канала может быть реализовано на уровне отдельных групп. Не ограничивая общности, предположим, что передаваемый сигнал может быть представлен как

$$s_{qS_r+s}^{(jS_r+m)} = \sum_{l=1}^{S_r S_t} \dot{s}_{ql}^{(j)} a_{l,S_r+s}^{(j)}, \quad s = 0, \dots, S_r - 1, \\ m = 0, \dots, S_t - 1, \quad q = 0, \dots, N/S_r - 1,$$

где S_r – коэффициент расширения в частотной области; S_t – коэффициент расширения во временной области; $S = S_r S_t$ – полный коэффициент расширения; $a_l = (a_{l,0}, \dots, a_{l,S-1})$ – l -я расширяющая последовательность; $\dot{s}_{ql}^{(j)}$ – модулированный сигнал, предназначенный l -му пользователю, использующему q -ю группу подканалов.

Предположим, что используются S ортогональных расширяющих последовательностей. Необходимо отметить, что в рассматриваемой системе расширяющие последовательности присваиваются не отдельным пользователям, а подканалам (или их группам). Это позволяет рассматривать каждый подканал (группу подканалов) как набор логических каналов, которые могут быть произвольным образом распределены между пользователями.

Пусть $\rho_{kq} \in \{0, \frac{1}{S}, \dots, \frac{S-1}{S}, 1\}$ – доля q -го подканала (группы подканалов), выделенная для передачи данных k -го пользователя; $\xi_{kq}^{(j)}$ – среднее (геометрическое) отношение канал-шум по q -й группе подканалов.

Допустим, что в каждом из логических каналов используется своя схема передачи со скоростью c_{kl} и средней мощностью сигнала

$$V_{kl}^2 = M[|\dot{s}_{ql}^{(j)}|^2], \quad k = k(q, l),$$

где $k(q, l)$ – номер пользователя, использующего l -й логический канал q -го физического подканала OFDM-системы (или их группы).

Здесь предполагается, что группы (при $S_r > 1$) образованы путем объединения смежных подканалов. В принципе, это ограничение может быть снято путем введения частотного перемежения (или, что эквивалентно, путем перенумерования коэффициентов $\mu_{kl}^{(j)}$), которое также может быть оптимизи-

ровано. Но, как будет показано ниже, даже такой простой подход обеспечивает достаточно хорошие результаты.

Адаптивная передача

Оптимизационная задача. Предположим, что для каждого из пользователей необходимо обеспечить некоторую фиксированную вероятность ошибки передачи. Ясно, что она зависит как от скорости передачи данных, так и от отношения сигнал-шум. В большинстве случаев эту зависимость можно приближенно представить как

$$f(c_{kq}) = \frac{2^{c_{kq}} - 1}{\Gamma}, \quad c_{kq} \geq 0,$$

где Γ зависит от требуемой вероятности ошибки. Как будет показано ниже, в случае наличия временных флуктуаций канала эта функция оказывается зависимой от состояния канала в начальный момент времени $\xi_{kq}^{(0)}$. Это дает возможность указать среднюю мощность сигнала, необходимую для обеспечения

заданной вероятности ошибки, как $V_{kq}^2 = \frac{f(c_{kq})}{\xi_{kq}}$ или $V_{kq}^2 = \frac{f(\xi_{kq}^{(0)}, c_{kq})}{\xi_{kq}^{(0)}}$.

Предположим, что каждый пользователь должен осуществлять передачу данных со скоростью R_k . Рассмотрим задачу нахождения такого распределения скоростей c_{kq} по подканалам, которое минимизировало бы общую требуемую мощность сигнала, т. е.

$$\min_{c_{kq}, \rho_{kq}} \sum_{k=1}^K \sum_{q=1}^{N/S_r} \frac{\rho_{kq} f(c_{kq})}{\xi_{kq}}$$

при условиях

$$\sum_{k=1}^K \rho_{kq} = 1; \\ \sum_{q=1}^{N/S_r} \rho_{kq} c_{kq} = R_k.$$

Применяя классическую теорию условного экстремума, можно получить следующую систему уравнений:

$$\begin{cases} 0 = (\beta_q^{(k)} - \beta_q) \rho_{kq} \\ R_k = \sum_{q=1}^{N/S_r} \rho_{kq} f^{-1}(\lambda_k \xi_{kq}) \\ 1 = \sum_{k=1}^K \rho_{kq} \\ \beta_q \leq \beta_q^{(k)} = \frac{(f^{-1}(\lambda_k \xi_{kq})) - \lambda_k \xi_{kq} f^{-1}(\lambda_k \xi_{kq})}{\xi_{kq}}, \end{cases}$$

где λ_k, β_q – множители Лагранжа. Заметим, что для заданного набора ρ_{kq} величины λ_k могут быть одно-

значно найдены из R_k . С другой стороны, как было показано в [3], из этих уравнений следует, что β_q должно быть равно $\beta_q = \min_k \beta_q^{(k)}$ и только пользователи с $\beta_q^{(k)} = \beta_q$ могут использовать q -й подканал (группу подканалов). Для решения представленной системы уравнений может быть использован следующий алгоритм.

1. Сформировать начальный набор значений ρ_{kq} .
2. Вычислить λ_k из второго уравнения и подставить это значение в четвертое уравнение, получив $\beta_q^{(k)}$.
3. Найти наихудший подканал и наихудшего

пользователя $(q_w, k_w) = \arg \max_{q, k: \rho_{kq} > 0} (\beta_q^{(k)} - \beta_q)$, назначенного на этот подканал, а также наилучшего

пользователя $k_b = \arg \min_k \beta_q^{(k)}$.

4. Уменьшить долю ρ_{k_w, q_w} подканала q_w , занимаемую пользователем k_w , на $1/S$ и увеличить долю ρ_{k_b, q_w} занимаемую пользователем k_b , на эту же величину.
5. Повторять шаги 2 – 4 заданное число раз.

Исходное распределение пользователей по подканалам может быть получено, например, следующим образом. Для каждого пользователя могут быть найдены наилучшие подканалы и для них установлено $\rho_{kq} = \delta > 0$. Те подканалы, которые не вошли в число наилучших ни для одного пользователя, могут быть назначены, например, пользователям с наилучшими отношениями канал–шум на них. При этом должно выполняться условие нормировки.

Иногда описанный алгоритм сталкивается с той же проблемой, что и стандартные итеративные алгоритмы оптимизации, а именно с возникновением колебаний на шаге 4. В этом случае несколько пользователей циклически обмениваются подканалами, что препятствует достижению оптимального решения. Причина заключается в том, что передача доли подканала от одного пользователя к другому может незначительно улучшить условия работы первого пользователя и существенно ухудшить показатели второго пользователя, вследствие чего на следующей итерации алгоритма будет произведен обратный обмен. Эта проблема может быть преодолена с помощью стандартного приема – «сглаживания». В данном случае он может быть реализован путем принудительного запрета на выбор в качестве «наихудших» на шаге 3 тех пользователей, которые были выбраны на предыдущих W итерациях в качестве «наилучших», где $W > 0$ – длина хранящейся истории.

Отметим, что использование расширения в частотной области ($S_f > 1$) приводит к снижению размерности оптимизационной задачи. Кроме того, сокращается объем передаваемой служебной информации. Отметим также, что применение кодового разделения позволяет естественным образом реализовать случай $S_f > 1$.

Чувствительность к изменениям состояния канала. В большинстве практических систем оказывается, что состояние канала подвержено сто-

хастическим флуктуациям. Ясно, что использование схемы передачи, не соответствующей текущему состоянию канала, может привести к катастрофическому ухудшению качества работы системы. В связи с этим возникает задача построения адаптивных методов, которые могли бы учитывать временные изменения состояния канала. Классическим методом решения этой задачи является предсказание состояния канала [4]. Однако в многопользовательской многочастотной системе применение данного метода оказывается затруднительным ввиду того, что он требует хранения чрезмерно большого объема информации о состоянии канала каждого из пользователей в предшествующие моменты времени. В связи с этим рассмотрим упрощенный подход, использующий информацию о состоянии канала только в один момент времени

В большинстве случаев радиоканал может быть охарактеризован моделью Релея, согласно которой величины $\xi_{kq}^{(j)}$ имеют экспоненциальное распределение ($\sqrt{\xi_{kq}^{(j)}} = \frac{|\mu_{kq}^{(j)}|}{\sigma}$ – распределение Релея) Так как передаточные коэффициенты радиоканала зависят, возникает необходимость рассмотрения их совместной плотности распределения (для упрощения расчетов далее индексы k и q будут опускаться). Можно показать [5], что условная плотность распределения отношения канал–шум на каждом из подканалов может быть вычислена как

$$\rho(\xi^{(j)} | \xi^{(0)}) = \frac{1}{\chi(1-\rho_j^2)} \exp\left(-\frac{\xi^{(j)} + \xi^{(0)}\rho_j^2}{\chi(1-\rho_j^2)}\right) \times I_0\left(\frac{2|\rho_j|}{\chi(1-\rho_j^2)} \sqrt{\xi^{(0)}\xi^{(j)}}\right),$$

где $\chi = M\{\xi^{(j)}\}$, $\rho_j = M\{\xi^{(0)}\xi^{(j)}\}$, $I_0(x)$ – модифицированная функция Бесселя нулевого порядка. Предположим, что вероятность ошибки для используемого семейства методов передачи может быть вычислена как

$$P_e(c, \xi, V) = AQ \left(\frac{\alpha V^2 \xi}{\sqrt{2^c - 1}} \right),$$

где c – скорость передачи, V^2 – мощность сигнала на данном подканале (или их группе), A и α – некоторые константы. Тогда условие поддержания заданной средней вероятности ошибки может быть сформулировано как

$$P_{cp} = \sum_{j=0}^J \int_0^\infty P_e(c, \xi, V) f(\xi^{(j)} | \xi^{(0)}) d\xi^{(j)} = \sum_{j=0}^J \frac{A}{1-\rho_j^2} \int_0^\infty Q\left(\frac{\alpha V^2 \chi x}{\sqrt{2^c - 1}}\right) \times$$

$$x \exp \left(-\frac{x + \frac{\xi^{(0)}}{\chi} \rho_j^2}{1 - \rho_j^2} \right) / \int_0 \left(\frac{2 |\rho_j|^2}{1 - \rho_j^2} \sqrt{\frac{\xi^{(0)}}{\chi}} x \right) dx.$$

Несложно заметить, что параметрами этого уравнения являются величина $F = \frac{\alpha V^2 \chi}{2^c - 1}$, которую можно рассматривать как нормированное отношение сигнал-шум, требуемое для достижения заданной вероятности ошибки, и отношение $\frac{\xi^{(0)}}{\chi}$, характеризующее отклонение отношения канал-шум в начальный момент времени от своего среднего значения.

Для каждого заданного $\frac{\xi^{(0)}}{\chi}$ данное уравнение может быть решено численно относительно F . Это дает возможность построить функцию $g\left(\frac{\xi^{(0)}}{\chi}, c\right) = F \frac{\xi^{(0)}}{\chi}$, задающую нормализованный запас мощности, требуемый для обеспечения заданной вероятности ошибки. Тогда функция $f(c)$, использованная при построении оптимизационного алгоритма для случая статического канала, может быть заменена на функцию

$$f(\xi^{(0)}, c) = g\left(\frac{\xi^{(0)}}{\chi}, c\right) \frac{2^c - 1}{\alpha}.$$

Это позволяет сохранить структуру предложенного оптимизационного алгоритма, модифицировав лишь используемые в нем коэффициенты ξ_{k_i} . Необходимо отметить, что численное решение нелинейного уравнения необходимо осуществлять лишь на этапе проектирования системы. Полученная таким образом функция g может быть сохранена в виде таблицы.

Исследование характеристик предложенного метода. Для оценки выигрыша, даваемого предложенным методом адаптивной передачи, было использовано имитационное моделирование. Рассматривалась стохастическая модель радиоканала со стационарным в широком смысле некоррелированным рассеянием (WSSUS) [6] с экспоненциальным профилем многопутевой интенсивности. Спектральная полоса, занимаемая исследуемой системой, предполагалась равной 20 МГц, число подканалов OFDM системы было положено равным 512. Параметрами модели являлись максимальная величина многопутевой задержки сигнала t_{max} , соответствующая затуханию сигнала на 30 дБ, и максимальная величина Доплеровского сдвига. Во всех случаях использовалась 2^c-КАМ и требовалось обеспечить вероятность ошибки на бит $2 \cdot 10^{-3}$. Мощность передаваемого сигнала, необходимая для достижения заданной скорости передачи данных и заданной вероятности ошибки, а также действительная вероятность ошибки, наблюдаемая приемником, рассматривались как критерии качества работы системы.

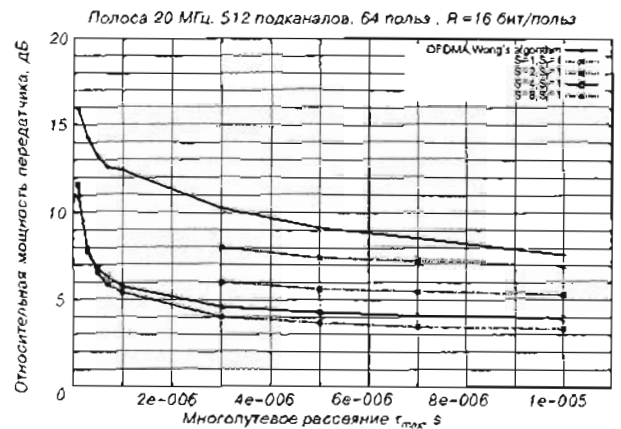


Рис. 1. Выигрыш, получаемый за счет разделения подканалов

На рис. 1 представлены результаты, иллюстрирующие выигрыш, получаемый за счет совместного использования подканалов несколькими пользователями.

Можно заметить, что при $S = 1$ (т. е. разделение подканалов отсутствует, система сводится к частотному разделению) предложенный метод позволяет получить выигрыш до 1 дБ по сравнению с алгоритмом Вонга [3], что еще раз подтверждает субоптимальность последнего. Кроме того, с увеличением S выигрыш за счет совместного использования подканалов быстро возрастает и достигает 5 дБ при $S = 8$. Дальнейшее увеличение S (т. е. повышение точности вычисления коэффициентов разделения ρ_{kq}) не дает существенного выигрыша.

На рис. 2 представлены результаты, полученные для систем с различными параметрами частотно-временного расширения S_f и $S_t = S_f/S_r$. Здесь состояние канала предполагалось неизменным на протяжении всего цикла передачи. Несложно заметить, что использование расширения в частотной области ($S_f > 1$) приводит к незначительному увеличению мощности передатчика, требуемой для достижения заданных параметров системы. Кроме того, как будет показано ниже, реальная вероятность ошибки, наблюдаемая приемником, оказывается несколько больше требуемой. Это связано с возникновением

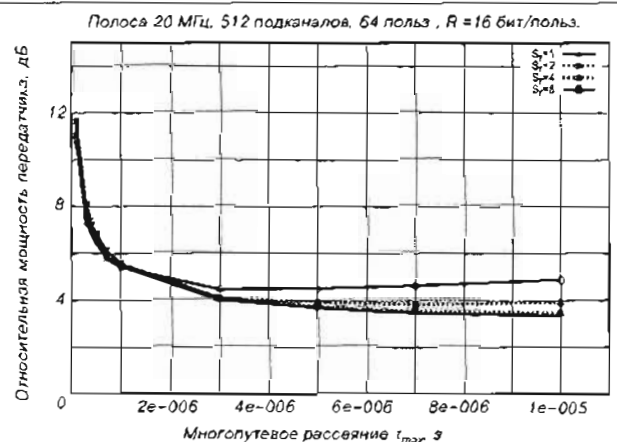


Рис. 2. Сравнение систем с различными параметрами частотно-временного расширения

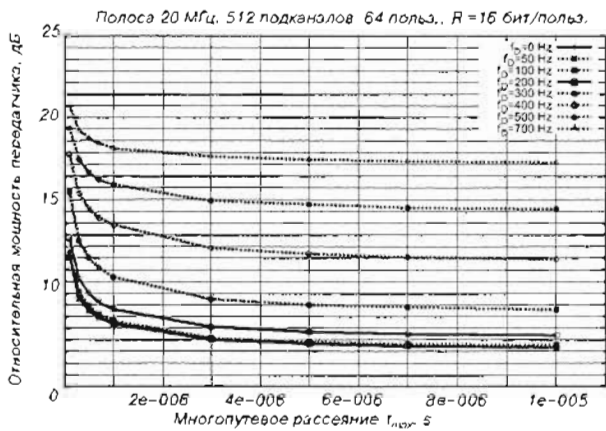


Рис. 3. Влияние временных изменений канала на требуемую мощность передатчика

межпользовательской интерференции вследствие нарушения ортогональности расширяющих последовательностей. Однако эти недостатки компенсируются существенным снижением сложности оптимизации и объема передаваемой служебной информации. Необходимо также отметить, что использованная тривиальная процедура формирования групп подканалов обеспечивает достаточно хорошие показатели качества работы системы.

На рис. 3 представлены аналогичные кривые, полученные для случая стохастически изменяющегося канала. Здесь предполагалось, что адаптация производится после передачи каждых $J = ST$ OFDM-символов, где $T = 3$. Можно заметить, что с ростом величины максимального доплеровского рассеяния f_D мощность передатчика, требуемая для обеспечения заданной вероятности ошибки, быстро возрастает. Из рассмотрения известных результатов для подобных неадаптивных систем следует, что применение описанного адаптивного алгоритма с приведенными выше параметрами нецелесообразно при $f_D > 400$ Гц. Отсюда можно заключить, что временной интервал между обновлениями схемы передачи должен составлять

$$t_{\text{обновл}} \approx \frac{1}{4f_D}$$

На рис. 4 представлена зависимость истинной вероятности ошибки, наблюдаемой приемником, от величины максимального доплеровского рассеяния. Отметим, что при $f_D = 0$ (статический канал) в системе с частотным расширением истинная вероятность ошибки оказывается несколько больше требуемой. Это связано с наличием межпользовательской интерференции, вызываемой потерей ортогональности расширяющих последовательностей. Однако с ростом f_D истинная вероятность ошибки во всех случаях несколько уменьшается, что связано с возникновением эффекта разнесения.

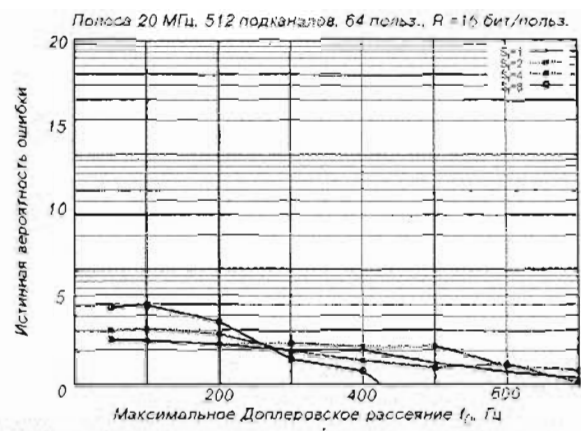


Рис. 4. Влияние временных изменений канала на истинную вероятность ошибки

Выводы

В работе описана адаптивная многопользовательская система. Показано, что за счет оптимизированного совместного использования подканалов может быть получен выигрыш до 5 дБ по сравнению с адаптивной системой с частотным разделением. Сравнение систем с различными конфигурациями частотного и временного расширения позволяет сделать вывод о предпочтительности использования расширения в частотной области ($S_f > 1$) ввиду существенного снижения сложности оптимизации и объема передаваемой служебной информации при незначительных потерях в качестве работы системы. Вместе с тем, анализ устойчивости предложенного метода к временным стохастическим изменениям состояния канала показывает, что применение предложенного адаптивного метода требует достаточно частых обновлений используемой схемы передачи.

Литература

1. Прокис Дж. Цифровая связь. – М.: Радио и связь, 2000. – 797 с.
2. Yang L.-L., Hanzo L. Multicarrier DS-SS-CDMA: A multiple access scheme for ubiquitous broadband wireless communications // IEEE Communications Magazine. – 2003. – Vol. 41. – N 10, P. 116–124.
3. Wong C. Y., Cheng R. S., Letaief K. B., Murch R. D. Multiuser OFDM with adaptive subcarrier, bit, and power allocation // IEEE Journal on selected areas in Communications. – 1999. – Vol. 17. – N 10. – P. 1747–1758.
4. Falahati S., Svensson A., Ekman T., Sternad M. Adaptive modulation systems for predicted wireless channels // IEEE Transactions on Communications. – 2004. – Vol. 52. – N 2. – P. 307–316.
5. Mallik R. K. On multivariate Rayleigh and exponential distributions // IEEE Transactions On Information Theory. – 2003. – Vol. 49. – N 6. – P. 1499–1515.
6. Hoher P. A statistical discrete-time model for the WSSUS multipath channel // IEEE Transactions On Vehicular Technology. – 1992. – Vol. 41. – N 4. – P. 461–468

УДК 004.3.067

НОВЫЙ МЕТОД ОЦЕНКИ АЛГОРИТМОВ РАСПРЕДЕЛЕНИЯ OVSF-КОДОВ В СТАНДАРТЕ WCDMA

И. А. Мельник,
ведущий инженер-программист
ООО «Научно-производственная фирма "Беркут"»

OVSF (Orthogonal Variable Spreading Factor) коды используются для поддержки разных скоростей передачи данных для различных пользователей в WCDMA системах. В данной работе, проанализированы существующие алгоритмы распределения кодов и решаемые ими проблемы. Для того чтобы оценить эффективность алгоритмов, мы предлагаем метод, который учитывает возникающие блокировки между кодами во время распределения. На базе этого метода можно построить аналитические модели для оценки алгоритмов.

Orthogonal Variable Spreading Factor (OVSF) codes are used to support different transmission rates for different users in WCDMA systems. In this paper, we analyze existing code allocation algorithms and problems that they have to solve. To estimate the efficiency of the algorithms we propose a method, which consider blockings between codes during assignment. On basis of this method it is possible to build analytical models for an estimation of different algorithms.

Введение

WCDMA (Wideband CDMA) – системы третьего поколения (3G), которые поддерживают переменную скорость передачи данных для приложений с различным «качеством услуги» (QoS – Quality of Service). Появление 3G систем было связано с появлением возможности передачи данных на больших скоростях (до 2 Мбит/с). В любой CDMA системе множественный доступ к каналам передачи данных обеспечивается с помощью кодов, при этом полоса частот – одна на всех пользователей. Поэтому схемы управления кодами являются составляющей частью системы, и главной целью их является эффективное использование ресурсов сети. В данной статье предлагается вероятностный метод оценки алгоритмов распределения OVSF-кодов в стандарте WCDMA.

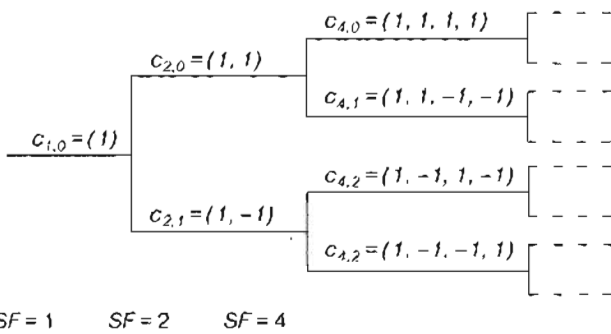
Каналообразующие коды в WCDMA

WCDMA – это телекоммуникационная система, которая разделяется на две основные части: радиосеть (UTRAN) и проводная сеть. В радиосети передаваемые пакеты информации распределяются с помощью RNC (Radio Network Controller), а именно, распределение представлено MAC-протоколом. MAC распределяет пакеты данных в оба выделенных канала: контрольный канал (DPCCH – Dedicated

Physical Control Channel) и канал данных (DPDCH – Dedicated Physical Data Channel), а также в разделяемый канал (PD-SCH – Physical Dedicated Shared Channel, далее DSCH). В DSCH может происходить распределение каналообразующих кодов между подключенными абонентами или мобильными станциями каждый раз для фрейма данных [1].

Отображение каналов в системе следующее. Логические каналы отображаются на транспортные, которые характеризуются тем, как информация будет передаваться по радиоинтерфейсу. Они предоставляются физическим уровнем для MAC-уровня. Транспортные каналы отображаются в физические (DPCH – Dedicated Physical Channel) на физическом уровне. В DPCH транспортные каналы мультиплексируются с помощью каналообразующих кодов. Далее информация из всех DPCH суммируется и скремблируется с помощью кода Голда. Одна базовая станция использует один код Голда, т. е. с помощью кодов Голда разделяются между собой сигналы базовых станций в прямом канале (от базовой станции к мобильной). Физический уровень контролируется с помощью RRC (Radio Resource Controller) [2].

В WCDMA в качестве каналообразующих кодов используются ортогональные OVSF-коды (Orthogonal Variable Spreading Factor), которые обеспечивают множественный доступ, а также используются для



SF = 1 SF = 2 SF = 4

■ Рис. 1. OVSF-кодированное дерево

расширения спектра сигнала. Расширение спектра достигается путем отображения каждого бита данных на выделенную кодовую последовательность. Длина последовательности называется расширяющим коэффициентом (SF – Spreading Factor). Возможные OVSF-коды могут быть представлены узлами полного двоичного дерева, которое изображено на рис. 1 [3].

На каждом уровне дерева все кодовые последовательности ортогональны. Более того, последовательности разной длины на разных уровнях тоже ортогональны друг другу, за исключением случая, когда более короткая последовательность является родительской для более длинной. Обозначим код c_i как родителя кода c_j . Выделяются следующие важные признаки дерева [4]:

1. Каждый узел c_j , за исключением корня, может быть сгенерирован из предка c_i . Допустим, c_i есть x . Далее мы можем получить $c_j = [x, x]$, если c_j – правый потомок c_i или $c_j = [x, -x]$, если c_j – левый потомок c_i , где $-x$ – инвертированная последовательность x .

2. Если $SF\ c_i = 2^k$, тогда $SF\ c_j = 2^{k+1}$. Следовательно, если скорость передачи данных $c_i = 2^l$, то скорость передачи $c_j = 2^{l-1}$. Таким образом, листья имеют минимальную скорость передачи данных, а корень – максимальную.

Одновременное использование потомков или предков не может быть разрешено, потому что коды, имеющие прямые родственные связи, не ортогональны.

Стандартом определены два критерия при распределении OVSF-кодов [5].

1. *Использование.* Схема выделения кодов, которая сохраняет большее количество кодов с малыми SF, имеет больший шанс на использование. Дело в том, что мобильная станция имеет ограничение k на количество одновременно используемых кодов. Если стратегия определила n кодов для передачи больше k , то мобильная станция блокируется.

2. *Сложность.* Чем больше используется кодов, тем сложнее система. Возможна ситуация, когда альтернативы из одного и двух кодов. В этом случае предпочтительнее выбор схемы с меньшим количеством кодов, т. е. выбор одного кода.

Существующие алгоритмы распределения кодов

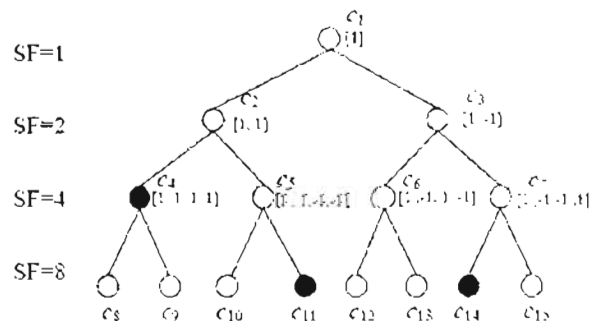
Описание основных (классических) стратегий распределения кодов можно найти в работе [6].

Целью или задачей стратегий и алгоритмов управления OVSF-кодами служит разрешение двух основных проблем [6]: 1) проблемы распределения (выдачи) кодов; 2) проблемы перераспределения кодов. Обе проблемы связаны с возникающими блокировками родительских и дочерних кодов при выделении кода. Чем больше блокировок – тем меньше емкость дерева (т. е. количество кодов, которые могут быть выделены новым МС), и тем больше вероятность отклонения запросов на подключение к системе. На рис. 2 показан пример дерева из четырех уровней, в котором выделено три кода

Проблема распределения кодов возникает в момент запроса при наличии нескольких подходящих кандидатов. Должна быть выбрана такая стратегия, которая при увеличении нагрузки стремится уменьшить количество блокируемых кодов при выделении кода. Также стратегия должна стремиться оставить коды с меньшим SF (скоростные коды) для удовлетворения запросов услуг с высокой скоростью.

Проблема перераспределения кодов возникает тогда, когда емкость дерева позволяет удовлетворить запрос, но подходящего свободного кода нет, т. е. он заблокирован из-за занятых потомков с большими SF. В этом случае можно «освободить» требуемый код путем размещения занятых потомков в соседних частях дерева на тех же SF-уровнях. Здесь должна быть выбрана такая стратегия, которая минимизирует затраты в процессе перераспределения кодов, чтобы уменьшить задержку при обработке запроса. Например, в ситуации, изображенной на рис. 2, коды с SF = 2 заблокированы и запрос на код будет отклонен. Если освободить код c_{14} , то коды c_7 и c_3 будут разблокированы и запрос на код с SF = 2 может быть удовлетворен. Алгоритм перераспределения в момент запроса на код с SF = 2 должен поменять выделенный мобильной станции код c_{14} на код c_{10} .

Перераспределение включает определение поддерева, которое должно быть освобождено (т. е.



■ Рис. 2. Пример используемого OVSF-дерева:

- – выделенные коды;
- – заблокированные коды;
- – свободные коды

поддерживаем с подходящим заблокированным корневым кодом, выделение которого произойдет в соответствии со стратегией выдачи кодов). Помимо этого, алгоритм перераспределения должен определить, куда будут перемещены занятые коды (т. е. поиск свободных кодов в соседних частях кодового дерева, в соответствии со стратегией выдачи кодов, с SF, равными SF занятых потомков подходящего заблокированного кода).

Основными можно назвать три стратегии [6]:

1. Случайный доступ (random).
2. Использование самого левого (leftmost).
3. Самый заполненный – первый (crowded-first).

Первая стратегия представляет собой случайный доступ к узлам дерева с требуемым SF с равными вероятностями. Если при поступлении вызова есть несколько свободных кодов с нужным SF, то случайным образом выбирается какой-нибудь код. В противном случае запрос отклоняется.

Суть второй стратегии состоит в том, что при наличии нескольких подходящих кодов выбирается самый левый. Другими словами, правая часть дерева оставляется для кодов с меньшим SF, или в правой части концентрируется вся оставшаяся емкость дерева.

В случае использования третьей стратегии при наличии нескольких подходящих кодов выбирается тот, чье поддерево до ближайшего корня заполнено в большей степени. При повторном возникновении конфликта уровень корня понижается, и сравниваются полученные поддеревья. Если конфликт не разрешен с учетом описанного алгоритма, то выбирается либо самый левый, либо самый правый коды, в зависимости от настройки.

Стратегии перечислены в порядке увеличения эффективности и увеличения вычислительных затрат при реализации.

Примеры других стратегий и алгоритмов можно найти в работах [1], [7] – [10].

Вероятностный метод оценки

Определим состояние как количество занятых кодов на уровнях дерева. Предположим, что имеется дерево из четырех уровней. Тогда состояние – это вектор из K_i элементов, $i = 1, 2, 3, 4$; K_i – число занятых кодов на уровне i . Увеличение значения K_i или переход в другое состояние означает выдачу кода, уменьшение – высвобождение. Первый переход происходит с интенсивностью запросов с QoS, требующих коды на различных уровнях дерева. Второй переход происходит с интенсивностью обработки вызова или, другими словами, зависит от среднего времени использования абонентом услуги (среднего времени занятости кода).

Прием запроса ограничивается емкостью дерева или выражением

$$K_1 + 2K_2 + 4K_3 + 8K_4 \leq C_{\max}, \quad (1)$$

где C_{\max} – количество листов кодового дерева; K_i – коды с наибольшим SF и наименьшей скоростью (листья).

В начальном состоянии все значения элементов K_i равны нулю. Построим граф возможных переходов из начального состояния. Пусть поток запросов кодов, поступающий в систему, – простейший. Простейший поток характеризуется следующими свойствами.

1. Интенсивность потока постоянна и не зависит от времени.

2. События в потоке появляются по одному, а не группами, одновременно по несколько событий сразу.

3. События, образующие поток, появляются в те или иные моменты времени независимо друг от друга; каждое из них вызвано своими собственными причинами.

В этом случае переход из состояния в новое состояние означает увеличение или уменьшение какого-то элемента K_i на единицу. Интенсивность переходов при увеличении значения K_i может быть неодинаковой и зависеть от скоростных характеристик кода или от услуг, запрашиваемых абонентами. Сопоставим каждому переходу, или ребру графа, вероятность перехода, или вероятность запроса абонентом услуги, которая предоставляется с помощью кода с определенными скоростными характеристиками. Сумма вероятностей переходов из состояния в другие возможные состояния равна единице.

В случае, когда используется алгоритм перераспределения кодов, то можно сказать, что если при поступлении запроса емкость дерева позволяет «вместить» его, то он будет обработан с вероятностью 1, т. е. если при поступлении запроса выполняется неравенство (1), то запрос будет обработан.

Вероятность блокировки запросов – это доля времени, в течение которого система находится в состояниях, когда переход в новое состояние нарушает условие (1).

Общая пропускная способность системы – это сумма произведений пропускной способности каждого состояния на долю времени, в течение которого система находится в этом состоянии (предельная вероятность). Пропускная способность состояния – это количество информации, например, в байтах, которое можно передать с помощью кодов, используемых в этом состоянии.

В случае, если алгоритм перераспределения не используется, то даже при выполнении условия (1) возможны ситуации, когда код не выдается. Это может происходить ввиду блокировок, возникающих при выдаче кода. Для отражения состояния по заблокированным кодам, каждому K_i сопоставим значение B_i , которое показывает количество заблокированных кодов на этом уровне. Теперь состояние – это количество занятых кодов и количество заблокированных кодов на каждом уровне дерева, т. е. состояние описывается двумя векторами: $\{K\}_N$, $\{B\}_N$, где N – количество уровней дерева.

При запросе кода на уровне i вероятность перехода на следующий уровень будет равняться нулю или блокировка вызова будет возникать тогда, ког-

да $B_i + K_i = c_i$, где c_i – максимальное количество кодов на уровне i . Соответственно, для принятия вызова необходимо, чтобы выполнялось условие

$$B_i + (K_i + 1) \leq c_i. \quad (2)$$

Выражение (1) при этом дополняется выражением (2), т. е. невыполнение одного из условий влечет за собой отказ в обработке запроса на код.

Вероятность блокировки в этом случае – доля времени, в течение которого система находится в состояниях, переход из которых в следующее нарушит условие (1) или (2).

Выдача кода дерева влечет за собой блокировку всех потомков и родителей кода из-за отсутствия ортогональности между ними. Таким образом, увеличение K_i влечет за собой увеличение значений B на всех уровнях, меньших i (дочерние коды), и может повлечь увеличение значений B на следующих уровнях, больших i (родительские коды).

Обозначим N количество уровней в дереве, n – уровень, на котором происходит выдача кода, $n - 1$ – уровень кодов с наибольшим SF и наименьшей скоростью (листья). Тогда при выдаче кода на уровне n количество блокируемых кодов на уровнях $1 \leq i \leq (N - n)$ (все дочерние) равно $B_i = 2^{N-n-i+1}$. Общее количество блокируемых дочерних кодов равно

$$B_{desc} = \sum_{i=1}^{N-n} 2^{N-n-i+1}. \quad (3)$$

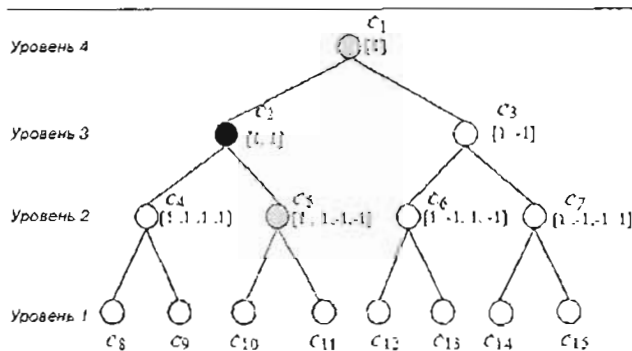
Таким образом, если не учитывать количество блокируемых родительских кодов, то при обработке запроса на код уровня n , т. е. при увеличении K_n , изменение вектора K будет следующим:

$$K_N, K_{N-1}, \dots, K_{n+1}, \dots, K_1.$$

При этом вектор B изменится следующим образом:

$$B_N, B_{N-1}, \dots, B_{N-n} + 2, B_{N-n-1} + 4, \dots, B_1 + 2^{N-n}. \quad (4)$$

Следовательно, выдача любого кода на уровне n всегда влечет за собой блокировку фиксированного количества дочерних кодов, рассчитываемых по формуле (3), при этом вектор B принимает вид (4). Другими словами, количество блокируемых дочерних кодов для каждого уровня фиксировано,



■ Рис. 3. Пример блокировки дочерних кодов при выдаче кода на третьем уровне:
 ● – выделенные коды;
 ■ – заблокированные коды;
 ○ – свободные коды

или наличие вектора B не увеличивает количество состояний, в которые можно перейти из текущего состояния.

На рис. 3 показан пример дерева, где выдан код c_2 . Количество заблокированных дочерних кодов равно шести. При выдаче кода c_3 количество заблокированных дочерних кодов будет тем же.

Рассмотрим блокировку родительских кодов. Количество блокируемых кодов зависит от того, какие «родственники» кода уже выданы. Интерес представляет любой «родственник» на том же уровне, что и целевой код, у которого родительский код является родителем целевого и имеет наименьшее расстояние (в уровнях) до него. Количество заблокированных кодов выражается формулой

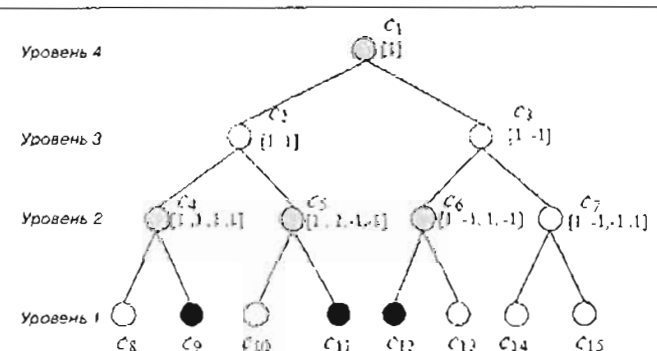
$$B_{asc} = l - i - 1, \quad (5)$$

где i – уровень целевого кода $c_{i,j}$; l – уровень родительского кода c_l , который одновременно является родителем выданного кода $c_{i,k}$. Из всех имеющихся $c_{l,k}$ – занятых кодов на уровне l – выбирается такой c_l , что расстояние от $c_{i,j}$ до c_l минимально

$$\min_l B_{asc} = l - i - 1.$$

В случае, если $c_{i,j}$ – первый выдаваемый код во всем дереве, то количество блокируемых родительских кодов всегда равно $N - i$, где N – общее количество уровней в дереве. Нужно отметить, что на каждом уровне, большем, чем n , может быть заблокирован только один родительский код. Например, если количество блокируемых родительских кодов равно двум, то это означает, что один из двух кодов находится на уровне $n + 1$, а второй – на уровне $n + 2$.

На рис. 4 для кода c_{11} занятыми родственниками на том же уровне являются коды c_9 и c_{12} . Ближайший общий родитель у кодов c_{11} и c_{12} – это код c_1 . Ближайший общий родитель у кодов c_{11} и c_9 – это код c_2 . В качестве l выбираем уровень 3, так как код c_2 ближе к коду c_{11} , чем код c_1 . По формуле (5) рассчитываем, что количество блокируемых родительских кодов при выдаче кода c_{11} , равно 1 – это код c_5 . Если бы код c_9 был свободен, то в качестве l мы выбрали бы уровень 4, и тогда количество блокируемых кодов равнялось бы 2 – это коды c_5 и c_2 .



■ Рис. 4. Пример блокировки родительских кодов при выдаче кода на первом уровне:
 ● – выделенные коды;
 ■ – заблокированные коды;
 ○ – свободные коды

Определение количества блокируемых родительских кодов при выдаче кода $C_{i,j}$ можно описать следующим алгоритмом

1. Если $C_{i,j}$ – это первый выдаваемый код, то $B_{asc} = N - i$, где N – общее количество уровней в дереве. Переход на шаг 5.

2. Отбросим все уровни, меньшие i , так, чтобы в полученном дереве код $C_{i,j}$ оказался листом.

3. Понизим корень текущего дерева и рассмотрим одно из двух поддеревьев, в котором находится $C_{i,j}$.

4. Есть ли хоть один занятый код на уровне i ?

4.1. Если истина и длина рассматриваемого поддерева равна 2, то $B_{asc} = 0$. Переход на шаг 5.

4.2. Если истина и длина рассматриваемого поддерева больше 2 – переход на шаг 3.

4.3. Если ложь, то $B_{asc} = N - 1$, где N – длина рассматриваемого поддерева. Переход на шаг 5.

5. Конец.

При начальном условии, когда все K_i равны нулю (т. е. ни один код не выдан), то количество блокируемых родительских кодов всегда фиксировано и равно $N - i$, где N – общее количество уровней в дереве; i – уровень выдаваемого кода. В случае, если хоть один K_i не равен нулю, то количество блокируемых кодов в общем случае может быть от нуля до $N - i - 1$. Элементы B_k , $k > i$, увеличиваются на единицу у всех уровней из промежутка

$$N \leq k < i.$$

Так как количество блокируемых дочерних кодов всегда фиксировано, то максимальное количество возможных новых состояний (альтернатив для перехода), при вводе вектора B , при выдаче кода на уровне j равно $N - i - 1$, в зависимости от раскладки уже занятых кодов на этом уровне.

Перестраивая граф состояний, получим дополнительные альтернативы перехода из состояния. Например, если дерево имеет четыре уровня и поступил запрос на код второго уровня, то до ввода вектора B была возможна только одна альтернатива, т. е. переход, после которого вектор состояний K имеет элемент K_2 (на единицу больше). После ввода вектора B изменения вектора K остаются теми же, но появляются три новых варианта.

1. Блокировка двух родительских кодов по одному на уровнях 3 и 4.

2. Блокировка одного родительского кода на уровне 3.

3. Ни один родительский код не блокируется.

Первый вариант возможен только в том случае, если выдаваемый код – первый во всем дереве. Таким образом, количество альтернатив перехода такое же, как и до ввода вектора B . Если выдаваемый код – не первый, то появляется дополнительная альтернатива, т. е. в данном случае имеется два возможных варианта перехода вместо одного. По большому счету, количество имеющихся альтернатив определяет минимально возможное количество кодов-кандидатов для выдачи: если есть две альтер-

нативы, то, как минимум, есть два подходящих кандидата для выдачи. Какой из вариантов будет выбран, зависит от используемого алгоритма выдачи кодов. Например, если используется случайный выбор (стратегия random), то вероятность переходов в обоих случаях равна 0,5, или, другими словами, данный алгоритм выбирает код из всех имеющихся кандидатов с равной вероятностью.

Таким образом, мы можем построить граф состояний, вероятности переходов у которого определены используемым алгоритмом распределения кодов, вероятностью запросов кодов различных характеристик, а также средней продолжительностью использования услуг абонентами. Далее, применяя к описанной системе известные формулы из теории массового обслуживания, можно получить характеристики системы, которые будут отражать работу того или иного алгоритма распределения кодов.

Заключение

В данной статье проведен обзор алгоритмов распределения каналов образующих кодов в стандарте WCDMA. Предложен вероятностный метод оценки алгоритма распределения кодов, который позволяет получить такие характеристики, как блокировка вызова, средняя пропускная способность при использовании того или иного алгоритма. Дальнейшие исследования будут направлены на получение графа состояний для конкретного OVFSF-дерева, на применение к полученной системе формул и оценке имеющихся на текущий момент алгоритмов распределения кодов данным методом.

Литература

1. Legg P., Villier E. Efficient channelisation code allocation in UMTS 2001 (<http://www.priorartdatabase.com/IPCOM/000004839/>).
2. TSG RAN WG1 Physical layer – General description Tech. Spec. TS25.201, 3GPP, December 2003 (<http://www.3gpp.org>).
3. TSG RAN WG1 Spreading and modulation (FDD) Tech. Spec. TS25.213, 3GPP, September 2003 (<http://www.3gpp.org>).
4. Yu C. W., Huang R. O. OVFSF Code Management Schemes on Ad Hoc Networks ICC, 2004, Paris (<http://www.chu.edu.tw/~cwyu/publication.htm>).
5. TSG RAN WG1 Radio resource management strategies Tech. Rep. TS25.922, 3GPP, December 2003 (<http://www.3gpp.org>).
6. Tseng Yu-Chee, Chao Chih-Min Code Placement and Replacement Strategies for Wideband CDMA OVFSF Code Tree Management IEEE Transactions on Mobile Computing 1(4): 293-302 (2002)
7. Chen Wen-Tsuen, Wu Ya-Ping, Hsiao Hung-Chang A Novel Code Assignment Scheme for W-CDMA Systems, Taiwan 300, R.O.C.
8. Dell'Amico M., Merani M. L., Maffio F. A Tree Partitioning Dynamic Policy for OVFSF Codes Assignment in Wideband CDMA, DISMI, DSI, DEI, Italy, March 2003.
9. Park Jun-Seong, Lee D. C. Enhanced Fixed and Dynamic Code Assignment Policies for OVFSF-CDMA Systems, Los Angeles, CA 90089, U. S. A.
10. Rouskas A. N., Skoutas D. N. Comparison of Code Reservation Schemes at the Forward Link in WCDMA, Samos 83200, Greece.

К 90 – ЛЕТИЮ АКАДЕМИКА Е. П. ПОПОВА

В ноябре 1999 г. не стало академика Е. П. Попова, выдающегося ученого в области механики и теории автоматического управления, крупного организатора науки и талантливого педагога. Евгений Павлович Попов родился 14 февраля 1914 года в Москве. Основные даты жизни и деятельности Е. П. Попова приведены в табл. 1, где в сжатой форме прослеживается его жизненный путь: от студента техникума точной индустрии – до академика, от солдата – до генерала.

В табл. 2 приведен список основных научных трудов Е. П. Попова (общий список насчитывает более 200 печатных работ).

Даже беглый анализ этих трудов позволяет выделить три крупные темы, связанные с тремя основными научными направлениями деятельности академика: механика, теория автоматического управления и робототехника. Заметим, что три государственных премии Е. П. Попова получены им соответственно за достижения именно в этих направлениях.

Меня, как ученика Е. П. Попова в области теории автоматического управления, в первую очередь интересовали его работы именно в этой области.

О том, что я не самозванный ученик Евгения Павловича, свидетельствует, например, следующий факт. Рассказывая в своих «Воспоминаниях» о подготовленных им в Военной академии им. А. Ф. Можайского докторов наук, он пишет: «Наконец, еще один мой ученик Р. М. Юсупов защитил в то время докторскую диссертацию по беспоисковым самонастраивающимся системам автоматического управления. Потом пошли уже более молодые сотрудники»¹.

В теорию автоматического управления Е. П. Попов пришел фактически из механики в 1949 г., когда он в возрасте 35 лет был назначен начальником кафедры автоматики и телемеханики Ленинградской военно-воздушной инженерной академии. Это была первая кафедра такого профиля среди военных вузов и одна из первых кафедр по автоматике среди всех вузов страны. Так, в ЛПИ такая кафедра уже существовала, но только с уклоном в релейную автоматику. В то же время теория автоматического регулирования еще только зарождалась. Полноценных книг и учебных курсов не было. Единственным источником для изучения нового направления был журнал «Автоматика и телемеханика», созданный в 1936 г. Е. П. Попову удалось также достать отрывочные конспекты лекций А. А. Фельдбаума, которые он читал на кафедре электротехники в Артиллерийской академии в Москве. Позднее он узнал, что подобные лекции читал на спецкурсах в МВТУ А. А. Воронов.

Е. П. Попов достаточно оперативно вошел в курс дела и уже в 1950 году в «Трудах ЛКВВИА» он опубликовал большой обзорный материал «Работы оте-

чественных ученых по созданию теории автоматического регулирования»². Заметим, что в этих же трудах кроме обзора Е. П. Попова была опубликована статья Я. З. Цыпкина «Устойчивость и автоколебания релейных систем автоматического регулирования», а ответственным редактором трудов был В. И. Сифоров, являвшийся в то время заместителем начальника ЛКВВИА по учебной и научной работе. Статья Я. З. Цыпкина была написана по материалам доклада, сделанного им в ЛКВВИА 3 февраля 1950 г. по приглашению Е. П. Попова.

В 1952–1953 годах в ЛКВВИА вышла в двух частях первая книга Е. П. Попова «Теория автоматического регулирования». В 1954 г. «Гостехиздатом» (впоследствии «Физматгиз») была издана «Динамика систем автоматического регулирования», ставшая настольной книгой нескольких поколений специалистов по автоматическому управлению. Очень скоро она вышла в переводе в Англии, Германии и США

Наибольшую известность Е. П. Попов получил благодаря своим работам в области приближенных методов исследования нелинейных автоматических систем. Интересуясь точными методами исследования нелинейных колебаний, он пришел к выводу, что они, во-первых, слишком сложны и, во-вторых, годятся в основном только для систем, описываемых уравнениями второго порядка.

Поэтому он вполне естественно обратился к приближенным методам, в частности, к методу Н. М. Крылова и Н. Н. Боголюбова. В результате Е. П. Попов пришел к идее метода гармонической линеаризации. Подробно этот метод изложен в книге Е. П. Попова и И. П. Пальтова «Приближенные методы исследования нелинейных автоматических систем»³.

Идея метода гармонической линеаризации достаточно проста. Пусть на вход нелинейного звена (рис. 1) подается гармонический сигнал $x = A \sin \omega t$.

Периодический сигнал на выходе нелинейного звена разлагается в ряд Фурье. С учетом фильтрующих свойств линейной части системы для случая симметричных нелинейностей сигнал y в первом приближении можно представить в виде

$$y \approx D_1 \sin \omega t + C_1 \cos \omega t = q(a)x + \frac{q^1(a)}{\omega} p x,$$

где $q(a) = \frac{D_1}{a}$, $q^1(a) = \frac{C_1}{a}$; $p = \frac{d}{dt}$, $q(a)$, $q^1(a)$ – коэффициенты гармонической линеаризации.

¹ Попов Е. П. Воспоминания. – М.: Изд-во МГТУ им. И. Э. Баумана, 1996. – С. 132.

² Труды ЛКВВИА. Вып. 32. – Л.: Изд-во ЛКВВИА, 1950.

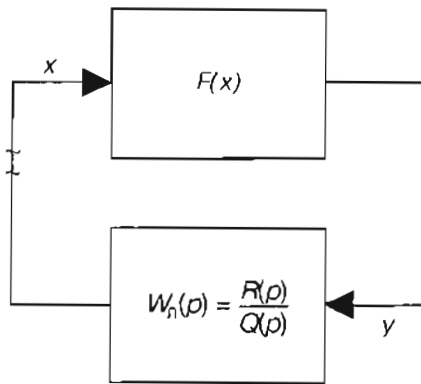
³ Попов Е. П., Пальтов И. П. Приближенные методы исследования автоматических систем. – М.: Физматгиз, 1960.

■ Таблица 1. Основные даты жизни и деятельности Е. П. Попова

Событие	Год
Окончание техникума точной индустрии	1934
Работа на заводе «Гизприбор»	1933 –1934
Учеба в МВТУ им. И. Э. Баумана, получение диплома инженера-механика	1934 –1939
Служба в армии	1939 –1943
Переход в ЛКВВИА в Йошкар-Оле и получение первого офицерского звания техника-лейтенанта	1943, май
Защита в МВТУ кандидатской диссертации на тему: «Динамика и прочность пружин»	1943, ноябрь
Утверждение в ученой степени кандидата технических наук	1944, январь
Защита в МВТУ докторской диссертации на тему: «Расчет гибких деталей приборов и машин (прикладная теория изгиба прямого и кривого бруса малой жесткости)»	1946
Утверждение в ученой степени доктора технических наук	1946, ноябрь
Получение ученого звания профессора	1948
Получение Сталинской (Государственной с 1954 г.) премии за работы в области теории и расчета гибких упругих деталей	1949
Назначение начальником кафедры автоматики и телемеханики ЛВВИА	1949
Создание на базе кафедры Е. П. Попова кафедр «Основы автоматики», «Электронные вычислительные машины военного применения», «Инфракрасная техника и аэрофотооборудование», научно-исследовательской лаборатории «Автоматизированные системы управления войсками» и факультета «Автоматизированные системы управления»	1952–1968
Назначение на работу (по совместительству) на должность руководителя Секции беспилотных объектов в НТК ВВС	1954
Избрание в члены-корреспонденты АН СССР	1960
Изменение названия и содержания кафедры «Системы управления ракет и КА»	1960
Присвоение воинского звания «генерал-майор»	1961
Назначение на должность Председателя Секции прикладных проблем при Президиуме АН СССР и переезд в Москву	1964
Демобилизация из армии и переход на должность заведующего кафедрой прикладной математики в МВТУ	1971
Назначение заведующим кафедрой следящих систем (М-7)	1971
Получение второй Государственной премии в коллективе авторов под руководством В. В. Солодовникова	1972
Открытие по инициативе Е. П. Попова специальности «Робототехнические системы» при головной роли кафедры М-7 МВТУ	1976
Создание научно-учебного центра «Робототехника» МВТУ и новой кафедры «Робототехнические системы»	1981
Переход в Отделение информатики, вычислительной техники и автоматизации РАН	1983
Получение третьей Государственной премии	1984
Избрание в действующие члены РАН	1992
Кончина Е. П. Попова	03.11.1999

■ Таблица 2. Основные научные труды Е. П. Попова

№ п/п	Название книг	Издательство	Год издания	Количество печатных листов
1	«Теория и расчет гибких упругих деталей»	ЛККВИА	1947	22
2	«Нелинейные задачи статики тонких стержней»	«Гостехиздат»	1948	11
3	«Курс теоретической механики (динамика)» (в соавт. с Н. В. Бутениным и С. И. Лазеевым)	ЛКВВИА	1948	25
4	«Курс теоретической механики (статика и динамика)» (в соавт. с Н. В. Бутениным и С. И. Лазеевым)	ЛКВВИА	1950	25
5	Работы отечественных ученых по созданию теории автоматического регулирования	Труды ЛКВВИА, вып. 32	1950	2
6	«Теория автоматического регулирования», ч. I	ЛКВВИА	1952	36
7	«Теория автоматического регулирования», ч. II	ЛКВВИА	1953	21
8	«Динамика систем автоматического регулирования»	«Гостехиздат»	1954	45
9	Лекции по теории автоматического регулирования	ЛКВВИА	1957	24
10	«Автоматическое регулирование и управление» (5 изданий)	«Физматгиз», «Наука»	1956–1966	19
11	«Приближенные методы исследования нелинейных автоматических систем» (в соавт. с И. П. Пальтовым)	«Физматгиз»	1960	45
12	«Системы управления космических аппаратов»	ЛКВВИА	1962	11
13	«Система управления ракеты» (в соавт. с А. Ромейковым)	ЛКВВИА	1964	6
14	«Теория систем автоматического регулирования» (три издания) (в соавт. с В. А. Бесекерским)	«Наука»	1966–1975	50
15	Серия из 10 книг по нелинейным системам автоматического управления (редактор и соавтор)	«Машино- строение»	1970–1991	220
16	«Прикладная теория процессов управления в нелинейных системах»	«Наука»	1973	35
17	«Манипуляционные работы. Динамика и алгоритмы» (в соавт. с А. Ф. Верещагиным и С. Л. Зенкевичем)	«Наука»	1978	20
18	«Система управления манипуляционных роботов» (в соавт. с В. С. Медведевым, А. Г. Лесковым и А. С. Ющенко)	«Наука»	1978	20
19	Серия из 4 книг по проектированию следящих систем (редактор)	«Машино- строение»	1978–1984	64
20	Серия из 6 книг по курсу ТАР (редактор и соавтор)	«Наука»	1978–1984	96
21	«Роботы и человек» (в соавт. с А. С. Ющенко)	«Наука»	1983	15
22	Серия из 9 книг по робототехнике (редактор и соавтор)	«Машино- строение»	1984–1989	180
23	«Теория расчета гибких упругих стержней»	«Наука»	1986	20
24	«Робототехника и гибкие производственные системы»	«Наука»	1987	12
25	«Основы робототехники» (в соавт. с Г. В. Письменным)	«Высшая школа»	1990	15
26	Серия из 8 сборников АН по научным проблемам робототехники (соредактор и соавтор)	Издательство СССР	1980–1990	80



■ Рис. 1. Структурная схема нелинейной автоматической системы

Тогда передаточная функция разомкнутой нелинейной системы принимает вид

$$W(p, a) = \frac{R(p)}{Q(p)} \left[q(a) + \frac{q'(a)}{\omega} p \right]$$

Коэффициенты гармонической линеаризации для некоторых типовых нелинейностей представлены в табл. 3.

С использованием метода гармонической линеаризации Е. П. Попов и его ученики провели ряд теоретических и прикладных исследований. В частности, было выявлено, что нелинейность привода рулей в присутствии вибраций приводит к неустойчивости движения полета некоторых типов летательных аппаратов (самолетов-снарядов, ракет и т. д.). Была также создана теория нелинейных корректирующих устройств в автоматических системах. Ряд других разработок описан в книгах Е. П. Попова.

Обобщая научную, организаторскую и педагогическую деятельность Е. П. Попова, хотелось бы выделить следующие четыре направления, на развитие которых он оказал серьезное влияние.

1. Развитие теории автоматического регулирования и управления в масштабах страны.

2. Разработка и внедрение в практику проектирования и создания систем автоматического управления инженерных методов и методик.

3. Организация системы подготовки военных и гражданских инженеров и ученых в области автоматических систем управления.

4. Развитие таких новых научно-технических направлений, как автоматизация, робототехника и информатика.

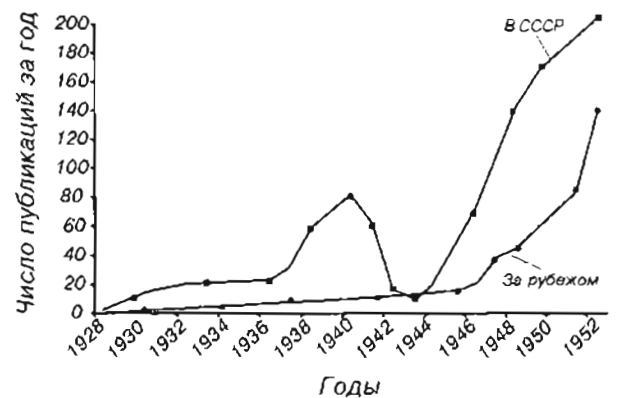
С именем Е. П. Попова и его коллег-академиков В. А. Трапезникова, А. А. Воронова, Я. Э. Цыпкина, А. А. Красовского, Б. Н. Петрова, членов-корреспондентов А. М. Летова, А. И. Лурье, В. И. Зубова и других связана целая эпоха успешного развития в нашей стране науки об автоматическом управлении и системах управления. Об этом свидетельствует хотя бы тот факт, что первый конгресс ИФАК по автоматическому управлению состоялся в 1960 г. в Москве под председательством президента ИФАК А. М. Ле-

това. Затем еще два конгресса были проведены на территории СССР – в Ташкенте и Таллине. Кроме того, регулярно проводились весьма популярные в СССР и в мире Всесоюзные совещания по автоматическому управлению. Наши специалисты были широко представлены в различных рабочих органах ИФАК. В 1940–1950-х годах, как отмечает А. А. Красовский¹, СССР имел несомненный приоритет в классической инженерной теории автоматического и автоматизированного регулирования с опережением США и других стран в этой области приблизительно на 15 лет. В качестве аргумента в пользу этого утверждения А. А. Красовский приводит сравнительные данные по числу публикаций на эту тему в СССР и за рубежом (рис. 2). Основной причиной нашего успеха в этот период он считает интеграционные процессы в науке «на базе целевой установки и целевого планирования индустриализации страны». Тогда в государственную программу автоматизации были привлечены крупные энергетики, электротехники, радиофизики, механики, математики. Они смогли осуществить интеграцию методов теории колебаний, теории устойчивости, нелинейной механики, радиотехники в комплексе с работами пионеров теории регулирования предшествующего столетия – И. И. Ползунова, И. А. Вышнеградского, П. Л. Чебышева, А. М. Ляпунова и др.

К сожалению, с конца 80-х годов прошлого столетия отечественная школа по теории управления, по нашему мнению, несколько замедлила свое развитие. В своей статье А. А. Красовский говорит даже о кризисном состоянии теории управления.

Можно назвать несколько причин, как-то объясняющих это негативное явление.

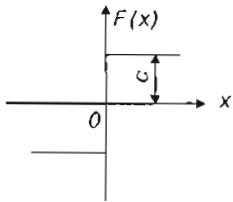
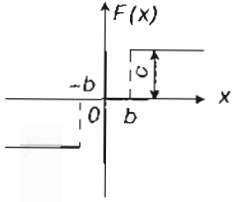
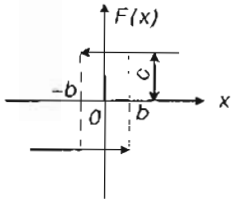
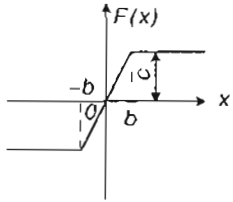
Во-первых, это, конечно, общий кризис, охвативший нашу науку на фоне реформирования политической, экономической и социальной сфер страны



■ Рис. 2. Сравнительные данные по числу публикаций в СССР и за рубежом по развитию классической инженерной теории автоматического регулирования

¹ Красовский А. А. Науковедение и состояние теории процессов управления // Автоматика и телемеханика. – 2000. – № 4.

■ Таблица 3. Коэффициенты гармонической линеаризации для некоторых типовых нелинейностей

Характеристика нелинейного звена		$d\phi$	$d\psi$
Идеальная релейная		$\frac{4c}{\pi a}$	0
Релейная с зоной нечувствительности		$\frac{4c}{\pi a} \sqrt{1 - \frac{b^2}{a^2}}$ $a \geq b$	0
Релейная с гистерезисной петлей		$\frac{4c}{\pi a} \sqrt{1 - \frac{b^2}{a^2}}$ $a \geq b$	$-\frac{4cb}{\pi a^2}$ $a \geq b$
С насыщением		$\frac{2c}{\pi b} \left(\arcsin \frac{b}{a} + \frac{b}{a} \sqrt{1 - \frac{b^2}{a^2}} \right)$ $a \geq b$	0

после распада СССР. Во-вторых, по мнению А. А. Красовского, «кризис современной теории автоматического и автоматизированного управления обусловлен нарушением интеграционных принципов науковедения, органического сочетания теории и практики». В-третьих, дальнейшему развитию теории управления, как нам представляется, не способствовали некоторые организационные решения, принятые при создании в составе АН СССР в 1983 г. Отделения информатики, вычислительной техники и автоматизации. Несомненно, такое отделение было необходимо. Но его создание в определенной мере произошло за счет ведущих тогда управленцев – академиков А. А. Воронова, Е. П. Попова, Г. С. Поспелова, Я. Э. Цыпкина, С. В. Емельянова, И. М. Макарова, Н. Н. Моисеева, Н. А. Кузнецова, которые перешли в новое отделение из Отделения механики и процессов управления. В результате ряды управленцев были расколоты, а автоматизация была организационно оторвана от управления (в рамках АН СССР).

Сегодня многих из названных академиков уже нет с нами. Ушли от нас и такие выдающиеся ученые, лидеры и организаторы науки в сфере управления, как

академики В. А. Трапезников, Б. Н. Петров, А. И. Берг, В. М. Глушков

Наконец, в-четвертых, с начала 80-х годов происходило бурное развитие информатики и вычислительной техники, начали появляться новые научные организации и коллективы по этой тематике, которые, естественно, оттянули на себя часть кадров из смежных областей, в том числе из управления. Многие управленцы переквалифицировались в информатиков.

Некоторое время теория управления и информатика развивались в значительной степени, по крайней мере в организационном плане, независимо и даже под «разными знаменами»: ИФАК и ИФИП. Но объективная реальность и жизнь диктуют и формируют свои процессы, связанные с тем, что сегодня управление и принятие управленческих решений невозможны без наличия соответствующей информации об объекте, о цели управления, об окружающей среде и т. д. Управление по своей природе есть совокупность действий, имеющая в основном сугубо информационный характер. Сегодня управление практически невозможно представить без компьютеров или микропроцессоров. Особо остро пробле-

ма использования компьютеров и информационных технологий возникает в системах управления в реальном времени бизнесом, т. е. в системах управления производственными и социальными коллективами. Традиционная теория управления и кибернетика в основном рассматривали технические системы и производственные процессы, как правило, без включения человека (коллектива) в качестве объекта управления или элемента системы управления.

Особенности рыночной экономики и конкурентной борьбы требуют автоматизации в условиях реального времени и бизнес-процессов. Естественно, что реализация этого процесса невозможна без широкого использования компьютерных и сетевых технологий в интересах обеспечения актуальной информацией всех субъектов, участвующих в бизнес-процессе, – руководителей, исполнителей (сотрудников), партнеров и потребителей.

В результате всего этого естественные процессы сближения (интеграции) управления и информационных технологий начали активно развиваться. Интенсивно увеличивается площадь пересечения областей теории управления и информатики (рис. 3).

Дальнейшее развитие систем управления, связанное с их интеллектуализацией¹, происходило за счет активного использования вычислительной техники и различных технологий искусственного интеллекта, развиваемых начиная примерно с 1950-х годов в недрах информатики. Наиболее популярными технологиями искусственного интеллекта явились ситуационное управление, экспертные системы, нейронные сети, генетические (эволюционные) алгоритмы, многоагентные системы, системы, основанные на знаниях.

В результате можно считать, что информатика стимулировала развитие таких новых разделов интеллектуального управления, как ситуационное управление, нейроуправление, управление на основе эволюционных алгоритмов, на основе технологии многоагентных систем и на основе знаний (см. рис. 3).

В уже упомянутой работе С. Н. Васильева, А. К. Жерлова и др. среди систем управления на основе знаний выделены системы управления на основе правил, логических моделей, с применением автоматического доказательства теорем, на основе нечетких правил.

Принято считать, что системы интеллектуального управления обладают способностью к пониманию и обучению в отношении объекта управления, возмущений, внешней среды, условий работы, цели управления.

Относительно самостоятельно применительно к общественным и организационным системам развивается такое новое направление в управлении, как информационное «Под информационным управлением понимается процесс выработки и реализации управленческих решений в ситуации, когда управляющее воздействие носит неявный, косвенный ха-

*Информационное управление;
интеллектуальное управление;
ситуационное управление;
нейроуправление;
управление, основанное на знаниях;
управление на основе эволюционных алгоритмов;
многоагентное управление*



*Адаптивные компьютерные системы;
проактивные компьютерные системы;
адаптивное управление;
адаптивное предприятие*

■ Рис. 3. Интеграция областей информатики и теории управления

актер, и объекту управления представляется определяемая субъектом управления информация о ситуации (информационная картина), ориентируясь на которую, этот объект как бы самостоятельно выбирает линию своего поведения².

В свою очередь, управленческая терминология проникает в информатику и вычислительную технику. Сегодня весьма популярными в области вычислительной техники и информационных технологий становятся понятия и, соответственно, стратегии адаптивных и проактивных компьютерных систем³, адаптивного управления и адаптивного предприятия⁴. Эти стратегии интенсивно развиваются компаниями IBM, Intel Research, Hewlett Packard, Microsoft, Sun и др.

¹ Тимофеев А. В., Юсупов Р. М. Интеллектуальные системы управления // Изв. РАН. Технич. кибернетика. – 1994. – № 5; Васильев С. Н., Жерлов А. К., Федосов Е. А., Федунев Б. Е. Интеллектуальное управление динамическими системами. М.: Физматгиз, 2000.

² Виноградов С. М., Войтович Н. А., Вус М. А. и др. Информационное общество: информационные войны, информационное управление, информационная безопасность. – СПбГУ, 1999.

³ Вонт Р., Перинг Т., Тенненхау Д. Адаптивные и проактивные компьютерные системы // Открытые системы. – 2003, октябрь.

⁴ Черняк Л. От адаптивной инфраструктуры – к адаптивному предприятию // Открытые системы. – 2003, октябрь.

Адаптивные и проактивные компьютерные (proactive computing) системы призваны решить проблемы, ограничивающие развитие ИКТ на современном этапе, в частности, за счет придания системам таких адаптационных способностей, как самоконтроль, самовосстановление, самоконфигурирование, самооптимизация, самообслуживание, самоорганизация. Свойства проактивных систем расширяют наши представления о применении компьютеров за счет необходимости мониторинга окружающей среды (мира) и влияния на него. Одна из основных задач адаптивных систем и адаптивного управления – способность приспосабливаться к требованиям бизнеса.

Применительно к управлению бизнесом и соответствующими предприятиями сформировалось представление, получившее название RTE (Real Time Enterprise), т. е. предприятие, работающее в реальном времени¹. Для проектирования и создания таких предприятий разрабатываются новые языки моделирования бизнес-процессов и их интеграции, развиваются соответствующие Web-службы и Web-порталы, создаются средства общения на естественном языке человека с машиной и т. д.

Очевидно, что указанная тенденция естественно (естественно-стихийного) сближения информатики и управления должна быть поддержана организационно. Определенные движения в этом направлении уже намечились. Так, в Киеве выпускается международный журнал «Проблемы управления и информатики». В 1997 г. в Санкт-Петербурге по инициативе СПИИРАН была проведена Международная конференция «Информатика и управление». В 2003 г. по инициативе Санкт-Петербургского государственного электротехнического университета была проведена Всероссийская конференция «Управление и информационные технологии». Аналогичные конференции и семинары начали проводиться и в других научных центрах и регионах.

Таким образом, мы являемся свидетелями и участниками сближения теоретических и прикладных проблем управления и информатики. Возможно, это приведет к появлению определенного комплексного междисциплинарного научно-прикладного направления на стыке этих двух дисциплин (информатики и управления).

Вспомним, что в свое время кибернетика кроме винеровского определения как науки об управлении и связи в животном и машине имела и определения, являющиеся практически синонимами определения информатики.

Так, в «Энциклопедии» кибернетика определялась как «наука об общих законах получения, хранения, передачи и преобразования информации в сложных управляющих системах»².

В предисловии к книге Ф. Джорджа аналогично кибернетика определяется как «наука об общих законах получения, хранения, передачи и переработки информации»³.

В статье М. Г. Гаазе-Рапопорта дается следующее определение кибернетики: «Кибернетика – это научно-методологическое направление, рассматривающее весь объективно существующий мир с единой, а именно информационной, точки зрения»⁴.

Сегодня в условиях активной информатизации общества имеет место определенное отождествление понятий информационного и киберпространства.

Кстати, попытки отождествления в целом кибернетики и информатики делались и ранее. Так, в работе Б. В. Бирюкова, в частности, предлагалось определить информатику как кибернетику на современном этапе развития и ввести синтезированное научное направление «информатика – кибернетика»⁵.

В работах основоположников кибернетики отмечалось, что основными понятиями этой науки являются управление и информация.

С учетом всего сказанного происходящее явление сближения управления и информатики можно рассматривать как процесс возрождения кибернетики в условиях бурного развития вычислительной техники и инфотелекоммуникационных технологий.

Представляется, что применительно к этому процессу можно, пока несколько условно, говорить о рождении (возрождении) нового научно-технического направления – неокибернетики.

В определенной мере это утверждение может явиться ответом на вопрос, поставленный в дискуссии, открытой на страницах журнала «Проблемы управления и информатики» (№ 3, 2001 г.) статьей Н. Н. Дидука и В. Н. Коваля «Существует ли наука кибернетика? (О роли кибернетики в естествознании)». Наше предварительное мнение по этому вопросу было изложено в этом же журнале (№ 6, 2001 г.) в статье Р. И. Полонникова и Р. М. Юсупова «Восприимет ли кибернетику XXI век?»

Сегодня можно надеяться, что кибернетика как наука об общих законах управления, информации и информационного взаимодействия будет развиваться в рамках неокибернетики.

Р. М. Юсупов,

доктор тех. наук, профессор, директор СПИИРАН

¹ Черняк Л. На пути к предприятию, управляемому в реальном времени // Открытые системы. – 2002, декабрь.

² Энциклопедия кибернетики, Ч. I. – Киев. – 1974 г.

³ Джордж Ф. Основы кибернетики. – М.: Радио и связь, 1984.

⁴ Гаазе-Рапопорт М. Г. Куда идет кибернетика? // Кибернетика. Дела практические. – М.: Наука, 1984.

⁵ Бирюков Б. В. Кибернетика, информатика, вычислительная техника, автоматика: проблемы становления и развития. Вклад отечественной науки // Кибернетика. прошлое для будущего. – М.: Наука, 1989.

ПАМЯТИ ВИКТОРА ИЛЬИЧА ВАРШАВСКОГО (1933 – 2005)

Наука потеряла одного из выдающихся своих представителей. 3 января 2005 года в Тель-Авиве после тяжелой продолжительной болезни на 72-м году жизни скончался доктор технических наук, профессор Виктор Ильич Варшавский.

Виктор Ильич Варшавский родился в Ленинграде в 1933 году в семье известного советского писателя-фантаста Ильи Варшавского. После окончания школы поступил в Ленинградский институт точной механики и оптики, который успешно закончил в 1956 году, получив диплом инженера по специальности «Приборы управления стрельбой». С 1956 по 1960 годы работал в должности инженера и руководителя исследовательской группы Научно-исследовательского института Министерства судостроительной промышленности в Ленинграде.

В 1962 году в Ленинградском институте авиационного приборостроения защитил диссертацию на соискание ученой степени кандидата технических наук по специальности «Техническая кибернетика» и с 1963 года был постоянным членом комиссии Академии наук СССР по «Теории релейных устройств и конечных автоматов».

В 1970 году защитил диссертацию на соискание ученой степени доктора технических наук. Длительное время являлся заведующим лабораторией кибернетики и заведующим отделом Ленинградского отделения центрального экономико-математического института АН СССР, а также профессором кафедры вычислительной техники и затем, вплоть до 1993 года, кафедры математического обеспечения ЭВМ Ленинградского электротехнического института.

С 1988 по 1993 годы Виктор Ильич Варшавский работал в должности научного руководителя, а также в должности директора по научно-исследовательской работе в Научно-исследовательском кооперативе «Трасса», занимавшемся разработкой асинхронных систем по договорам с рядом институтов АН СССР и отраслевых НИИ.

В 1993 году В. И. Варшавский вместе с группой своих учеников и соратников был приглашен на работу во вновь созданный международный университет города Айзу-Вакамацу в Японии, где проработал до 2000 года в должности профессора и заведующего лабораторией проектирования компьютерной логики.

В 2002–2003 годах В. И. Варшавский – заведующий отделом логического управления в фирме «Технологии нейронных сетей» (Тель-Авив, Израиль).

С 1985 года В. И. Варшавский – Член Совета Академии Наук СССР по искусственному интеллекту, с 1990 года – член ACM (Association For Computing Machinery). Виктор Ильич был награжден Почетным знаком «Изобретатель СССР», а в 1988 году Золотой медалью ВДНХ за победу в соревновании на лучший проект по микроэлектронике.

Научные труды В. И. Варшавского сыграли большую роль в становлении кибернетики и исследова-

ний в области искусственного интеллекта в нашей стране. В начале 1960-х годов он вместе с М. Л. Цетлиным начал проводить зимние школы под Ленинградом, привлекая представительную группу ученых, специализировавшихся на исследованиях в области математики, биологии, теории автоматов, в области коллективного поведения, распознавания образов, вычислительной техники, передачи информации, а также в смежных областях. Несомненно, что ценность и актуальность междисциплинарных проблем В. И. Варшавский и М. Л. Цетлин почувствовали раньше других.

Позднее интересы В. И. Варшавского переключились на новую малоисследованную тогда область – создания асинхронных электронных устройств и систем, т. е. систем, в которых не используются «часы». Такие системы обладают рядом полезных свойств, однако методы их построения были совершенно неясны. В. И. Варшавский стал пионером в данной области исследований, что широко признано в асинхронном сообществе за рубежом.

В. И. Варшавский был выдающимся схемотехником и изобретателем. Он имеет более 150 авторских свидетельств СССР, патентов Японии и США, заявок на патенты. Одного перечисления его схемотехнического вклада в асинхронику было бы достаточно, чтобы причислить Виктора Ильича к ее элите. Но нельзя забыть и его вклад в теоретические аспекты этой науки. Он автор, соавтор и редактор восьми книг, из которых пять посвящены асинхронике (три из них – на английском языке), более 200 научных статей и отчетов (в последних он выступал в роли научного руководителя).

Кроме асинхроники, достаточно упомянуть ставшие классическими научные результаты в исследовании пороговой логики и искусственных нейронов, которыми В. И. Варшавский занимался в начале своей научной карьеры и к изучению которых вернулся в последние годы, коллективного поведения автоматов, ставшего темой докторской диссертации Виктора Ильича. Интересно заметить, что в последнее время заметно вырос интерес к поведению стохастических автоматов с переменной структурой.

В. И. Варшавский был не только выдающимся ученым, но и замечательным организатором и наставником, умевшим сплотить вокруг себя активный творческий коллектив. На всех этапах своей жизни Виктор Ильич, всячески помогая ученикам, не переставал учиться и сам. Никогда не стремясь к «деланию карьеры», он не занимал высоких административных должностей, главным для него было – творить, и это не могло не вдохновлять нас – его учеников и друзей. Мы будем помнить его всегда.

Л. Розенблюм, М. Кишиневский, А. Кондратьев, О. Маевский, Ю. Мамруков, В. Мараховский, Н. Стародубцев, А. Таубин, А. Яковлев.

Министерство образования и науки Российской Федерации
 Национальная Комиссия РФ по делам ЮНЕСКО
 Комитет по науке и высшей школе Правительства Санкт-Петербурга
 Совет ректоров вузов Санкт-Петербурга
 Государственный Университет Аэрокосмического Приборостроения (ГУАП)
 Кафедра ЮНЕСКО «Дистанционное инженерное образование» ГУАП
 Институт развития дополнительного профессионального образования
 Журнал «Информационно-управляющие системы»



МЕЖДУНАРОДНЫЙ СЕМИНАР «ОБРАЗОВАНИЕ ДЛЯ ВСЕХ»

27 июня – 01 июля 2005 года

Место проведения семинара: комфортабельный теплоход «Санкт-Петербург», который совершит круиз из Санкт-Петербурга по рекам Нева и Свирь, Ладожскому и Онежскому озерам

В работе семинара примут участие более 200 ученых и преподавателей из России, стран СНГ, Европы, Америки и Азии.

Круглые столы

Роль ЮНЕСКО в активизации интеграции университетов в программу «Образование для всех».

Председатели: г. Хаддад, директор сектора высшего образования ЮНЕСКО, Г. Орджоникидзе, ответственный секретарь комиссии РФ по делам ЮНЕСКО

Создание международных университетских исследовательских комплексов.

Председатели: К. Гилерма, директор высшей инженерной школы Руана ESIGELEC, Г. Коржавин, генеральный директор ФГУП ЦНИИ «Гранит».

Информационно-коммуникационные технологии в науке и образовании.

Председатель: Ф. Кео, директор московского бюро ЮНЕСКО.

Особенности образования взрослых.

Председатели: В. Безлепкин, директор Института дополнительного профессионального образования, В. Кинелев, директор Института информационных технологий в образовании ЮНЕСКО.

Проблемы и перспективы системы либерального образования в XXI веке.

Председатели: Дж. Беккер, декан факультета по международному образованию, Бардколледж, В. Монахов, директор Смольного института свободных искусств и наук.

Качество образования – как основное направление программы «образование для всех».

Председатели: Дж. Эклоф, профессор Стокгольмской Школы Экономики, Е. Горбашко, профессор Санкт-Петербургского государственного университета экономики и финансов.

Теоретические основы и международный опыт борьбы с терроризмом.

Председатели: В. Боер, декан факультета, профессор Санкт-Петербургского государственного университета аэрокосмического приборостроения, О. Михно, президент Института культуры «Аргентина – Россия».

Перспективы Российско-Латино-Американского сотрудничества в области образования, науки и культуры.

Председатели: Дж. Пуглииз, заместитель министра, Министерство образования, науки и технологий Ар-

гентины, А. Самойлов, проректор Санкт-Петербургского государственного университета аэрокосмического приборостроения.

Правила оформления тезисов

Параметры редактора Word для Windows: размер бумаги – А4, поля – 2,5 см, ориентация книжная, шрифт – Times New Roman, кегль (высота шрифта) – 12, красная строка – 1 см, межстрочное расстояние – одинарное, выравнивание – по ширине, расстановка переносов – автоматическая. Тексты тезисов представляются на русском и английском языках.

Порядок формирования текста тезисов: название доклада, инициалы, фамилия, ученая степень и должность автора, страна, город, название организации, текст тезисов

Контрольные сроки

Все материалы должны быть представлены на дискете 3,5" или по электронной почте int@aanet.ru не позднее **10 мая 2005 г.**

Культурная программа

Свирьстрой: автобусная экскурсия в Свято-Троицкий собор Александра-Свирского мужского монастыря

Остров Кижь: пешеходная экскурсия «Заповедник Кижь»

Остров Валаам: пешеходная экскурсия «Валаамский монастырь»

Условия участия

Стоимость участия в семинаре для российских участников – 450 евро (оплата производится в рублях по курсу ЦБ РФ на день выставления счета). Данная стоимость включает в себя регистрационный взнос, публикацию тезисов докладов, проживание на теплоходе, трехразовое питание, экскурсионную и культурную программу во время круиза.

Дополнительная информация и справки

190000, Санкт-Петербург, Большая Морская ул., 67
 Кафедра ЮНЕСКО «Дистанционное инженерное образование» ГУАП

Оргкомитет международного семинара

Тел (812) 312-09-37

Факс (812) 315-77-78

E-mail: int@aanet.ru

http://www.quap.ru/conf/voxi/voxi_ru.html

**БЕЛОГОЛОВЫЙ
Андрей
Владимирович**



Ассистент кафедры безопасности информационных систем Санкт-Петербургского государственного университета аэрокосмического приборостроения.

В 2000 году окончил Санкт-Петербургский государственный университет аэрокосмического приборостроения

В 2004 году защитил диссертацию на соискание ученой степени кандидата технических наук. Является автором 14 научных публикаций.

Область научных интересов – теория кодирования, криптография, сжатие изображений.

**ГАГАРИН
Константин
Юрьевич**



Докторант Санкт-Петербургского государственного политехнического университета. В 1990 году окончил Ленинградский механический институт. В 1996 году защитил диссертацию на соискание ученой степени кандидата технических наук.

Является автором 20 научных публикаций.

Область научных интересов – цифровая обработка сигналов.

**ЗИКРАТОВ
Игорь
Алексеевич**



Начальник научно-исследовательского отдела Военного университета противовоздушной обороны (филиал, г. Санкт-Петербург). Доцент Военного университета противовоздушной обороны.

В 1987 году окончил Киевское высшее инженерное радиотехническое училище противовоздушной обороны им. Маршала авиации А. И. Покрышкина

В 1999 году защитил диссертацию на соискание ученой степени кандидата технических наук. Является автором 30 научных публикаций.

Область научных интересов – прикладные вопросы электродинамики.

**КОРНЕЕВ
Георгий
Александрович**



Аспирант Санкт-Петербургского государственного университета информационных технологий, механики и оптики.

В 2004 году окончил Санкт-Петербургский государственный университет информационных технологий, механики и оптики. Является автором шести научных публикаций.

Область научных интересов – технологии программирования, теория программирования, конечные автоматы, объектно-ориентированное проектирование.

**КРУК
Евгений
Аврамович**



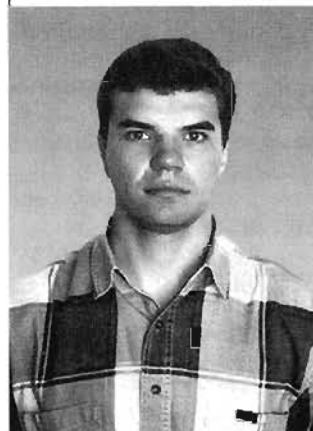
Профессор, заведующий кафедрой безопасности информационных систем Санкт-Петербургского государственного университета аэрокосмического приборостроения

В 1974 году окончил Санкт-Петербургский государственный университет аэрокосмического приборостроения.

В 1999 году защитил диссертацию на соискание ученой степени доктора технических наук. Является автором 90 научных публикаций.

Область научных интересов – теория кодирования, криптография.

**МЕЛЬНИК
Игорь
Александрович**



Ведущий инженер-программист ООО НПФ «Беркут».

В 2001 году окончил Санкт-Петербургский государственный университет аэрокосмического приборостроения.

Является автором двух научных публикаций.

Область научных интересов – сети передачи данных.

ОВЧИННИКОВ
Андрей
Анатольевич



Ассистент кафедры безопасности информационных систем Санкт-Петербургского государственного университета аэрокосмического приборостроения.
В 2000 году окончил Санкт-Петербургский государственный университет аэрокосмического приборостроения.
В 2004 году защитил диссертацию на соискание ученой степени кандидата технических наук. Является автором десяти научных публикаций.
Область научных интересов – теория кодирования, криптография

СТЕПАНЕНКО
Кирилл
Васильевич



Адъюнкт очной адъюнктуры Санкт-Петербургского филиала Военного университета противовоздушной обороны.
В 2000 году окончил Военный университет противовоздушной обороны (филиал, г. Санкт-Петербург).
Является автором шести научных публикаций.
Область научных интересов – прикладные вопросы электродинамики.

ТРИФОНОВ
Петр
Владимирович



Аспирант Санкт-Петербургского государственного политехнического университета.
В 2003 году окончил Санкт-Петербургский государственный политехнический университет.
Является автором одиннадцати научных публикаций.
Область научных интересов – цифровая связь, помехоустойчивое кодирование, распределенные системы.

ШАЛЫТО
Анатолий
Абрамович



Заведующий кафедрой технологий программирования Санкт-Петербургского государственного университета информационных технологий, механики и оптики. Ученый секретарь НПО "Аврора".
В 1971 году окончил Ленинградский электротехнический институт по специальности "Автоматика и телемеханика".
В 1999 году защитил диссертацию на соискание ученой степени доктора технических наук.
Является автором более 250 научных публикаций, трех монографий и 70 изобретений.
Член редакционной коллегии журнала "Информационно-управляющие системы".
Область научных интересов – системы логического управления, автоматное программирование.

ШАМГУНОВ
Никита
Назимович



Аспирант Санкт-Петербургского государственного университета информационных технологий, механики и оптики.
В 2001 году окончил Уральский государственный университет.
Является автором пяти научных публикаций.
Область научных интересов – технологии программирования, теория программирования, конечные автоматы, объектно-ориентированное проектирование.

УДК 621.3

Математические модели и быстрые алгоритмы векторно-разностного кодирования цифровых речевых сигналов

Гагарин К. Ю. – Информационно-управляющие системы, 2005. – № 1. – С. 2–9.

Представлены математические модели и алгоритмы векторно-разностных схем кодирования речевых сигналов на основе многоканальных линейных систем. Предложены алгоритмы вычисления автокорреляционных матриц через быстрые гиперкомплексные преобразования Фурье.

Список лит.: 10 назв.

УДК 621.396

Обоснование масштаба цифровых карт местности, используемых при расчете напряженности поля радиосигналов

Зикратов И. А., Степаненко К. В. – Информационно-управляющие системы, 2005. – № 1. – С. 10–15.

Предложен подход, основанный на теории случайных полей, позволяющий обосновать полноту представления данных о подстилающей поверхности типа «шероховатый рельеф» в цифровых картах местности, используемых в специализированных геоинформационных системах проектирования радиолиний. Приведены примеры корреляционной связи результатов расчета поля рассеяния с параметрами цифровой модели местности.

Список лит.: 10 назв.

УДК 681.3.06

State Machine – расширение языка Java для эффективной реализации автоматов

Шамгунов Н. Н., Корнеев Г. А., Шалыто А. А. – Информационно-управляющие системы, 2005. – № 1. – С. 16–24.

В данной статье предлагается новый язык объектно-ориентированного программирования State Machine, являющийся расширением языка программирования Java. В язык вводится ряд синтаксических конструкций, позволяющих программировать с использованием понятий автомат, состояние и событие. Для обработки новых синтаксических конструкций разработан препроцессор, преобразующий код на языке State Machine в код на языке Java. При этом новые синтаксические конструкции преобразуются в код на языке Java в соответствии с предложенным ранее авторами паттерном State Machine.

Список лит.: 18 назв.

УДК 621.3

Mathematical models and fast algorithms of vector-differential coding of digital speech signals

Gagarin K. Yu. – IUS, 2005. – N 1. – P. 2–9.

In this paper we present a methods of fast Fourier transform algorithmic design in hypercomplex algebras with multiplicative groups of imaginary orts. These orts are an extension of rational quantities field.

Refs: 10 titles.

УДК 621.396

The digital landscape maps scale substantiation, used for calculation of radio signals field intensity

Zikratov I. V., Stepanenko K. V. – IUS, 2005. – N1. – P. 10–15.

The approach based on the theory of casual field is proposed. It allows to prove completeness of data presentation about a spreading surface such as «rough relief» on the digital landscape maps used in specialized geoinformation systems of radiolines designing. The examples of correlation connection of calculation results of dispersion field with parameters of digital landscape model are perfomed.

Refs: 10 titles.

УДК 681.3.06

State Machine – Java language expansion for effective realisation of automatic devices

Shamgunov N. N., Korneev G. A., Shalyto A. A. – IUS, 2005. – N 1. – P. 16–24.

This paper presents a new Object-oriented programming language State Machine which is an extension of the Java language. This language presents a number of syntactic constructions that provides an ability to create programs with using notion of automaton, state and event. Author developed a preprocessor that transforms State Machine code to Java code. This preprocessor transforms new syntactic constructions to Java code according to State Machine pattern which was presented by these authors earlier.

Refs: 18 titles.

УДК 621.391.251

Многopopоговое декодирование кодов с низкой плотностью проверок на четность

Белоголовый А. В., Крук Е. А. – Информационно-управляющие системы, 2005. – № 1. – С. 25–31.

В статье предлагается новый алгоритм декодирования. Новый алгоритм принадлежит к классу быстрых алгоритмов, но благодаря использованию порогов при декодировании качество работы приближается к результатам медленных декодеров. В работе приводится описание алгоритма и сравнение его с традиционными и быстрыми декодерами на основе моделирования.

Список лит.: 8 назв.

УДК 621.391.251

К вопросу о построении LDPC-кодов на основе Евклидовых геометрий

Овчинников А. А. – Информационно-управляющие системы, 2005. – № 1. – С. 32–40.

В данной статье рассматриваются коды с малой плотностью проверок на четность, основанные на конечных Евклидовых геометриях. Для ряда таких кодов получены оценки их минимального расстояния и спектра, предложены процедуры укорочения Евклидово-геометрических конструкций и оценки расстояния полученных кодов. Приведены результаты моделирования рассматриваемых кодов в канале с аддитивным белым гауссовским шумом (АБГШ).

Список лит.: 22 назв.

УДК 20.53.23; 49.31.00

Адаптивная передача в многопользовательских многочастотных системах вещания

Трифонов П. В. – Информационно-управляющие системы, 2005. – № 1. – С. 41–45.

Рассматривается задача оптимизации схемы передачи в цифровых многопользовательских системах вещания. Показано, что путем реализации управляемого разделения канала может быть достигнуто существенное снижение мощности передатчика, требуемой для достижения заданных параметров системы. Рассматриваются различные варианты реализации этого подхода. Исследуется влияние стохастических свойств радиоканала на эффективность метода.

Список лит.: 6 назв.

УДК 621.391.251

Multi-threshold decoding of low density parity check codes

Belogoloviy A. V., Kruk E. A. – IUS, 2005. – N 1. – P. 25–31.

In the paper the new decoding algorithm is proposed. The new algorithm belongs to fast decoding algorithms class, but due to the threshold use in decoding the performance becomes closer to slow decoders results. The algorithm description is shown and traditional and fast decoders based on simulations are compared in the paper.

Refs: 8 titles.

УДК 621.391.251

The problem of the LDPC-codes constructing on the base of Euclidean geometries

Ovchinnikov A. A. – IUS, 2005. – N 1. – P. 32–40.

In this paper we consider the low-density parity-check codes based on final Euclidean geometries. For the number of such codes the estimations of their minimal distance and spectrum are obtained. The shortening procedures and the estimations of codes distance are suggested. The simulation results in channel with AWGN are presented.

Refs: 22 titles.

УДК 20.53.23; 49.31.00

Adaptive data transmission in multi-frequency broadcast systems

Trifonov P. V. – IUS, 2005. – N 1. – P. 41–45.

Optimization of data transmission scheme in digital multi-user multi-frequency broadcast systems is considered. It is shown that optimization of channel sharing can provide considerable reduction of transmitters power required for achieving the target system parameters. Different approaches for this are considered. The influence of stochastic characteristics of channel on the performance of the suggested method effectiveness is analyzed.

Refs: 6 titles.

УДК 004.3.067

Новый метод оценки алгоритмов распределения OVSF кодов в стандарте WCDMA

Мельник И. А. – Информационно-управляющие системы, 2005. – № 1. – С. 46–50.

OVSF (Orthogonal Variable Spreading Factor) коды используются для поддержки разных скоростей передачи данных для различных пользователей в WCDMA системах. В данной работе, проанализированы существующие алгоритмы распределения кодов и решаемые ими проблемы. Для того чтобы оценить эффективность алгоритмов мы предлагаем метод, который учитывает возникающие блокировки между кодами во время распределения. На базе этого метода можно построить аналитические модели для оценки алгоритмов.

Список лит.: 10 назв.

UDK 004.3.067

A new estimation method of OVSF code allocation algorithms for WCDMA systems

Melnik I. A. – IUS, 2005. – N 1. – P. 46–50.

Orthogonal Variable Spreading Factor (OVSF) codes are used to support different transmission rates for different users in WCDMA systems. In this paper, we analyze existing code allocation algorithms and problems that they have to solve. To estimate the efficiency of the algorithms we propose a method, which consider blockings between codes during assignment. On basis of this method it is possible to build analytical models for an estimation of different algorithms.

Refs: 10 titles.

ПАМЯТКА ДЛЯ АВТОРОВ

Поступающие в редакцию статьи проходят обязательное рецензирование.

При наличии положительной рецензии статья редактируется и рассматривается редакционной коллегией. Принятая в печать статья направляется автору для согласования редакторских правок. После согласования автор представляет в редакцию окончательный вариант текста статьи, а также фотографию и краткое изложение сведений о себе.

Процедуры согласования текста статьи, предоставления фото (размером 4×5,5 см) и сведений об авторе могут осуществляться как непосредственно в редакции, так и по e-mail (электронный вариант фото в виде файла *.tif, *.jpg с разрешением 300 dpi).

При отклонении статьи редакция представляет автору мотивированное заключение и рецензию. При необходимости доработать статью — рецензию.

Редакция журнала напоминает, что ответственность за подбор, достоверность и точность фактов, экономико-статистических и технических показателей, собственных имен и прочих сведений, а также за то, что в материалах не содержится сведений, не подлежащих открытой публикации, несут авторы публикуемых в журнале материалов и рекламодатели.



190000, С.-Петербург,
Б.Морская, 67,
ГУАП, ИКБСС
Контактный адрес:
e-mail: ekrouk@vu.spb.ru

Институт компьютерной безопасности вычислительных систем и сетей (ИКБСС) создан в С.-Петербургском государственном университете аэрокосмического приборостроения на базе кафедры Безопасности информационных систем в 2003 году.



Сегодня институт объединяет более 20 сотрудников, среди которых 2 доктора и 6 кандидатов технических наук. Институт ведет исследования по широкому кругу вопросов создания безопасных информационных технологий.

Основными направлениями его работы являются:

- | Системы с публичным ключом
- | Цифровые средства защиты авторских прав
- | Распределение ключей в многоуровневых системах
- | Использование помехоустойчивого кодирования при передаче, хранении и защите информации
- | Протоколы обмена и защиты информации в сенсорных сетях

В институте успешно ведутся исследовательские проекты в рамках сотрудничества с такими ведущими компьютерными и телекоммуникационными фирмами мира, как Intel, Samsung, Siemens.

Институт компьютерной безопасности вычислительных систем и сетей приглашает к сотрудничеству все заинтересованные фирмы и организации.

**ИНСТИТУТ КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ
ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ**