ISSN 1816-0301

ИНФОРМАТИКА

3 (39)

ИЮЛЬ-СЕНТЯБРЬ
2013

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

ОБЪЕДИНЕННЫЙ ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ
НАЦИОНАЛЬНОЙ АКАДЕМИИ НАУК БЕЛАРУСИ

ИНФОРМАТИКА

ЕЖЕКВАРТАЛЬНЫЙ НАУЧНЫЙ ЖУРНАЛ

Издается с января 2004 г.

№ 3(39) • июль-сентябрь 2013

СОДЕРЖАНИЕ

ЗАЩИТА ИНФОРМАЦИИ

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

Сдано в набор 01.07.2013. Подписано в печать 15.08.2013.
Формат 60×84 1/8. Бумага офсетная. Гарнитура Таймс.
Усл. печ. л. 16,0. Уч.-изд. л. 15,7. Тираж 100 экз. Заказ 8.

Государственное научное учреждение «Объединенный институт проблем информатики Национальной академии наук Беларуси».
ЛИ № 02330/0549421 от 08.04.2009.
Ул. Сурганова, 6, 220012, Минск.

Отпечатано с оригинала-макета на ризографе Объединенного института проблем информатики Национальной академии наук Беларуси.
Ул. Сурганова, 6, 220012, Минск.

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

INFORMATICS

PUBLISHED QUATERLY

Issued since 2004

№ 3(39) • July-September 2013

CONTENTS

INFORMATION PROTECTION

ОБРАБОТКА СИГНАЛОВ, ИЗОБРАЖЕНИЙ И РЕЧИ

УДК 621.391.268

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

АЛГОРИТМ СОПРОВОЖДЕНИЯ ОБЪЕКТОВ В ОПТИКО-ЭЛЕКТРОННЫХ СИСТЕМАХ НА ОСНОВЕ МЕТОДА НАИМЕНЬШИХ КВАДРАТОВ

Рассматривается алгоритм сопровождения изображений объектов в оптико-электронных системах (ОЭС) на основе метода наименьших квадратов (МНК) при использовании предположения о гладкости траектории движения и скорости ее изменения. Особенность решаемой задачи состоит в оценке не только координат траектории, но и ее скорости, что ранее в МНК не рассматривалось. На конкретном примере дается сравнительная оценка дисперсий ошибок сопровождения для фильтра Калмана и полученного алгоритма.

Введение

Пассивные ОЭС видимого и инфракрасного диапазонов с матричными фотоприемными устройствами (ФПУ) используются для решения задач наблюдения и наведения воздушных объектов. Они подразделяются на системы сопровождения и обзорно-поисковые [1]. Конечной целью обработки видеоинформации в них является построение траекторий движения обнаруженных объектов [2, 3] на основе измерений их текущих координат. Эта процедура называется сопровождением, для чего используются соответствующие алгоритмы.

В настоящее время основными методами нахождения алгоритмов сопровождения являются фильтр Калмана (ФК) [4] и метод максимального правдоподобия (МП) [5]. Наивысшей точностью по критерию минимума дисперсий ошибок сопровождения обладает ФК, однако для своей реализации он требует знания модели движения объектов. Поскольку траектории движения и условия измерений весьма разнообразны, обоснованное нахождение таких моделей проблематично.

Алгоритмы сопровождения на основе МП обладают худшей точностью в указанном смысле, однако используют значительно меньший объем априорной информации. В то же время такие алгоритмы требуют больших вычислительных затрат, поскольку приходится решать нелинейные уравнения посредством итерационных процедур, что затрудняет их использование в реальном масштабе времени.

Более практичной основой построения алгоритма сопровождения является МНК [6, 7]. С его помощью удается получать алгоритмы в явной форме записи и для их нахождения использовать минимум априорной информации о траекториях движения. Она состоит в требованиях гладкости траекторий, что соответствует физическим свойствам объекта сопровождения. Однако малый объем априорной информации приводит к тому, что задача нахождения алгоритма сопровождения становится неоднозначной и для получения решения требуется использовать способы регуляризации [8]. Особенностью применения МНК в ОЭС является не только необходимость построения траекторий движения, но и прогнозирование их значений на следующий период измерений. Это делается для селекции координат сопровождаемого объекта относительно посторонних посредством формирования строба сопровождения на последующем кадре изображения [1]. В основе прогнозирования лежит необходимость оценки скорости изменения траектории. В настоящее время решение задачи построения траектории движения с одновременным определением скорости ее изменения на основе МНК отсутствует.

Целью работы является изложение метода нахождения алгоритмов сопровождения в ОЭС на основе МНК с одновременной оценкой скорости изменения координат.

1. Алгоритм сопровождения на основе МНК

МНК использует эмпирические, а не статистические характеристики траекторий движения объектов и ошибок измерений [5, 6]. Рассмотрим метод нахождения алгоритмов сопровождения с учетом следующих предположений:

1. Полагаем, что априорные сведения о траектории движения состоят в том, что она и скорость ее изменения считаются функциями непрерывными во времени и гладкими. Гладкость понимается в том смысле, что на интервалах времени порядка нескольких периодов измерений она может быть аппроксимирована полиномиальной функцией.

2. Траектория отображается дискретно в прямоугольной системе координат (r_1, r_2), связанной с плоскостью матричного детектора ФПУ. Проекции траектории на эти оси полагаем статистически независимыми процессами, которые отображаются посредством координат r_k в дискретные моменты времени k. Помимо них интерес представляют значения скорости $\vartheta_k = (r_k - r_{k-1})/T$, где T – период измерений.

3. Измерения возможны только для координат и производятся в соответствии с линейной моделью $z_k = hr_k + v_k$, где h – масштабный множитель, отражающий передаточные функции измерителя. Здесь не указаны индексы проекций, так как дальнейшее изложение справедливо для обеих из них. Ошибки измерений v_k считаются статистически независимыми случайными величинами с нулевым математическим ожиданием. Оценки координат и скорости по результатам измерений обозначаются символами $\hat{r}_k$ и $\hat{\vartheta}_k$.

В МНК получение данных оценок осуществляется путем минимизации функционала качества $Q_k(r_k, \vartheta_k)$, зависящего от квадрата невязки $(z_k - hr_k)^2$, а также условий гладкости траектории и ее скорости, представленных в квадратичной форме. Предлагается следующий вариант функционала:

$$Q_k(r_k, \vartheta_k) = (z_k - hr_k)^2 + \alpha_1 T^2 \vartheta_k^2 + \alpha_2 T^2 (\vartheta_k - \vartheta_{k-1})^2. \quad (1)$$

В этом выражении первое слагаемое определяется невязкой решения, второе – условием гладкости траектории, третье – условием гладкости скорости. Весовые коэффициенты α_1 и α_2 играют роль коэффициентов регуляризации [8].

Искомая оптимальная оценка координаты $\hat{r}_k$ находится дифференцированием функционала (1) по r_k, приравниванием результата к нулю и решением уравнения

$$\left.\frac{\partial Q_k(r_k, \vartheta_k)}{\partial r_k}\right|_{\substack{r=\hat{r} \\ \vartheta=\hat{\vartheta}}} = h^2\hat{r}_k - hz_k + \alpha_1\hat{r}_k - \alpha_1\hat{r}_{k-1} + \alpha_2\hat{r}_k - \alpha_2\hat{r}_{k-1} - \alpha_2 T\hat{\vartheta}_{k-1} = 0,$$

которое дает следующий алгоритм оптимальной МНК-оценки координаты:

$$\hat{r}_k = \hat{r}_{k-1} + K_1(z_k - h\hat{r}_{k-1}) + K_2 T\hat{\vartheta}_{k-1}. \quad (2)$$

Здесь использованы обозначения

$$K_1 = \frac{h}{h^2 + \alpha_1 + \alpha_2}, \quad K_2 = \frac{\alpha_2}{h^2 + \alpha_1 + \alpha_2}. \quad (3)$$

Для получения оптимальной оценки скорости $\hat{\vartheta}_k$ в функционале (1) вместо величины r_k используют выражение $r_k = \vartheta_{k-1} T + r_{k-1}$. Оценка скорости находится дифференцированием (1) по ϑ_k, что приводит к уравнению

$$\left.\frac{\partial Q_k\left(r_k,\vartheta_k\right)}{\partial\vartheta_k}\right|_{\substack{r=\hat{r}\\ \vartheta=\hat{\vartheta}}} = h^2T\hat{\vartheta}_k + h^2\hat{r}_{k-1} - hz_k + \left(\alpha_1+\alpha_2\right)T\hat{\vartheta}_k - \alpha_2T\hat{\vartheta}_{k-1} = 0\,.$$

Его решение с учетом выражений (3) дает следующий алгоритм оценки скорости:

$$\hat{\vartheta}_k = K_2\hat{\vartheta}_{k-1} + \frac{K_1}{T}\left(z_k - h\hat{r}_{k-1}\right). \tag{4}$$

Уравнения (2) и (4) образуют алгоритм сопровождения МНК. Для определения коэффициентов регуляризации α_1 и α_2 может быть использован ряд подходов, описанных в [8]. В настоящей работе предлагается иной способ, соответствующий смыслу решаемой задачи. Предположим, что координата изменяется по линейному закону $r_k = \vartheta kT$ со скоростью ϑ и измеряется без ошибок, т. е. $z_k = h\vartheta kT$. Используя алгоритм (2), в установившемся режиме найдем ошибку оценки координаты $e_{rk} = r_k - \hat{r}_k$, которая называется динамической ошибкой по скорости [9]. Выходной процесс ищем в виде функции $\hat{r}_k = \vartheta kT + e_r$ с постоянной динамической ошибкой e_r. Подставляя функции z_k и $\hat{r}_k$ в формулу (2), можно получить следующее выражение для динамической ошибки:

$$e_r = \vartheta T\,\frac{K_1h + K_2 - 1}{K_1h}\,.$$

Отсюда видно, что динамическая ошибка будет стремиться к нулю, если величина $K_1h + K_2 \to 1$. С помощью (3) это условие можно выразить через коэффициенты α_1 и α_2 следующим образом: $e_r \to 0$, если $\dfrac{h^2+\alpha_2}{h^2+\alpha_1+\alpha_2} \to 1$. Условие выполняется, когда коэффициенты $\alpha_1, \alpha_2 \to 0$. Таким образом, динамическая ошибка уменьшается с уменьшением величины коэффициентов регуляризации.

Теперь предположим, что на вход поступает только дискретный белый шум ошибок измерений, т. е. $z_k = v_k$ с нулевым математическим ожиданием и постоянной дисперсией σ_v^2. Подставляя z_k в формулу (2), возводя обе ее части в квадрат и усредняя слагаемые, находим, что величина дисперсии случайной ошибки $\sigma_{erk}^2 = \left\langle \hat{r}_k^2 \right\rangle$ на выходе в установившемся режиме определяется выражением

$$\sigma_{er}^2 = \frac{K_1^2\sigma_v^2 - 2\left(1 - K_1h + K_2T\right)K_2T\left\langle \hat{r}_{k-1}\cdot\hat{r}_{k-2}\right\rangle}{1-\left(1-K_1h+K_2T\right)-K_2^2T^2}\,.$$

В этом выражении величина $\left\langle \hat{r}_{k-1}\cdot\hat{r}_{k-2}\right\rangle$ есть взаимная дисперсия случайной ошибки в соседние моменты времени $k-1$ и $k-2$. Поскольку алгоритм используется для выделения гладкого входного сигнала, можно полагать, что случайная ошибка также сильно сглаживается и за период измерений взаимная дисперсия будет величиной положительной. При этом предположении имеем верхнюю оценку дисперсии случайной ошибки в виде неравенства

$$\sigma_{er}^2 < \frac{K_1^2\sigma_v^2}{1-\left(1-K_1h+K_2T\right)-K_2^2T^2}\,.$$

Дисперсия будет стремиться к нулю, если $K_1 \to 0$. С учетом обозначений (3) уменьшение дисперсии можно выразить условием $h^2\big/\left(h^2+\alpha_1+\alpha_2\right)^2 \to 0$. Отсюда следует, что для снижения величины дисперсии случайной ошибки необходимо выполнение одного из условий: $\alpha_1 \to \infty$ или $\alpha_2 \to \infty$.

Таким образом, выбор коэффициентов регуляризации α_1 и α_2 носит компромиссный характер с точки зрения уменьшения как динамических, так и случайных ошибок. Обеспечим этот компромисс следующим образом.

Если $\alpha_1 \to \infty$, $\alpha_2 = \text{const}$, то из (2) и (3) следует $\hat{r}_k \to \hat{r}_{k-1}$. Если $\alpha_2 \to \infty$ при $\alpha_1 = \text{const}$, то $\hat{r}_k \to \hat{r}_{k-1} + \hat{\vartheta}_{k-1}T$. В случае когда $\alpha_1 \to 0$ и $\alpha_2 \to 0$, оценка $\hat{r}_k \to z_k/h$. Выберем алгоритм оценки в виде среднего значения этих величин. В итоге алгоритм МНК оценки координаты принимает вид

$$\hat{r}_k = \hat{r}_{k-1} + \frac{1}{3h}\left(z_k - h\hat{r}_{k-1}\right) + \frac{T}{3}\hat{\vartheta}_{k-1}, \tag{5}$$

что соответствует значениям $\alpha_1 = \alpha_2 = h^2$.

Алгоритм оценки скорости (4) приводится к выражению

$$\hat{\vartheta}_k = \frac{1}{3}\hat{\vartheta}_{k-1} + \frac{1}{3Th}\left(z_k - h\hat{r}_{k-1}\right). \tag{6}$$

При исследовании влияния сглаживания скорости на величину дисперсии ошибки представляет интерес частный случай, когда учитывается сглаживание только координат ($\alpha_2 = 0$). Для него алгоритм МНК (2) приводится к виду

$$\hat{r}_k = \hat{r}_{k-1} + \frac{h}{h^2 + \alpha_1}\left(z_k - h\hat{r}_{k-1}\right). \tag{7}$$

При нахождении коэффициента регуляризации α_1 используем подход, аналогичный описанному выше. Так, при $\alpha_1 \to \infty$ $\hat{r}_k \to \hat{r}_{k-1}$, а при $\alpha_1 \to 0$ $\hat{r}_k \to \hat{r}_{k-1} + \left(z_k - h\hat{r}_{k-1}\right)/h$. В качестве решения выбираем среднее значение этих величин и получаем алгоритм МНК в виде

$$\hat{r}_k = \hat{r}_{k-1} + \frac{1}{2h}\left(z_k - h\hat{r}_{k-1}\right); \tag{8}$$

$$\hat{\vartheta}_k = \frac{1}{2Th}\left(z_k - h\hat{r}_{k-1}\right), \tag{9}$$

что соответствует значению $\alpha_1 = h^2$.

Таким образом, алгоритм сопровождения МНК при условиях сглаживания координат и скорости задается уравнениями (5), (6) и в дальнейшем обозначается «МНК-1». При сглаживании только координат алгоритм задается уравнениями (8), (9) и обозначается через «МНК-2».

2. Сравнение точности оценок алгоритмами ФК и МНК

Оценка точности алгоритмов МНК по величинам дисперсий ошибок оценивания координат и скорости может быть получена методом статистических испытаний для конкретных моделей движений объектов. При этом представляет интерес сравнение данных результатов с результатами алгоритма ФК, поскольку он определяет потенциальные возможности по точности оценок. Кроме того, необходимы сравнения точности алгоритмов МНК-1 и МНК-2 между собой, что позволяет оценить степень влияния сглаживания скорости на величину дисперсии ошибки.

В качестве примера использована модель движения, заданная следующими стохастическими конечно-разностными уравнениями:

$$r_{k+1} = ar_k + Tq_k; \tag{10}$$

$$q_{k+1} = bq_k + T\xi_k; \tag{11}$$

$$\vartheta_{k+1} = (r_{k+1} - r_k)/T = (a-1)/T + q_k . \qquad (12)$$

Уравнение (10) формирует траекторию изменения координаты r_k, (11) задает величины приращений координат, а (12) вытекает из первых двух и описывает скорость изменения координат. В этих уравнениях T – период измерений, a и b – постоянные параметры, выбор которых обеспечивает желаемую длительность корреляции τ_r траектории. Процесс ξ_k полагается стационарным дискретным белым гауссовым шумом с нулевым математическим ожиданием и постоянной дисперсией σ_ξ^2.

В рассматриваемом примере выбраны следующие значения параметров: $T = 1$ (с), $a = 0{,}95$, $b = 0{,}9$, $\sigma_\xi^2 = 0{,}23$ (град.^{2}/с^4). При этих величинах в установившемся режиме обеспечивается значение дисперсии изменения координат $\sigma_r^2 = 158$ (град.^{2}) и длительность корреляции $\tau_r = 20$ (с). Модель измерений выбрана в виде уравнения $z_k = r_k + v_k$ ($h = 1$) при различных дисперсиях ошибок измерений σ_v^2. Они определяются путем задания отношения сигнала к шумам $\rho = \sigma_r^2/\sigma_v^2$, которые при вычислениях выбраны $\rho = 2,\ 5,\ 10$.

Для заданной модели способами, описанными в литературе [4], найден алгоритм ФК, в том числе уравнение Риккати для матрицы ковариации ошибок оценивания всех координат модели движения. На основе модели получены графики изменения дисперсий ошибок оценок координат σ_{er}^2 и скорости $\sigma_{e\vartheta}^2$ (рис. 1).

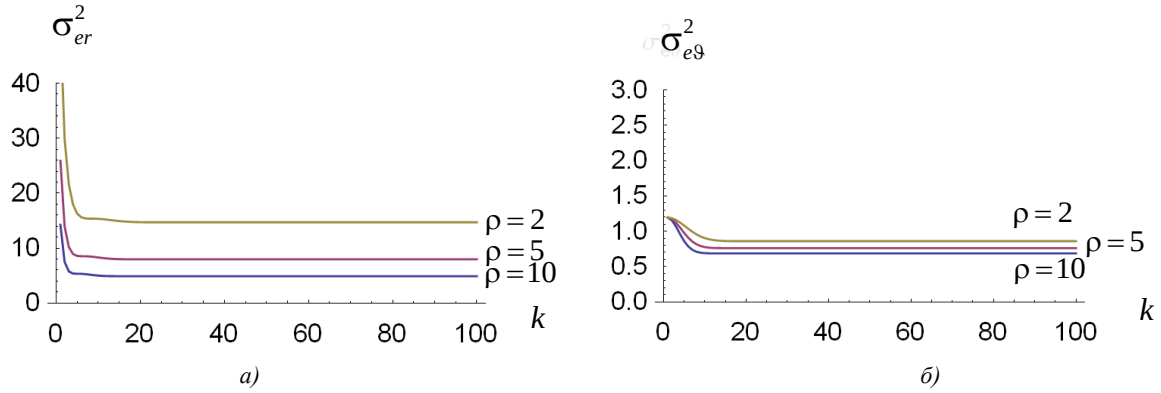


Рис. 1. Текущие значения дисперсий ошибок алгоритма ФК: *а)* координат; *б)* скорости

Для алгоритмов МНК использовались 10^3 реализаций траекторий выбранной модели движения. Находились ошибки оценок координат и скорости алгоритмами МНК-1 и МНК-2, а дисперсии ошибок оценок координат σ_{er}^2 и скорости $\sigma_{e\vartheta}^2$ определялись на основе усреднения квадратов ошибок по всему набору реализаций (рис. 2).

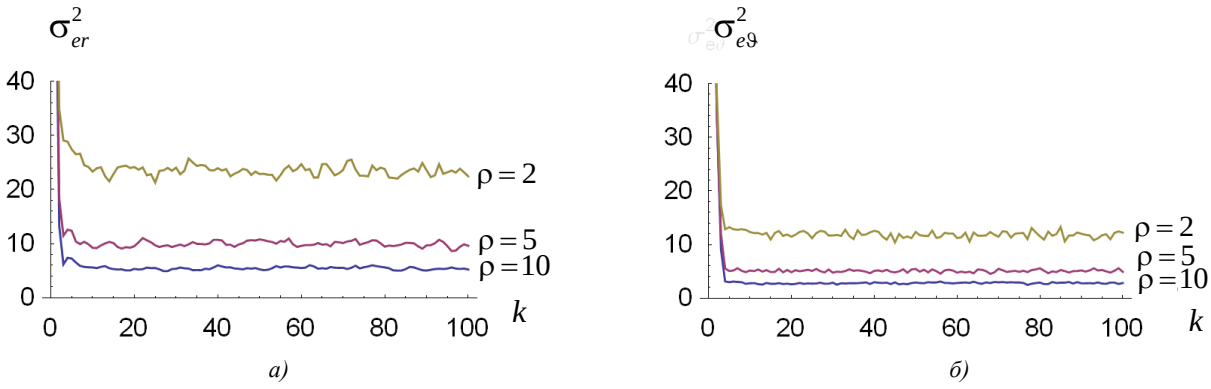


Рис. 2. Текущие значения дисперсий ошибок алгоритма МНК-1: *а)* координат; *б)* скорости

На рис. 3 показаны аналогичные результаты для алгоритма МНК-2 со сглаживанием только координат.

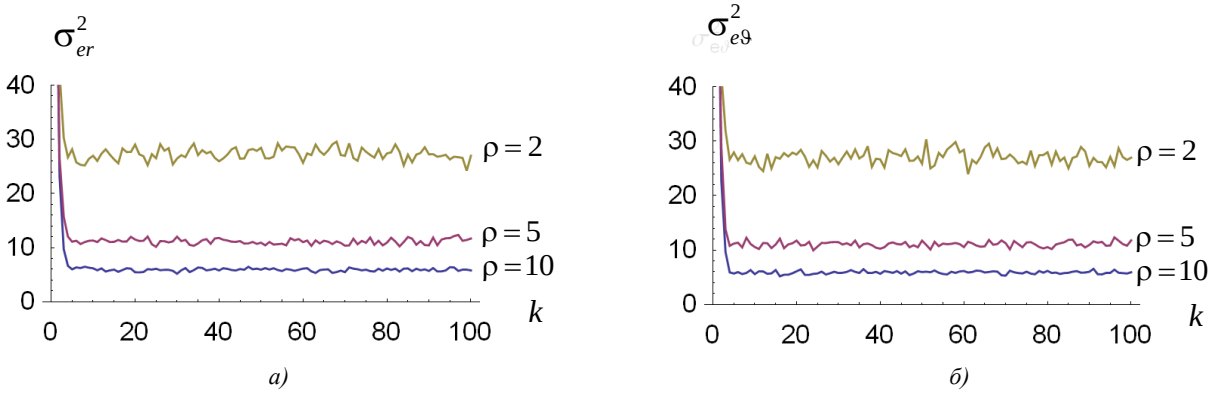


Рис. 3. Текущие значения дисперсий ошибок алгоритма МНК-2: *а)* координат; *б)* скорости

Сравнение этих результатов с дисперсиями ошибок ФК (см. рис. 1) позволяет оценить влияние неучета априорной информации о модели движения на точность оценки координат и скорости алгоритмами МНК. При больших отношениях сигнала к шумам ρ дисперсии ошибок оценок координат мало отличаются друг от друга. С уменьшением ρ дисперсии ошибок у алгоритмов МНК становятся больше, чем у ФК, особенно для алгоритма МНК-2 без сглаживания скорости.

Дисперсии ошибок оценок скорости алгоритмами МНК существенно выше, чем у ФК, даже при больших ρ. Это является следствием того, что выбор значений коэффициентов регуляризации α_1, α_2 в данной работе проведен только с учетом ошибок в оценке координат. Сглаживание скорости в алгоритме МНК-1 приводит к снижению ошибок в ее оценке при незначительных отличиях в величинах дисперсий ошибок оценок координат, что делает предпочтительным его использование по сравнению с алгоритмом МНК-2.

На рис. 4 изображены графики результатов сопровождения одной из реализаций модели траектории движения объекта, заданной уравнениями (10) и (11), для трех вариантов алгоритмов сопровождения. Здесь тонкой сплошной линией показана реализация траектории изменения координаты r_k, штриховой – ее измерения z_k при значении $\rho = 10$, толстой сплошной – результаты оценивания различными алгоритмами.

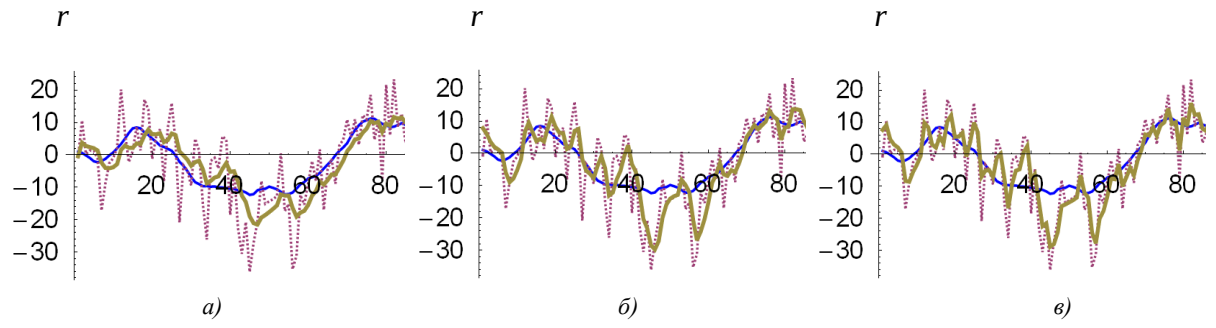


Рис. 4. Исходная траектория и результаты ее сопровождения алгоритмами на основе: *а)* фильтра Калмана; *б)* МНК-1 со сглаживанием координаты и скорости; *в)* МНК-2 со сглаживанием только координаты

Заключение

При отсутствии априорной информации о характеристиках траекторий движения воздушных объектов нахождение алгоритмов сопровождения можно осуществлять на основе МНК. Для него априорными являются эмпирические предположения о гладкости траектории

движения и скорости ее изменения. Особенностью решения задачи сопровождения является необходимость оценок не только координат положения объекта, но и скорости, что используется для экстраполяции координат на следующий период измерений. В рамках МНК задача решается на основе введения функционала качества сопровождения, в составе которого необходимо учитывать гладкость изменения координат и скорости. Такая задача относится к категории плохо определенных, и для ее решения необходимо использовать способы регуляризации путем введения в состав функционала слагаемых с весами, задаваемыми коэффициентами регуляризации.

В работе предложены вариант функционала качества и способ нахождения входящих в него коэффициентов регуляризации. Получены алгоритмы сопровождения МНК со сглаживанием координат и скорости, а также со сглаживанием только координат. На конкретном примере дана сравнительная оценка дисперсии ошибок сопровождения алгоритмами МНК и фильтром Калмана. Сделан вывод о том, что при использовании МНК целесообразно выбирать алгоритм сопровождения со сглаживанием координат и скорости МНК-1, поскольку он обеспечивает более высокую точность оценок скорости по сравнению с вариантом МНК-2 сглаживания только координат.

В целях снижения дисперсий ошибок в дальнейшем целесообразно рассмотреть другие виды функционалов качества и способы нахождения коэффициентов регуляризации.

Список литературы

1. Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление / Б.А. Алпатов [и др.]. – М. : Радиотехника, 2008. – 176 с.

2. Артемьев, В.М. Обнаружение объектов конечных размеров на изображениях в условиях неопределенности / В.М. Артемьев, А.О. Наумов, Л.Л. Кохан // Информатика. – 2010. – № 4. – С. 5–14.

3. Артемьев, В.М. Классификация и селекция изображений воздушных объектов в обзорных оптико-электронных системах / В.М. Артемьев, А.О. Наумов, Л.Л. Кохан // Информатика. – 2012. – № 1. – С. 18–26.

4. Chui, C.K. Kalman filtering with real-time applications / C.K. Chui, G. Chen. – Berlin : Springer-Verlag, 1991. – 250 p.

5. Репин, В.Г. Статистический синтез при априорной неопределенности и адаптация информационных систем / В.Г. Репин, Г.П. Тартаковский . – М. : Сов. радио, 1977. – 432 с.

6. Эльясберг, Л.Е. Определение движения по результатам измерений / Л.Е. Эльясберг. – М. : Наука, 1976. – 415 с.

7. Линник, Ю.В. Метод наименьших квадратов и основы математико-статистической теории обработки наблюдений / Ю.В. Линник. – М. : Физматгиз, 1962. – 432 с.

8. Сизиков, В.С. Математические методы обработки результатов измерений / В.С. Сизиков. – СПб. : Политехника, 2001. – 299 с.

9. Артемьев, В.М. Справочное пособие по методам исследования радиоэлектронных следящих систем / В.М. Артемьев. – Минск : Высш. школа, 1984. – 168 с.

Поступила 11.02.13

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

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

ALGORITHM FOR OBJECTS TRACKING IN OPTRONIC SYSTEMS BASED ON THE LEAST-SQUARES METHOD

An algorithm of objects tracking in an optronic systems based on the least-squares method is suggested, which assumes smoothness of the movement trajectory and its speed variation. The peculiarity of the problem is not only the estimation of the trajectory coordinates, but also its speed, which has not been considered in the least-squares method before. A comparison of the tracking error variance for the Kalman filter and the suggested algorithm is given for an example.

ИНФОРМАТИКА

2013 **июль-сентябрь** **№ 3**

УДК 004

Б.А. Залесский

КОМБИНАТОРНЫЙ АЛГОРИТМ ВЫДЕЛЕНИЯ КОНТУРОВ ОБЪЕКТОВ НА ЦИФРОВЫХ ИЗОБРАЖЕНИЯХ

Рассматривается алгоритм выделения контуров на изображениях, который основывается на использовании комбинаторных методов, применяемых для кластеризации ориентированного градиента. Алгоритм позволяет оценить с достаточно высокой точностью положение и углы наклона контуров, а также получить удобное для анализа формы объектов векторное представление контуров ломаными или гладкими кривыми.

Введение

Задача выделения контуров хорошо известна в обработке изображений. Контуры изображения достаточно широко используются для интерактивного и автоматического выделения объектов, улучшения качества изображений, решения различных задач распознавания, регистрации изменений, компьютерного зрения и т. д.

В данной работе контур определяется как связное в заданной системе окрестностей множество пикселов изображения, в которых перепад яркости больше установленного порога [1]. Граница связной области понимается как замкнутый контур.

В настоящее время существует множество различных подходов к выделению контуров. Наиболее ранние – оконные, появившиеся в 1980-е гг., основанные на использовании дифференциальных операторов, активно применяются до сих пор. Обычно авторы делят их на две группы в зависимости от типа используемого дифференциального оператора. К первой группе относятся алгоритмы, основанные на поиске больших значений откликов оконных операторов, вычисляющих первые разностные производные. Среди них алгоритмы Робертса, Собеля, Превитт, Кирша, Кани и др. [1, 2]. Алгоритмы второй группы для нахождения контуров используют различные разностные аналоги вторых производных в виде различных версий оператора Лапласа (включая LOG, DOG и т. д.) [1, 2]. Более поздние алгоритмы выделяют контуры с помощью преобразования Хафа, методов математической морфологии, теории графов, активных контуров, вариационных методов [1, 3]. Достаточно подробное описание известных алгоритмов и обширную библиографию по данной тематике можно найти в [4].

Большинство известных алгоритмов находят контуры в растровом виде, удобном для решения широкого круга задач обработки и распознавания изображений. Однако для некоторых задач, например связанных с анализом и распознаванием формы объектов или описанием чертежей и планов, проще использовать векторное представление контуров. Такое представление несложно получить в случае, когда на изображении присутствуют только контуры, состоящие из отрезков прямых или кусочно-гладких кривых, форма которых заранее известна. Если же форма контуров заранее неизвестна, построение их векторного представления превращается в трудоемкую задачу.

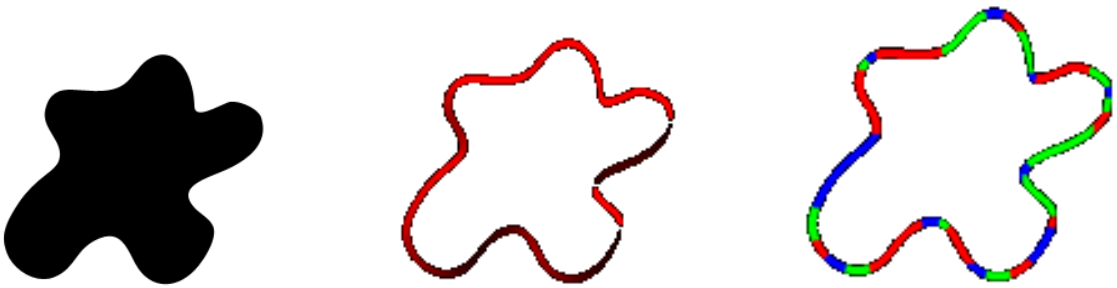
В качестве примера рассмотрим задачу выделения зданий, изображенных на аэрофотоснимке городского ландшафта (рис. 1, *а*), на основе анализа формы крыш. Для ее решения используются контуры изображения (рис. 1, *б*), на которых нужно найти участки, близкие по форме к отрезкам прямых линий, общие вершины таких участков и углы между линиями. В данном случае применение известных методов, большинство из которых использует растровое представление контуров, сопряжено с трудностями, вызванными целым рядом причин. Во-первых, на аэрофотоснимках в подавляющем числе случаев контуры не содержат идеально прямолинейных участков (тем более в присутствии растровых искажений), поэтому требуются операции, надежно выделяющие отрезки, близкие по форме к прямолинейным. Во-вторых, нередко в одном и том же месте сходятся несколько участков контуров, часть из которых порождена крышами, а другая – посторонними

объектами. В-третьих, часто края крыш порождают двойные или тройные контуры. Использование надежного векторного приближения контуров значительно облегчает решение задачи.

а) *б)*

Рис. 1. Изображение городского ландшафта и его контуры: *а)* часть оригинального аэрокосмического снимка городского ландшафта; *б)* бинаризованный градиент снимка

В предлагаемом алгоритме выделения контуров кластеризация ориентированного градиента *oG* используется для получения его достаточно точного векторного приближения на основе кластерного представления. Его особенность заключаются в том, что полутоновое изображение *oG*, яркости которого соответствуют углам наклона вектора первой производной к оси *OX*, кластеризуется с помощью предложенного ранее автором статьи комбинаторного алгоритма, специально модифицированного для решения сформулированной задачи [5–7]. Кластеры *oG* содержат пикселы, в которых значения углов наклона *oG* достаточно близки друг к другу, вследствие чего эти кластеры представляют собой сильно вытянутые множества, слабо искривляющиеся в пространстве (рис. 2). Их нетрудно аппроксимировать отрезками ломаных или кривых линий, которые можно использовать в качестве векторного представления контуров. При кажущейся простоте приведенного примера получить кластерное представление границы такой точности известными алгоритмами непросто.



а) *б)* *в)*

Рис. 2. Пример кластеризованного изображения *oG*: *а)* оригинальное модельное изображение; *б)* его ориентированный градиент. Яркостям красного цвета соответствуют углы наклона *oG* к оси *OX*, масштабированные к диапазону [0, 255]; *в)* кластеры *oG*, полученные после применения предложенного алгоритма, окрашенные для наглядности в три цвета

Алгоритм позволяет управлять кривизной получаемых кластеров. Полученные кластеры легко аппроксимируются отрезками ломаных или гладких кривых, с помощью которых и строится векторное представление контуров. Используя полученное векторное представление, можно выделять длинные участки контуров заданной гладкости и использовать полученные кривые для анализа формы объектов, их обнаружения и распознавания.

1. Описание алгоритма кластеризации контура

Предложенный алгоритм предназначен для построения векторного приближения контуров изображения путем кластерного представления *oG* с помощью разработанных ранее комбинаторных алгоритмов кластеризации [5–7], приближения кластеров отрезками прямых или гладких кривых, объединения отрезков в кусочно-гладкие кривые заданной кривизны.

Кластером в рассматриваемом случае называется максимальное по размеру связное множество пикселов изображения, обладающих общим для всех них свойством (все соседние пикселы кластера этим свойством не обладают).

Реализация алгоритма сводится к последовательному выполнению следующих тематических блоков:

1) вычисления *oG* изображения;

2) получения кластерного представления ориентированного градиента;

3) приближения кластеров *oG* отрезками прямых или гладких кривых;

4) объединения отрезков прямых или гладких кривых, приближающих кластеры, в кусочно-гладкие кривые заданной кривизны.

Дадим краткое описание блоков. Для этого обозначим через S множество пикселов $j = (x_1, y_2)$ полутонового или цветного $n_1 \times n_2$ изображения I с яркостями I_j, $0 \le I_j \le 255$, или значениями цвета $I_j = (R_j, G_j, B_j)$, $0 \le R_j, G_j, B_j \le 255$.

Опишем кратко *блок 1* для полутонового изображения $\boldsymbol{I}$, так как в случае цветного изображения производятся аналогичные действия для каждого цветового канала либо оно сначала преобразуется в полутоновое. Для окна наперед заданного вида вычисляется оконный градиент $\boldsymbol{G}$ изображения $\boldsymbol{I}$, представляющий собой $n_1 \times n_2$-матрицу с элементами – векторами $G_j = (g_{\text{гор},j}, g_{\text{верт},j})$. Для тестов и вычислительных экспериментов использовались окна, состоящие из двух прямоугольников, симметрично расположенных относительно пикселов $\boldsymbol{j}$, в которых определялся градиент. Числовые значения коэффициентов прямоугольников окна полагались +1 для одной половины окна и –1 для другой. Размеры одной половины окна выбирались от 1 x 1 до 11 x 11 пикселов (хотя теоретически здесь нет каких-либо ограничений).

Ориентированный градиент представляет собой $n_1 \times n_2$-матрицу $\boldsymbol{oG}$, элементы которой равны углам наклона вектора градиента G_j (измеренного с точностью 1°) к оси OX в случае, если длина вектора градиента $|G_j|$ больше наперед заданного порога $\tau > 0$, и равны какому-либо числу, например –1, в случае отсутствия градиента в пикселе $\boldsymbol{j}$ при $|G_j| \le \tau$. Для ускорения вычисления $\boldsymbol{oG}$ может быть применен метод бегущей суммы [8] или интегральное изображение [9].

Описание *блока 2* наиболее сложно с теоретической точки зрения. Детальное описание комбинаторного алгоритма кластеризации полутоновых изображений приведено в [5–7]. Суть алгоритма формулируется следующим образом: для целого $0 < k \le 255$ фиксируется произвольный набор целых чисел $0 = m_0 < m_1 < \ldots < m_k = 255$, в котором значения m_i выбираются исходя из поставленной задачи. Например, при сегментации они могут быть взяты на равном удалении от соседних значений, тогда $m_i = \left[\frac{255}{k}\right] i$ при i от 0 до $k-1$, или в соответствии с яркостной гистограммой $\boldsymbol{I}$.

Рассматривается множество $\mathcal{U}$ изображений $\boldsymbol{U}$ размера $n_1 \times n_2$, яркости которых U_j могут быть равными только m_i, $1 \le i \le k$. Тогда для произвольного полутонового изображения $\boldsymbol{I}$ его кластерное приближение $\boldsymbol{U^* = U^*(I)}$ строится как решение оптимизационной задачи

$$\boldsymbol{U^*} = \boldsymbol{U^*}(\boldsymbol{I}) = \arg\min_{U \in \mathcal{U}} \Big(\sum_{j \in S} \lambda_j \left| I_j - U_j \right| + \sum_{\substack{j_1, j_2 \in S \\ j_1 \sim j_2}} \beta_{j_1, j_2} \left| U_{j_1} - U_{j_2} \right| \Big), \qquad (1)$$

где $\lambda_j, \beta_{j_1, j_2} \ge 0$, а выражение $\boldsymbol{j}_1 \sim \boldsymbol{j}_2$ обозначает соседство пикселов в наперед заданной системе окрестностей. Чаще всего в обработке изображений используются классические 4- или 8-точечные системы окрестностей, хотя теоретически они могут быть любыми, нужно только помнить, что от вида окрестности зависит время поиска решения $\boldsymbol{U^*}$. Второе слагаемое в (1) играет роль

сглаживающей штрафной функции: меняя значения β_{j_1,j_2}, можно управлять размером и гладкостью границы кластеров на ***U****.

В случае кластеризации самого изображения ***I*** выбором значения β_{j_1,j_2} в зависимости от величины вектора градиента $|G_j|$ можно добиться того, чтобы полученные кластеры не пересекались с контурами. В [5–7] получены комбинаторные решения задачи (1), позволяющие находить кластерное представление ***U**** даже для больших изображений ***I*** путем сведения задачи к решению ***k*** независимых задач поиска минимального разреза ***k*** сетей (под сетью понимается ориентированный граф с источником и стоком [10]), которые строятся по ***I***. Там же подробно описано, как строятся сети.

Новизна настоящей работы заключается в том, что впервые предложенный метод кластеризации применяется для сегментации и сглаживания не самих изображений, а их контуров. Формально это означает, что в качестве исходного изображения рассматривается не само ***I***, а матрица его ориентированного градиента ***oG***. Кластерное представление ***V** = V* (oG)*** ориентированного градиента принимает значения m_i из множества целых чисел $0 = m_0 < m_1 < \ldots m_k = 359$ (если шаг дискретизации угла наклона градиента выбран равным 1º) и вычисляется как решение оптимизационной задачи

$$V^* = V^*(oG) = \arg\min_V \Big(\sum_{j \in S, oG_j \geq 0} \lambda_j \left| oG_j - V_j \right| + \sum_{\substack{j_1, j_2 \in S,\, oG \geq 0 \\ j_1 \sim j_2}} \beta_{j_1,j_2} \left| V_{j_1} - V_{j_2} \right| \Big), \qquad (2)$$

в которой, как и ранее, $\lambda_j, \beta_{j_1,j_2} \geq 0$, выражение $\boldsymbol{j}_1 \sim \boldsymbol{j}_2$ обозначает соседство пикселов в наперед заданной системе окрестностей, но суммирование берется только по пикселам, в которых присутствует градиент, т. е. в которых $oG \geq 0$. Решение задачи (2) с помощью предложенных методов комбинаторной оптимизации [5–7] строится аналогично решению задачи (1).

Следует заметить, что если для фиксированных λ_j увеличивать β_{j_1,j_2}, размеры кластеров возрастают, а их количество уменьшается. В пределе может получиться так, что каждый контур превратится в отдельный кластер. Однако это происходит при очень больших значениях β_{j_1,j_2}. Практические вычисления могут производиться в широком диапазоне значений параметров $\lambda_j, \beta_{j_1,j_2}$. Для упрощения процедуры выбора параметров можно положить все λ_j равными одному и тому же положительному числу, например $\lambda_j = 1$, тогда при четырехточечной системе окрестностей качественные с практической точки зрения результаты кластеризации могут быть получены при $0,25 \leq \beta_{j_1,j_2} \leq 20$, а при восьмиточечной – при $0,125 \leq \beta_{j_1,j_2} \leq 10$. Кривизной получаемых кластеров можно также управлять следующим образом: положить для некоторого числа $\rho \geq 1$

$$\beta_{j_1,j_2} = \begin{cases} \beta, \text{ если } \boldsymbol{j}_1 \sim \boldsymbol{j}_2, oG_{j_1} \geq 0, oG_{j_2} \geq 0 \text{ и } \left| oG_{j_1} - oG_{j_2} \right| \leq \rho; \\ 0 \text{ в противном случае.} \end{cases}$$

Это приведет к тому, что в кластер попадут только те соседние пикселы $\boldsymbol{j}_1$ и $\boldsymbol{j}_2$, в которых присутствуют градиенты G_{j_1} и G_{j_2}, образующие угол, не превосходящий ρ. Простейший пример кластеризации контуров приведен на рис. 2. Если использовать описанные выше стратегии выбора параметров, полученные кластеры *oG* будут содержать пикселы, в которых значения углов наклона *oG* близки друг к другу, вследствие чего большинство кластеров будут представлять собой вытянутые множества, слабо искривляющиеся в пространстве. Исключение могут составить лишь кластеры, возникшие в местах сильного перегиба контуров. Такие кластеры имеют небольшой размер (см. рис. 2), но опять-таки незначительную кривизну.

Описание *блока 3*. Свойства полученных кластеров позволяют использовать самые разнообразные методы для приближения кластеров отрезками прямых или гладких кривых, начиная от простого вычисления диаметра кластера или осей инерции или применения метода наименьших квадратов до использования аппроксимирующих сплайнов и других современных методов. Нужно только не забывать о том, что у коротких кластеров, образовавшихся в местах

перегиба, диаметр или большая ось инерции могут оказаться перпендикулярными к направлению контура в данном месте.

Результаты тестирования и практического применения предложенного алгоритма показали, что следует избегать соблазна использовать для приближения кластера ***oG*** отрезком прямой оценку его приближенного угла наклона m_i, так как в некоторых случаях реальный угол наклона кластера может существенно отличаться от этого значения. Ошибка может быть вызвана округлением значений угла наклона градиента до некоторого m_i, а также растровыми искажениями и шумами, присутствующими на изображении, или неточностями, возникшими при вычислении оконного градиента.

При реализации алгоритма параметры векторного представления кривых, приближающих кластеры, можно хранить в структуре, описывающей их свойства: количество кластеров ориентированного градиента обычно ограничено несколькими тысячами.

Блок 4 алгоритма заключается в построении векторного представления контуров путем объединения кривых, приближающих кластеры в более сложные и длинные линии. Здесь можно действовать несколькими способами. По-видимому, самый простой из них состоит в следующем: определить критерий соседства кластеров, например считать соседними только кластеры, имеющие общую границу, либо только кластеры, расположенные на расстоянии меньшем некоторого фиксированного числа. Последний критерий имеет смысл потому, что встречаются случаи, когда между кластерами контура расположена маленькая область, не принадлежащая исходному контуру (бинаризованному градиенту), либо маленький кластер в местах сильного перегиба контура, состоящий их одного или нескольких пикселов.

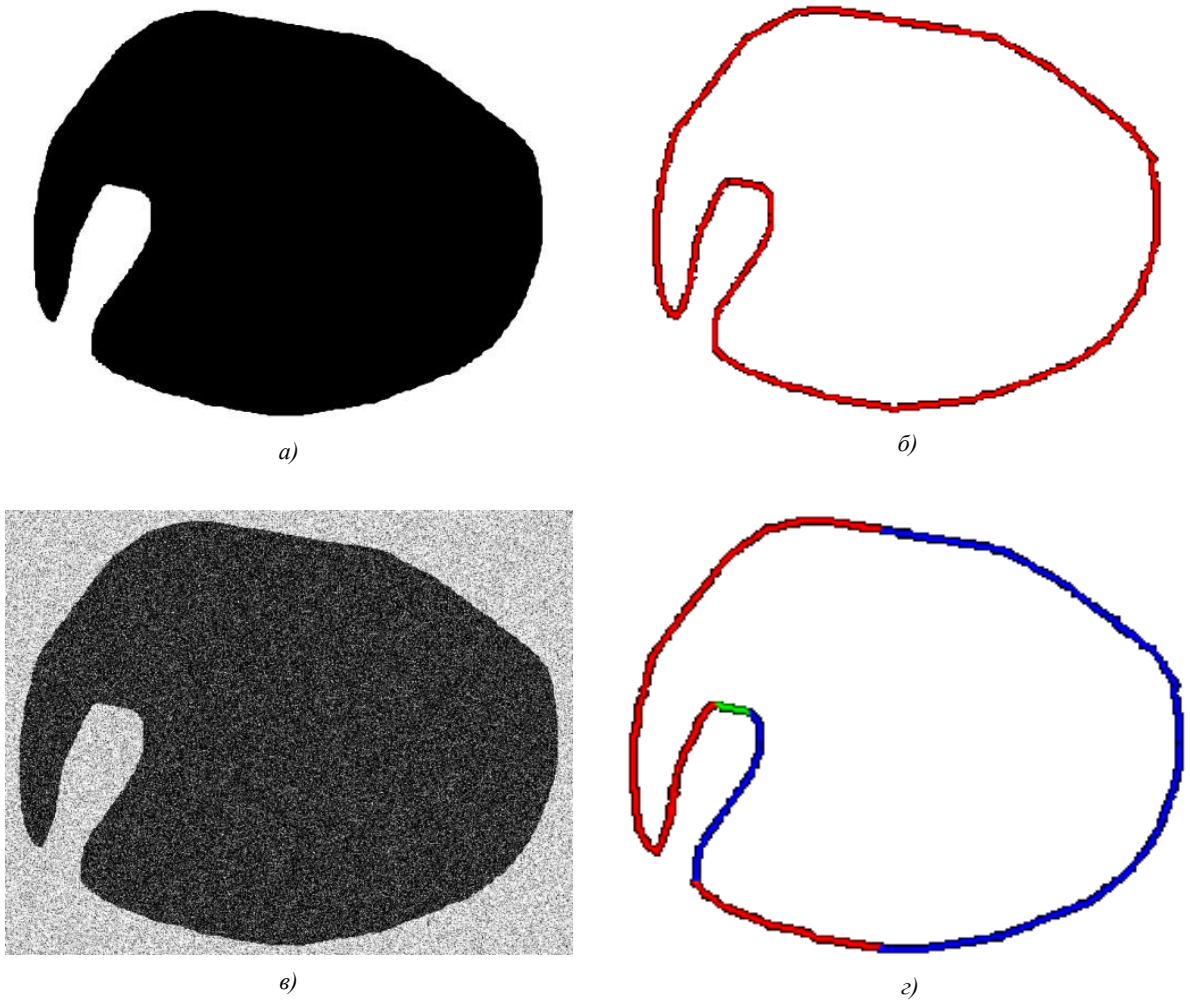
Далее следует построить граф, вершины которого соответствуют кластерам представления ***V****, а ребра – соседним кластерам. На основе полученного графа необходимо построить подграф, которой и будет задавать векторное представление контуров, оставляя в нем вершины и ребра, подходящие для решения поставленной задачи. Например, при решении некоторых задач может потребоваться учитывать только кластеры (и, следовательно, оставлять только вершины графа, соответствующие этим кластерам), площадь, размеры или форма приближающих кривых которых удовлетворяют определенным условиям. В качестве критерия оставления ребер может быть использован, например, допустимый угол между кривыми, приближающими соседние кластеры. Так, при поиске гладких участков контуров можно потребовать, чтобы углы между соседними приближающими кластеры кривыми были меньше некоторой заданной величины. При выделении же элементов крыш зданий можно потребовать, чтобы угол между приближающими кривыми менялся в заданном диапазоне, например был близок к 90 или 45 º. Примеры применения предложенного алгоритма приведены в следующем разделе.

2. Исследование свойств алгоритма

Исследование свойств алгоритма проводилось на рисунках геометрических фигур, созданных в графических редакторах, на снимках автомобилей и других транспортных средств. Снимки транспортных средств были выбраны для тестов потому, что большая часть их контуров имеет гладкую правильную форму, разную у разных моделей. В качестве линий, приближающих кластеры контуров, выбирались отрезки прямых, направление которых совпадает с направлением большей оси эллипса инерции кластера. Отрезки объединялись в кусочно-ломаные линии, если угол между соседними отрезками был меньше заданного.

Были также реализованы несколько версий алгоритма наращивания контуров, основанных на поиске направлений наименьшего изменения направления градиента. Однако их применение дало существенно худшие результаты векторизации. Одна из причин низкого качества векторизации – искажения значений ориентированного градиента, присутствующие даже на изображениях высокого качества. Например, значения ориентированного градиента каждой стороны многоугольника могут отличаться по значениям на 20 º. Вторая причина худшего качества векторного представления границы, полученного алгоритмом наращивания областей, вызвана ошибками, появляющимися в местах пересечения нескольких контуров.

На рис. 3 приведены результаты векторизации (предложенным комбинаторным алгоритмом) границы объекта, расположенного в левом верхнем углу, и его искаженной гауссовским шумом копии. В одну линию объединялись соседние отрезки, угол между которыми не превосходил 45 º. Объединение отрезков начиналось с верхнего правого в обе стороны, поэтому граница оригинального объекта оказалась представлена одной линией, окрашенной в красный цвет. Граница зашумленного объекта распалась на несколько линий потому, что в кластерном представлении зашумленного объекта появились маленькие кластеры, которые были исключены из рассмотрения.



а) *б)* *в)* *г)*

Рис. 3. Векторизованный градиент объекта: *а)* объект, созданный в графическом редакторе; *б)* векторное представление границы объекта с помощью отрезков прямых, направление которых задается большей осью эллипса инерции кластеров; *в)* тот же объект, искаженный гауссовским шумом; *г)* векторное представление границы искаженного шумом объекта с помощью отрезков прямых, направление которых задается большей осью эллипса инерции кластеров

Результаты векторного представления контуров автомобиля показаны на рис. 4. Приведены оригинальные изображения автомобилей размера 1600x1200 пикселов и векторных представлений их контуров, имеющих длину более 135 пикселов и представляющих собой кусочно-ломаные кривые, которые составлены из отрезков, образующих между собой угол меньше 90 º.

Полученные векторные представления контуров обладают большей гладкостью по сравнению с контурами, полученными методами наращивания областей. Они могут быть использованы как для решения задачи обнаружения и распознавания объектов по форме их границ, так и для улучшения качества изображений. Собственно, алгоритм и разрабатывался для решения этих задач.

Рис. 4. Изображения автомобилей и векторные представления их контуров, полученные с помощью предложенного алгоритма

Заключение

В статье предложен новый подход к задаче выделения и векторизации контуров на изображениях. Его смысл заключается в использовании комбинаторных методов кластеризации, разработанных автором ранее [5–7], для кластеризации ориентированного градиента. Полученные кластеры ориентированного градиента приближаются отрезками гладких кривых, которые объединяются в более длинные кусочно-гладкие кривые в соответствии с выбранными критериями соседства отрезков и допустимыми углами между ними. Построенные таким способом векторные представления контуров обладают заданной гладкостью и достаточно высокой точностью. Они удобны для решения задач обнаружения и распознавания объектов по форме. В ближайшем будущем планируется разработать подходы к решению этой задачи на основе анализа векторного представления контуров.

Список литературы

1. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1070 с.

2. Acharya, T. Image processing. Principles and Applications / T. Acharya, K. Ray Ajoy. – John Wiley & Sons, 2005. – 449 p.

3. Chan, T.F. Image Processing and Analysis / T.F. Chan, J.H.G Shen. – Philadelphia : Society for Industrial and Applied Mathematics, 2005. – 423 p.

4. Park, J.M. Encyclopedia of Computer Science and Engineering / J.M. Park, M. Lu. – Wiley-interscience, 2008. – 2365 p.

5. Zalesky, B.A. Integer Programming Methods in Image Processing and Bayes Estimation / B.A. Zalesky // Soft Computing in Image Processing – Recent Advances Series: Studies in Fuzziness and Soft Computing. – 2007. – Vol. 210. – P. 417–446.

6. Zalesky, B.A. Gibbs Classifiers / B.A. Zalesky // Probability Theory and Mathematical Statistics. – 2004. – Vol. 70. – P. 36–46.

7. Залесский, Б.А. Фильтрация и кластеризация мультиспектральных изображений с помощью алгоритма максимального потока в сети на основе вычисления градиента / Б.А. Залесский, Д.В. Прадун // Информатика. – 2010. – № 27. – С. 73–80.

8. McDonnell, M. Box-filtering techniques / M. McDonnell // Computer Graphics and Image Processing. – 1981. – Vol. 17, no. 1. – P. 65–70.

9. Viola, P. Rapid Object Detection using a Boosted Cascade of Simple Features / P. Viola, M.J. Jones // Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001). – 2001. – Vol. 1. – P. 511–518.

10. Пападимитриу, Х. Комбинаторная оптимизация / Х. Пападимитриу, К. Стайглиц. – М. : Мир, 1985. – 510 с.

Поступила 15.01.2013

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

B.A. Zalesky

COMBINATORIAL ALGORITHM FOR OBJECT CONTOURS DETECTION OF DIGITAL IMAGES

An algorithm for the extraction of image contours is presented. The algorithm is based on combinatorial methods of clustering the oriented gradient of an image. It allows to estimate the position and the slope angles of the contours with a sufficient accuracy, and to obtain their vector representation by the broken lines or pieces of smooth curves, which is convenient for analyzing the shape of objects.

УДК 004.032.6; 004.272.3

Ал.А. Петровский, А.В. Станкевич, А.А. Петровский

КОНВЕЙЕРНАЯ АРХИТЕКТУРА ДЕКОДЕРА CABAC СТАНДАРТА H.264/AVC ДЛЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Описывается архитектура декодера CABAC для мобильных приложений с разрешением до 625SD с трехступенчатым конвейером, позволяющая обеспечить декодирование одного бина за такт. Декодер совместим с профилями high profile, high 10 profile, high 4:2:2 profile, поддерживает режим MBAFF и блоки 8 × 8, а также масштабируем как по разрешению, так и по поддерживаемым инструментам декодирования, описанным в стандарте H.264. Выполняется сравнение с известными реализациями прототипа декодера CABAC на FPGA фирмы Xilinx.

Введение

В настоящее время наблюдается значительный рост спроса на высококачественное видео для мобильных устройств. При передаче и хранении видеоданных одной из ключевых проблем является компрессия этих данных при сохранении требуемого качества изображения. Для цифрового телевидения высокого разрешения (HDTV), мобильных телефонов, смартфонов, Blu-ray-дисков, потокового видео в Интернете и других приложений цифрового видео ITU-T и ISO/IEC рекомендован стандарт H.264/AVC (Advanced Video Coding: MPEG-4 Part 10) [1–6]. Высокая степень сжатия видеоданных в данном стандарте обеспечивается за счет большой вычислительной сложности, что требует высокой производительности аппаратуры при декодировании в реальном масштабе времени. Например, для видео с разрешением HDTV требуются производительность процессора порядка 83 гигаинструкций в секунду (GIPS) и пропускная способность подсистемы памяти 70 ГБ/с [7]. Реализация видеодекодеров на универсальных микропроцессорах с такими требованиями производительности вызывает серьезные затруднения, особенно для мобильных приложений, из-за высокого энергопотребления. Более предпочтительной является разработка специализированных вычислительных устройств на базе параллельно-поточной архитектуры [8].

Среди средств стандарта H.264, обеспечивающих высокую степень сжатия видеопотока, большую роль играет энтропийное кодирование синтаксических единиц и особенно контекстно-зависимое адаптивное двоичное арифметическое кодирование CABAC (context-adaptive binary arithmetic coding). Данный вид кодирования обеспечивает более высокую степень сжатия, чем контекстно-зависимое адаптивное кодирование с переменной длиной кодового слова CAVLC (context-adaptive variable-length coding), и на 9–14 % снижает требования по скорости передачи данных [9].

Известен ряд реализаций декодера CABAC как специализированных вычислительных устройств [10–13]. Реализации [10–12] не поддерживают такие важные инструменты стандарта H.264 для высокого профиля (high profile), как кодирование MBAFF (macroblock adaptive frame field) и кодирование с использованием блоков 8 x 8. Реализация [13] ориентирована на приложения HDTV и содержит избыточные аппаратные средства для получения требуемой высокой производительности. Так, для повышения производительности используются два вычислительных ядра для процесса DecodeDecision и четыре ядра для процесса DecodeBypass с соответствующими схемами управления. Избыточные аппаратные средства приводят к повышению энергопотребления, что критично для мобильных приложений.

Целью работы являлась разработка архитектуры и реализация прототипа декодера CABAC стандарта H.264 для мобильных приложений на базе FPGA с разрешением до 625SD (720x576 отсчетов яркости), частотой смены кадров 30 кадров/с и поддержкой профилей high profile, high 10 profile, high 4:2:2 profile, включая кодирование MBAFF и использование блоков 8 x 8.

1. Процесс декодирования стандарта H.264

Вследствие неоднозначного перевода на русский язык в различных источниках терминов стандарта H.264 далее будем использовать англоязычные названия процессов декодирования и обозначения переменных в соответствии со стандартом. Русскоязычные термины будут поясняться в скобках англоязычными оригинальными терминами.

Восстановление отдельных кадров видеосигнала из закодированного видеопотока по стандарту H.264/AVC MPEG-4 Part 10 выполняется в соответствии с блок-схемой (рис. 1). Здесь входной видеопоток представляет собой бинарную последовательность (bitstream) закодированных кадров изображения, которые разбиты на секции (в частном случае размер секции равен кадру), состоящие из цепочки синтаксических единиц (СЕ) макроблоков размером 16x16 отсчетов яркости и соответствующих им отсчетов цветности. Входной видеопоток декодируется одним из декодеров: декодером экспоненциальных кодов Голомба, декодером CAVLC или декодером CABAC – в соответствии с типом СЕ. Полученный результат декодирования – это значения отдельных СЕ и блоков остаточных данных (residual). Блок остаточных данных представляет собой набор разностных отсчетов яркости и цветности между ссылочным и текущим кадрами, закодированных дискретным косинусным преобразованием. Таким образом, формирование разностного кадра получается путем применения обратного косинусного преобразования к блоку восстановленных остаточных данных после деквантования. Восстановленный кадр получается сложением результирующего разностного кадра с кадром-прогнозом, который в режиме intra строится из ранее декодированных отсчетов текущего кадра или в режиме inter по одному или двум ранее восстановленным ссылочным кадрам, полученным с помощью векторов из блока компенсации движения (motion compensation). Для устранения артефактов на границах макроблоков восстановленный кадр фильтруется (deblocking filtering).

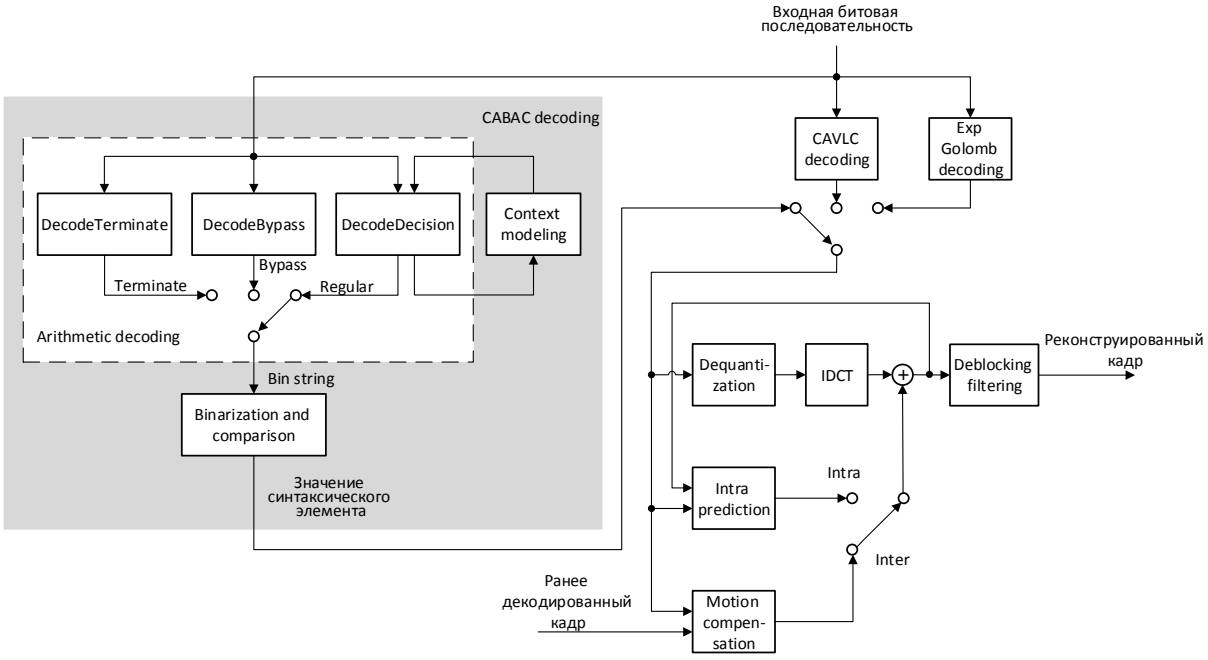


Рис. 1. Процесс декодирования стандарта H.264

В декодере CABAC входная кодированная битовая последовательность декодируется в последовательность бинов (bin string), которой впоследствии с помощью процесса бинаризации ставится в соответствие образец последовательности бинов для данной СЕ. При совпадении декодированной последовательности бинов с образцом бинаризации определяется искомое значение анализируемой СЕ.

Двоичное арифметическое декодирование использует только два символа. В CABAC они именуются как наиболее вероятный символ MPS и наименее вероятный символ LPS. Символы MPS и LPS могут быть как нулем, так и единицей.

Для каждого декодируемого бина осуществляется выбор контекстной модели, количественно характеризующейся двумя контекстными переменными *pStateIdx* и *valMPS*. Переменная *pStateIdx* является индексом вероятностного состояния и позволяет определить часть кодового интервала, соответствующего вероятности наименее вероятного символа LPS. Переменная *valMPS* хранит значение наиболее вероятного двоичного символа. Контекстная модель может быть выбрана из некоторого конечного множества доступных моделей в зависимости от ранее декодированных значений этой же СЕ для соседних макроблоков и ранее декодированных значений бинов текущей СЕ. Этот процесс называется контекстным моделированием. Таким образом, каждый бин декодируется согласно выбранной вероятностной модели. По результатам декодирования бина выбранная контекстная модель обновляется [14].

Арифметическое кодирование основывается на делении кодового интервала на подинтервалы в зависимости от вероятности появления символов. В стандарте H.264 данный процесс называется DecodeDecision. Исходный кодовый интервал *codIRange* делится на две части: *codIRangeLPS* и *codIRangeMPS* – в соответствии с выражениями

$$codIRangeLPS = codIRange \cdot PLPS, \quad codIRangeMPS = codIRange - codIRangeLPS,$$

где *PLPS* – вероятность LPS.

Далее проводится сравнение текущего значения переменной *codIOffset* со значением *codIRangeMPS* и в зависимости от результата сравнения формируются значение текущего бина и новая контекстная модель. Переменная *codIOffset* заполняется битами входной кодированной последовательности.

При декодировании некоторых синтаксических единиц или частей синтаксических единиц может использоваться процесс DecodeBypass, имеющий меньшую вычислительную сложность, чем рассмотренный процесс декодирования DecodeDecision. Характерной особенностью процесса DecodeBypass является отсутствие изменений значений переменной *codIRange*.

При декодировании синтаксической единицы *end_of_slice_flag*, а также при декодировании бина синтаксической единицы *mb_type*, указывающего режим I_PCM, должен реализовываться процесс DecodeTerminate стандарта H.264. В стандарте отмечается, что процесс DecodeTerminate может быть реализован как DecodeDecision при значениях переменных контекстной модели *pStateIdx* = 63 и *valMPS* = 0 (значение *pStateIdx* = 63 для других контекстных моделей встретиться не может). Однако в стандарте не отмечается то важное обстоятельство, что процесс ренормализации для DecodeTerminate по сравнению с DecodeDecision выполняется только для случая, когда $codIOffset \geq codIRange$.

2. Декодер CABAC

2.1. Организация вычислительного процесса декодера CABAC

Проведем анализ алгоритма декодирования стандарта H.264 (рис. 2).

Шаги алгоритма для декодирования одного бина нельзя выполнить за один такт, и их прямая последовательная реализация потребует не менее трех тактов. Рассмотрим возможности распараллеливания вычислительного процесса алгоритма и организации конвейерных вычислений.

Из приведенной на рис. 2 последовательности шагов алгоритма параллельно в одном такте могут выполняться шаги 4 и 5, поскольку они будут реализовываться разными блоками декодера.

Третий шаг не может быть совмещен в одном такте ни с одним из других шагов, поскольку вычисление значения контекстного индекса *ctxIdx* и получение текущей контекстной модели должны осуществляться перед декодированием очередного бина. Полученное значение бина используется для процесса бинаризации и вычисления контекстного индекса для следующего бина.

Для того чтобы арифметический декодер, реализующий вычисления на шаге 3, без простоев проводил вычисления в каждом такте, необходимо совмещение в одном такте шагов 1 и 2. Тогда можно будет построить трехступенчатый конвейер со следующим распределением шагов алгоритма: первая ступень – шаги 1 и 2, вторая ступень – шаг 3, третья ступень – шаги 4 и 5. При этом следует учитывать то обстоятельство, что в процессе декодирования возможно возникновение ситуа-

ции, когда следующий декодируемый бин использует то же значение *ctxIdx*, что и предыдущий. Выполнение шага 3 на такт позже, чем шагов 1 и 2, приведет в этом случае к получению неправильной контекстной модели для шага 3, поскольку она будет корректно обновлена только в следующем такте. Модификация алгоритма декодирования одного бина, позволяющая совместить в одном такте шаги 1 и 2 и выполнить их параллельно шагу 3, показана на рис. 3.

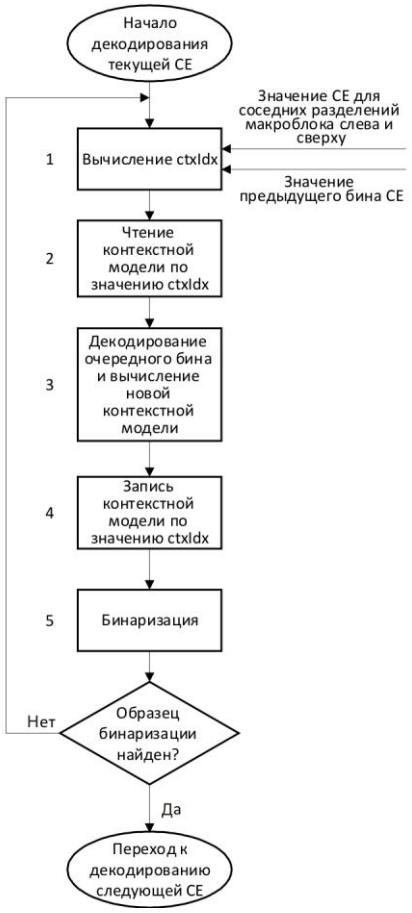


Рис. 2. Алгоритм декодирования одной СЕ

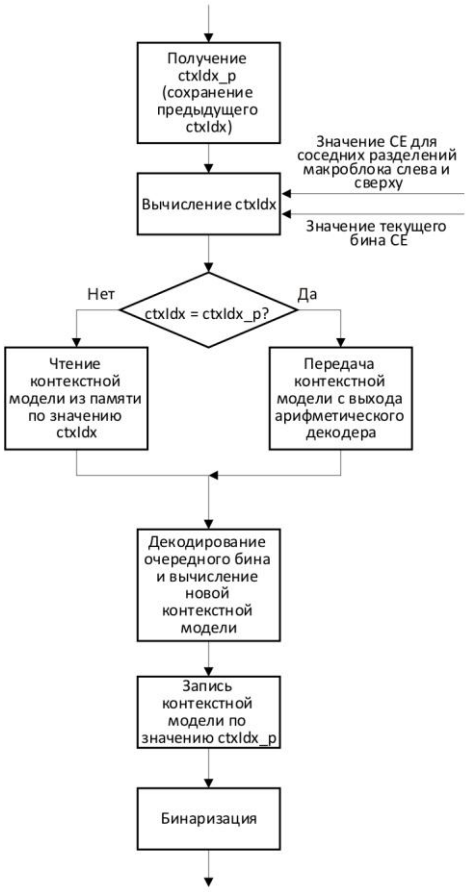


Рис. 3. Модифицированный алгоритм декодирования

Предлагаемая модификация алгоритма предполагает задержку на такт значения *ctxIdx* (на рис. 3 задержанное значение обозначено как *ctxIdx_p*). Если значения *ctxIdx* и *ctxIdx_p* совпадают, то в качестве контекстной модели для декодирования очередного бина должна быть выбрана модель с выхода арифметического декодера, а не прочитанная из памяти контекстных моделей. Значение *ctxIdx_p* также понадобится при обновлении контекстной модели. Поскольку арифметическое декодирование будет происходить на такт позже, чем расчет *ctxIdx*, при вычислении *ctxIdx* необходимо использовать значение текущего декодированного бина данной СЕ.

Реализация вычислительного процесса декодирования с учетом сделанных замечаний представлена в табл. 1.

2.2. Архитектура декодера CABAC

Для реализации поточных вычислений одного бина за такт работы декодера (рис. 4) был организован трехступенчатый конвейер в соответствии с табл. 1. Такт работы конвейера равен такту сигнала частоты синхронизации. В табл. 2 приведены блоки декодера, входящие в соответствующие ступени конвейера, и указаны операции, выполняемые ими. Следует иметь в виду, что вычислительные операции выполняются в течение такта, а фиксация результатов происходит по нарастающему фронту сигнала синхронизации. Все блоки декодера работают параллельно, однако обрабатывают информацию для различных бинов в соответствии с номером ступени конвейера. После завершения процесса декодирования текущего бина контекстная модель для следую-

щего бина фиксируется во входных регистрах второй ступени конвейера, новая контекстная модель сохраняется в памяти контекстных моделей по значению индекса *ctxIdx_p*, а декодированный бин заносится во входной сдвиговый регистр процессора образца бинаризации.

Таблица 1

Поточный вычислительный процесс декодирования

Ступень конвейера	Операции процесса декодирования	
1	Вычисление *ctxIdx* с учетом декодированного значения бина, полученного во второй ступени	
	Передача на вход арифметического декодера контекстной модели с выхода арифметического декодера в случае *ctxIdx* = *ctxIdx_p*, в противном случае – передача на вход арифметического декодера контекстной модели, прочитанной из памяти контекстных моделей по *ctxIdx*	
2	Декодирование очередного бина. Вычисление и сохранение новой контекстной модели по значению *ctxIdx_p*	Получение *ctxIdx_p* (сохранение предыдущего *ctxIdx*)
3	Бинаризация	

В начале каждой секции осуществляется инициализация памяти контекстных моделей. Для этого из памяти контекстных таблиц считываются пары значений переменных стандарта H.264 *m* и *n* и с помощью процессоров переменных контекстной модели рассчитываются значения *pStateIdx* и *valMPS*. Для ускорения процесса инициализации память контекстных таблиц располагает широкой 256-разрядной выходной шиной данных и в декодере имеется 16 идентичных параллельно работающих процессоров переменных контекстной модели. Начальные контекстные модели обновляются в процессе работы декодера.

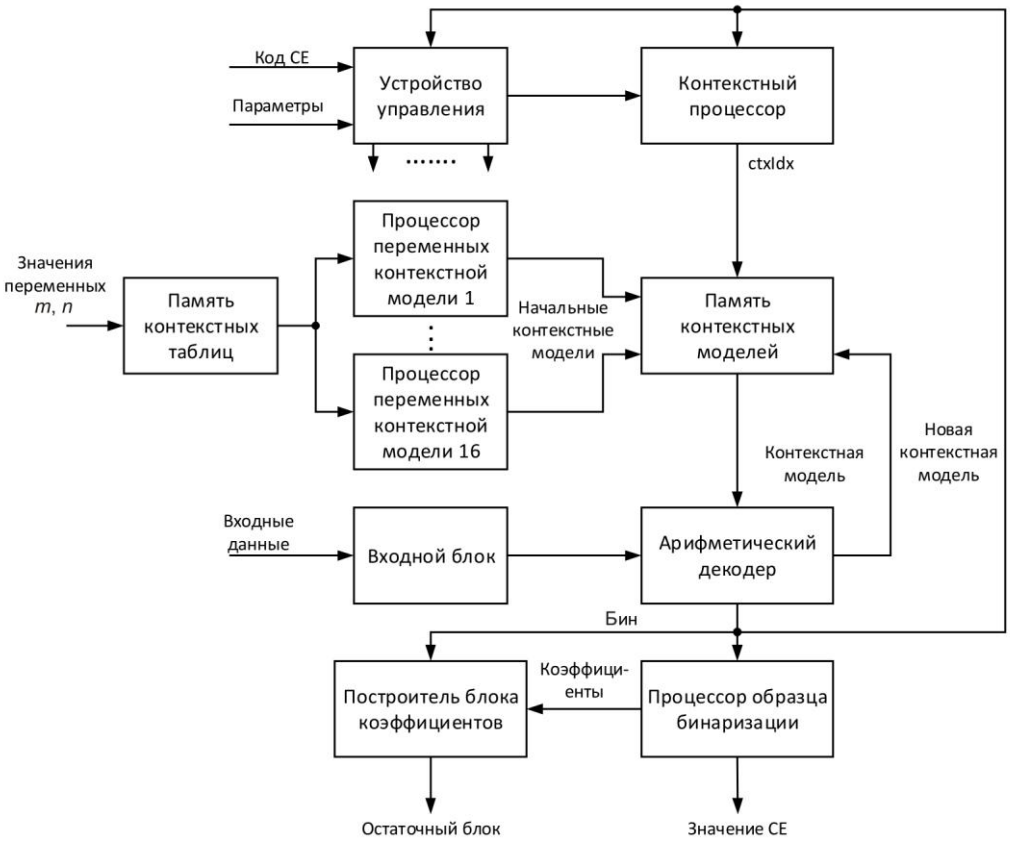


Рис. 4. Структура декодера CABAC

Таблица 2

Поточный вычислительный процесс декодера CABAC

Ступень конвейера	Блок декодера	Такт работы конвейера		
		$k-1$	k	$k+1$
1	Контекстный процессор. Память контекстных моделей	Вычисление $ctxIdx_i$ с учетом bin_{i-1}, выбор контекстной модели для $ctxIdx_i$	Вычисление $ctxIdx_{i+1}$ с учетом bin_i, выбор контекстной модели для $ctxIdx_{i+1}$	Вычисление $ctxIdx_{i+2}$ с учетом bin_{i+1}, выбор контекстной модели для $ctxIdx_{i+2}$
2	Арифметический декодер. Память контекстных моделей	Вычисление bin_{i-1}, вычисление и сохранение новой контекстной модели для $ctxIdx_{i-1}$	Вычисление bin_i, вычисление и сохранение новой контекстной модели для $ctxIdx_i$	Вычисление bin_{i+1}, вычисление и сохранение новой контекстной модели для $ctxIdx_{i+1}$
3	Процессор образца бинаризации	Вычисление образца бинаризации для bin_{i-2}	Вычисление образца бинаризации для bin_{i-1}	Вычисление образца бинаризации для bin_i

Для хранения полных контекстных таблиц стандарта H.264 необходима память общим объемом 8 Кбайт (8192 значения в диапазоне от –128 до +127). Полная память контекстных таблиц реализуется как массив из 1024 64-разрядных слов. Поскольку часть контекстных моделей может не использоваться для каких-то конкретных профилей стандарта H.264, объем памяти контекстных таблиц может быть сокращен.

Полная память контекстных моделей имеет объем 1024 7-разрядных слова. Каждое слово хранит пару переменных контекстной модели *pStateIdx* и *valMPS*.

Входные закодированные данные поступают в декодер по 32-разрядной шине данных (входная битовая последовательность «нарезана» 32-разрядными словами). Разрядность шины данных выбрана исходя из разрядности современных микропроцессоров. Входной блок обеспечивает формирование 7-разрядного окна, смещающегося над 32-разрядным словом входных данных. Смещение задается числом бит, потребленных арифметическим декодером при декодировании. Необходимость формирования окна связана с тем обстоятельством, что при декодировании очередного бина может быть потреблено заранее неизвестное число бит входной закодированной последовательности. Окно будет всегда выравниваться по биту входного слова, соответствующему первому биту для очередного декодируемого бина. По мере декодирования бинов окно продвигается по входному 32-разрядному слову, начиная с первого еще не использованного при декодировании бита входного слова. Размер окна определяется максимальным числом бит, которое может быть потреблено декодером при операции ренормализации.

По значению контекстного индекса *ctxIdx* из памяти контекстных моделей извлекается соответствующая контекстная модель, которая используется при декодировании текущего бина. Значение контекстного индекса *ctxIdx* формируется с помощью контекстного процессора в соответствии с кодом декодируемой СЕ, текущими значениями параметров (типов секции и соседних макроблоков, индексов разделений и т. п) и значениями этой же СЕ для соседних слева и сверху макроблоков или разделений макроблоков. Диапазон возможных значений *ctxIdx* для каждого типа СЕ ограничен стандартом, поэтому для исключения операции сложения при расчете *ctxIdx* проведены предварительные вычисления с целью замены сумматоров мультиплексорами. Это позволяет уменьшить критическую задержку за счет удаления схем межразрядного переноса.

Для декодирования ряда СЕ (*mb_skip_flag, mb_field_decoding_flag, coded_block_pattern, mb_type, ref_idx, mvd, intra_chroma_pred_mode, coded_block_flag, transform_size_8x8_flag*) требуется информация о соседних слева и сверху значениях этой же СЕ для соседних разделений макроблока или соседних макроблоков с адресами mbAddrA и mbAddrB [14]. Для этого контекстный процессор содержит регистровые файлы для хранения значений СЕ для верхней строки макроблоков секции или значений СЕ для нижних макроблоков пары верхней строки

пар макроблоков при наличии такого деления. Запись в указанные регистры происходит по завершении декодирования макроблока. Кроме того, имеются регистровые файлы для хранения значений СЕ для верхних макроблоков пары макроблоков, которые используются для случая полевых пар макроблоков для кадров MBAFF.

В соответствии с выбранной контекстной моделью в арифметическом декодере осуществляется декодирование очередного бина. Декодированный бин поступает в процессор образца бинаризации для сравнения с образцами последовательности бинов для данной СЕ. Декодированное значение СЕ определяется при совпадении декодированной последовательности бинов с образцом бинаризации.

Декодированное значение СЕ или остаточный блок сопровождается сигналом готовности. Одновременно с сигналом готовности формируется сигнал, стробирующий значение числа бит входной последовательности, использованных для декодированной текущей синтаксической единицы (сигнал числа потребленных бит и сигнал готовности на рис. 4 не показан).

Все СЕ, кодируемые CABAC, разделены на две группы: одиночные СЕ и СЕ блока остаточных данных. К одиночным СЕ относятся: *mb_skip_flag*, *mb_field_decoding_flag*, *end_of_slice_flag*, *mb_type*, *transform_size_8x8_flag*, *coded_block_pattern*, *mb_qp_delta*, *prev_intra4x4_pred_mode_flag*, *rem_intra4x4_pred_mode*, *prev_intra8x8_pred_mode_flag*, *rem_intra8x8_pred_mode*, *intra_chroma_pred_mode*, *ref_idx_l0*, *ref_idx_l1*, *mvd_l0*, *mvd_l1*, *sub_mb_type*. К СЕ блока остаточных данных относятся: *coded_block_flag, significant_coeff_flag, last_significant_coeff_flag, coeff_abs_level_minus1, coeff_sign_flag*.

Одиночные СЕ декодируются поодиночке. Все СЕ блока остаточных данных обрабатываются последовательно друг за другом с помощью построителя блока коэффициентов до тех пор, пока не будут декодированы все уровни коэффициентов дискретного косинусного преобразования соответствующего блока остаточных коэффициентов (до 64 коэффициентов). Обработка полного блока коэффициентов как единой СЕ позволяет сократить накладные расходы, связанные с простаиванием декодера CABAC в паузах между завершением обработки предыдущей СЕ и началом обработки следующей.

В процессоре образца бинаризации осуществляется преобразование последовательного кода (бинов от арифметического декодера) в параллельный код для подбора образца бинаризации. Подбор образца осуществляется в зависимости от кода типа бинаризации, поступающего от устройства управления.

При декодировании СЕ, и особенно блоков остаточных данных, возможна ситуация, когда 32 разрядов входного слова будет недостаточно для кодирования значения текущей СЕ. В этом случае декодер CABAC установит сигнал cabac_shift_en на один период сигнала синхронизации (рис. 5), а на выход числа потребленных битов будет установлено значение 32 (при декодировании потреблено 32 бита). Сигнал готовности cabac_ready установлен не будет. Работа декодера будет приостановлена. Ведущее устройство (декодер H.264) должно в течение такта установить на вход cabac_data_input новое 32-разрядное слово данных, после чего в следующем такте процесс декодирования продолжится. На рис. 5 приняты следующие обозначения сигналов: start_cabac – сигнал разрешения работы декодера; clk – сигнал синхронизации; gclk_cabac – клапонированный сигнал синхронизации; cabac_consumed_bits_len – число бит, потребленных при декодировании; cabac_decoding_output – выход процессора образца (не используется при декодировании остаточного блока); coefflevel_i – уровень *i*-го коэффициента дискретного косинусного преобразования. Декодирование осуществляется в порядке, обратном порядку построения карты значащих коэффициентов остаточного блока [14]. В данном примере понадобились дополнительные биты закодированной последовательности для декодирования coefflevel_0.

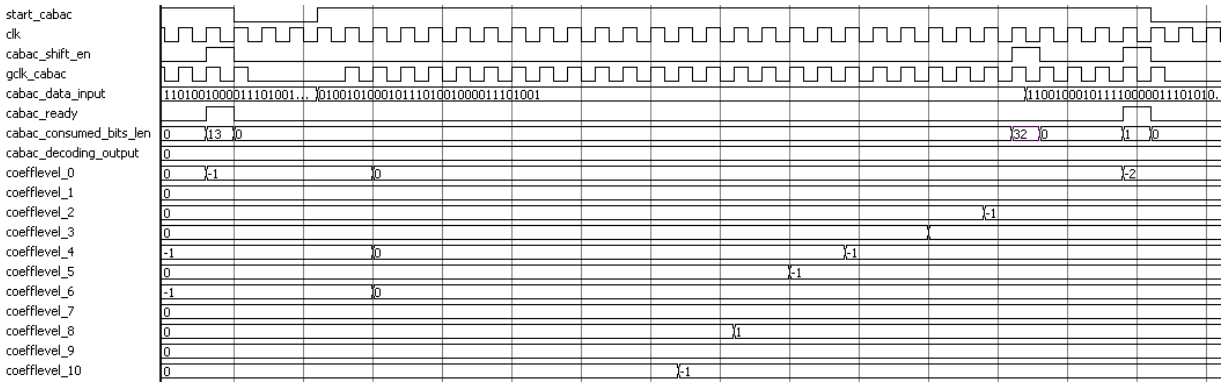
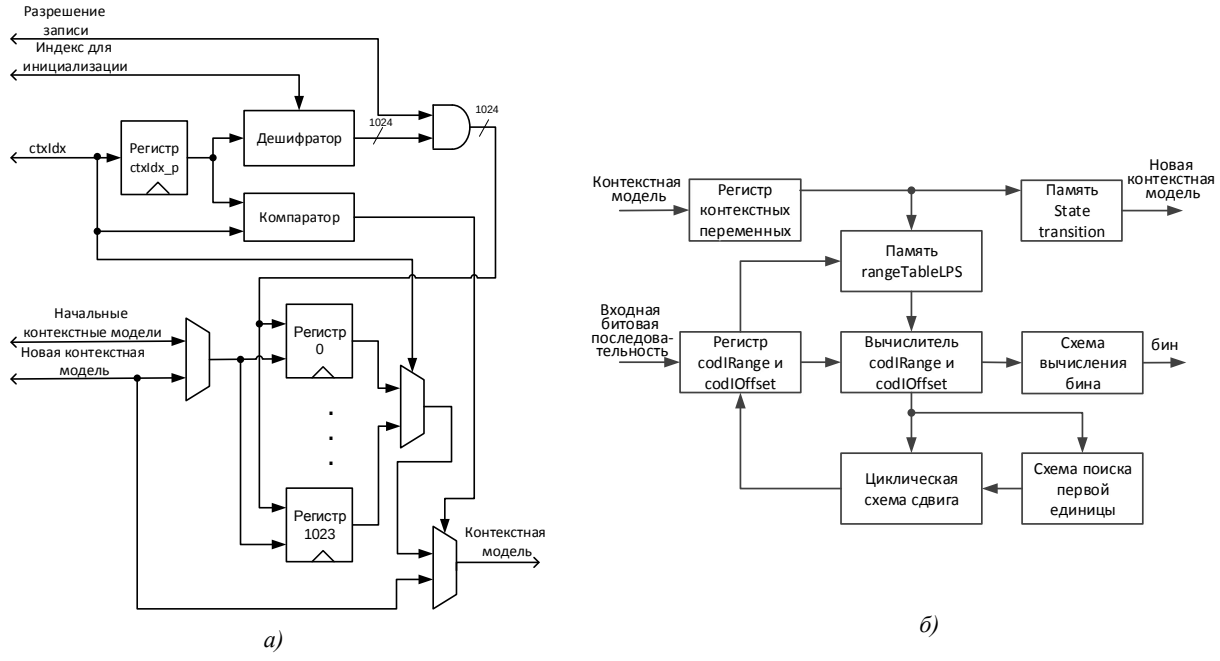


Рис. 5. Временная диаграмма декодирования блока остаточных данных с подкачкой очередного входного слова из внешнего буфера

2.3. Память контекстных моделей

Контекстные модели должны храниться в загружаемой памяти. Текущее значение индекса *ctxIdx* адресует конкретную контекстную модель. Память должна быть многопортовой, поскольку в нее необходимо одновременно записывать новое значение контекстной модели по индексу *ctxIdx_p* (синхронная запись) и читать контекстную модель по индексу *ctxIdx* (асинхронное чтение). Кроме того, должна быть предусмотрена запись начальных моделей в начале каждой секции через третий порт либо требуется использовать мультиплексор данных для входного порта. Из приведенных требований следует, что синхронную память использовать в данном случае нельзя, поскольку для такой памяти результат чтения появится на выходе с задержкой на такт относительно появления текущего индекса *ctxIdx*. Память с одним или двумя синхронными портами для записи и асинхронным портом для чтения достаточно сложно реализуется в заказной микросхеме. Поэтому для того чтобы чтение из памяти контекстных моделей происходило в том же такте, что и вычисление индекса *ctxIdx*, память была реализована на регистрах (рис. 6, *а*).

Количество 7-разрядных регистров на рис. 6, *а* приведено для случая хранения полного числа контекстных моделей стандарта H.264.



а) *б)*

Рис. 6. Структура блоков декодера: *а)* память контекстных моделей; *б)* арифметический декодер

В начале каждой секции по индексу для инициализации памяти и сигналу разрешения записи, поступающих от устройства управления, происходит занесение в память начальных контекстных моделей. В процессе декодирования чтение текущих контекстных моделей организуется путем мультиплексирования выходов регистров в зависимости от значения индекса *ctxIdx*. Компаратор на равенство обеспечивает управление выходным мультиплексором контекстных моделей для обработки ситуации, когда *ctxIdx = ctxIdx_p*.

2.4. Арифметический декодер CABAC

Арифметический декодер (рис. 6, *б*) позволяет декодировать один бин за такт работы декодера.

Разрядность всех операционных устройств – девять разрядов (минимально оговоренная стандартом H.264).

По сигналу инициализации происходит занесение начальных значений переменных *codIRange* и *codIOffset* в соответствующие регистры. В каждом такте работы арифметического декодера значения *codIRange* и *codIOffset* обновляются в соответствии с алгоритмом. Значения контекстных переменных *pStateIdx* и *valMPS* текущей вероятностной модели в каждом такте принимаются во входные регистры.

Таблицы арифметического декодера хранятся в ПЗУ. При реализации алгоритма DecodeDecision используются таблицы значений rangeTabLPS и таблицы для определения новых значений pStateIdx. Табличная реализация алгоритма арифметического декодирования предназначена для исключения операции умножения при подинтервальном разбиении исходного диапазона. Таблицы имеют следующий объем и разрядность: rangeTableLPS memory – 256 8-разрядных слов, state transition LPS memory и state transition MPS memory – 64 6-разрядных слова.

Из алгоритма арифметического декодирования следует, что необходимо осуществлять сравнение значений переменных *codIRange* и *codIOffset*. Аппаратные затраты на реализацию цифровых компараторов для проверки условия неравенства являются достаточно большими. С учетом необходимости реализации последующей операции вычитания *codIOffset* – *codIRange* такую проверку можно реализовать путем анализа знака операции вычитания. Выбор декодированных значений и значений переменных для различных процессов DecodeDecision, DecodeBypass и DecodeTerminate реализуется с помощью соответствующих мультиплексоров.

При арифметическом декодировании возможно возникновение необходимости проведения ренормализации. Для реализации процедуры ренормализации за один такт используются многоразрядные циклические сдвигающие устройства на базе мультиплексоров (Barrel Shifter). Число необходимых сдвигов формируется схемой обнаружения старшей двоичной единицы в текущем значении *codIRange*. Число разрядов, потребленных из входной закодированной последовательности, передается во входной блок (этот выход на структуре не показан).

В соответствии со стандартом H.264 процессы DecodeDecision, DecodeBypass и DecodeTerminate не могут выполняться параллельно; следовательно, для экономии аппаратных ресурсов их можно выполнить на одной аппаратуре.

Помимо декодированного значения бина арифметический декодер формирует новую контекстную модель для текущего контекстного индекса *ctxIdx*.

2.5. Реализация декодера CABAC на FPGA

Для проверки работоспособности предложенных архитектурных решений была выполнена реализация декодера CABAC на FPGA Xilinx.

Для сравнения характеристик арифметического декодера (см. разд 2.3) с известными реализациями [15, 16] было выбрано семейство Viretex 4. Аппаратные затраты и производительность предлагаемого арифметического декодера, а также сравнение их с известными реализациями на базе FPGA приведены в табл. 3.

Таблица 3

Сравнение известных реализаций арифметического декодера на FPGA

Характеристика декодера	Источник		Предлагаемый декодер
	[15]	[16]	
Тип FPGA	90 nm FPGA семейства Virtex 4	Altera Stratix II S60	Virtex 4 XC4VLX15
Затраты аппаратных ресурсов FPGA	302 slice	389 Adaptive Logic Module	273 slice
Максимальная тактовая частота, МГц	100	105	105
Количество тактов для декодирования одного бина	2	1	1

Семейство Altera Stratix II является приблизительным аналогом Xilinx Virtex 4, выпускаемым по такой же 90 nm технологии. При сопоставлении ресурсов следует иметь в виду, что один ALM (Adaptive Logic Module) Stratix II приблизительно соответствует одной секции Virtex 4, поскольку они содержат по два LUT, два триггера, а также некоторые другие аналогичные друг другу ресурсы для реализации арифметических операций.

В реализации [15] декодирование одного бина осуществляется за два такта. В течение первого такта из таблицы содержимое извлекается *Range LPS* и рассчитываются количественные характеристики контекстной модели. В течение второго такта вычисляется декодированное значение бина. Реализация [16] так же, как и предлагаемая, обеспечивает получение одного декодированного бина за такт частоты синхронизации.

Результаты реализации арифметического декодера на базе FPGA показывают, что при сопоставимой с реализацией [16] производительности предлагаемый декодер имеет меньшие затраты аппаратных ресурсов кристалла FPGA, а по сравнению с реализацией [15] имеется выигрыш в два раза по производительности при меньших аппаратных затратах.

Для реализации полного декодера CABAC были выбраны более производительные семейства Virtex 5 и Virtex 6. Характеристики реализации декодера CABAC после процедуры синтеза проекта с помощью ISE 12.2 приведены в табл. 4 и 5.

Блочная память применяется для реализации памяти контекстных таблиц, блоки DSP48E используются в качестве умножителей в процессорах переменных контекстной модели.

Из приведенных результатов следует, что основные аппаратные затраты приходятся на два блока: контекстный процессор и память контекстных моделей.

Основным блоком, вносящим комбинационную задержку, является контекстный процессор. Арифметический декодер по отчету процедуры синтеза может работать для XC6VLX195T на тактовой частоте 206 МГц.

Таблица 4

Аппаратные затраты ресурсов FPGA и производительность декодера CABAC

Характеристика	Virtex 5	Virtex 6
Тип FPGA	XC5VLX155T	XC6VLX195T
Количество триггеров	23446	23073
Количество LUT	15225	12473
Количество блоков памяти	4	4
Количество блоков DSP48E	16	16
Максимальная тактовая частота, МГц	75	94

Таблица 5

Распределение аппаратных затрат по блокам декодера CABAC

Блок декодера	Доля затрат для триггеров, %	Доля затрат для LUT, %
Контекстный процессор	61,0	39,3
Устройство управления	0,7	1,8
Входной блок	0,2	0,5
Арифметический декодер	0,1	1,2
Процессор образца бинаризации	0,2	1,8
Построитель блока коэффициентов	4,7	1,9
Память контекстных таблиц	0,1	0,2
Процессор контекстных переменных	2,0	3,1
Память контекстных моделей	31,0	50,2

Следует отметить, что поддержка возможности кодирования MBAFF значительно увеличивает затраты аппаратных ресурсов и снижает максимальную тактовую частоту декодера. Так, для кристалла XC6VLX195T исключение поддержки кодирования MBAFF снижает аппаратные затраты по триггерам на 31 %, просмотровым таблицам (LUT) на 30 % и повышает максимальную тактовую частоту на 4 % по сравнению с данными табл. 4.

Провести сравнение полученных характеристик с реализациями [10–13] затруднительно, поскольку данные реализации выполнены на ASIC.

Для тестирования декодера было разработано программное обеспечение на основе Reference Software JM [17]. Результаты тестирования считались правильными при полном совпадении результатов декодирования предложенным декодером и тестовым обеспечением на базе JM. Тестирование подтвердило работоспособность декодера с учетом режимов MBAFF и I_PCM.

Заключение

В работе предложена архитектура конвейерного декодера CABAC для мобильных приложений с разрешением до 625SD и поддержкой профилей high profile, high 10 profile, high 4:2:2 profile, включая кодирование MBAFF и использование блоков 8 x 8. Декодер обеспечивает декодирование одного бина за такт и имеет трехступенчатый конвейер.

Реализован прототип декодера на FPGA Xilinx семейств Virtex 5 и Virtex 6 с максимальными тактовыми частотами 75 и 94 МГц соответственно.

Анализ известных источников по реализациям CABAC показывает, что при декодировании одного бина за такт достаточно частоты синхронизации 100 – 160 МГц для обработки разрешений HD (105 МГц для 1920x1088@30 frames/s по сведениям [13], 160 МГц для 1920x1088@30 frames/s по сведениям [18]). Приведенные сведения позволяют сделать вывод о достаточной производительности разработанного декодера для разрешений 720x576 даже при его реализации на FPGA семейства Virtex 6.

Декодер является масштабируемым как по разрешению, так и по поддерживаемым инструментам кодирования стандарта H.264. Основное ограничение на дальнейшее расширение возможностей предложенного декодера CABAC связано с обрабатываемыми типами остаточных блоков. Так, декодер поддерживает следующие типы остаточных блоков: block of luma DC transform coefficient levels, block of luma AC transform coefficient levels, block of 16 luma transform coefficient levels, block of chroma DC transform coefficient levels when ChromaArrayType is equal to 1 or 2, block of chroma AC transform coefficient levels when ChromaArrayType is equal to 1 or 2, block of 64 luma transform coefficient levels.

Список литературы

1. ITU-T. ISO/IEC. ITU-T Rec. H.264 Advanced video coding for generic audiovisual services / ISO/IEC 14496-10 MPEG-4 AVC. ITU. Geneva [Electronic resource]. – Mode of access : http://www.itu.int/rec/T-REC-H.264. – Date of access : 31.05.2013.

2. Overview of the H.264/AVC video coding standard / T. Wiegand [et al.] // IEEE Transaction on Circuits and Systems for Video Technology. – 2003. – Vol.13, no. 7. – P. 560–576.

3. Marpe, D. The H.264/MPEG4 Advanced Video Coding standard and its applications / D. Marpe, T. Wiegand, G.J. Sullivan // IEEE Communications Magazine. – 2006. – Vol. 44, no. 8. – P. 134–143.

4. Kwon, S.-K. Overview of H.264/MPEG-4 part 10 / S.-K. Kwon, A. Tamhankar, K.R. Rao // Journal of Visual Communication and Image Representatin. – 2006. – Vol. 2, no. 17. – P. 186–216.

5. Schäfer, R. The emerging H.264/AVC standard / R. Schäfer, T. Wiegand, H. Schwarz // EBU Technical Review. – 2003, no. 293. – P. 1–12.

6. Richardson, I.E.G. H.264 and MPEG-4 Video Compression / I.E.G. Richardson. – Chichester, UK : Wiley, 2003. – 305 p.

7. Furht, B. Handbook of Mobile Broadcasting: DVB-H, DMB, ISDB-T, and mediaflo / B. Furht, S. Ahson. – NW, USA : CRC Press, 2008. – 726 p.

8. H.264/AVC Decoder Prototype Using a Platform Based SoC Design Methodology / Al. Petrovsky [et al.] // In Proc. of the 10th International Conference Pattern Recognition and Information Processing (PRIP)'2009. – Minsk, 2009. – P. 165–170.

9. Marpe, D. Context-Based Adaptive Binary Arithmetic Codingin the H.264/AVC Video Compression Standard / D. Marpe, H. Schwarz, T. Wiegand // IEEE Transactions on circuits and systems for video technology. – 2003. – Vol. 13, no. 7. – P. 620–636.

10. Chen, J.W. A hardware accelerator for context-based adaptive binary arithmetic decoding in H.264/AVC / J.W. Chen, C.R. Chang, Y.L. Lin // In Proc. IEEE ISCAS. – Kobe, Japan, 2005. – P. 4525–4528.

11. Yu, W. A high performance CABAC decoding architecture / W. Yu, Y. He // IEEE Transaction Consumer Electronics. – 2005. – Vol. 51, no 4. – P. 1352-1359.

12. Kim, C.H. High speed decoding of context-based adaptive binary arithmetic codes using most probable symbol prediction / C.H. Kim, I.C. Park // In Proc. IEEE ISCAS. – Island of Kos, Greece, 2006. – P. 1707–1710.

13. Yang, Y.-C. High-Throughput H.264/AVC High-Profile CABAC Decoder for HDTV Applications / Y.-C. Yang, J.-I. Guo // IEEE Transaction on Circuits and Systems for Video Technology. – 2009. – Vol. 19, no 9. – P. 1395–1399.

14. Recommendation ITU-T H.264 [Electronic resource]. – Mode of access : http://www.itu.int/rec/T-REC-H.264-200903-S/en. – Date of access : 31.05.2013.

15. Hardware Assisted Rate Distortion Optimization with Embedded CABAC Accelerator for the H.264 Advanced Video Codec / J. L. Nunez-Yanez [et al.] // IEEE Transactions on Consumer Electronics. – 2006. – Vol. 52, no. 2. – P. 590–597.

16. Optimizing the critical loop in the H.264/AVC CABAC decoder / H. Eeckhaut [et al.] // In Proc. 2006 IEEE International Conference on Field Programmable Technology (FPT2006). – Bangkok, Thailand, 2006. – P. 113–118.

17. H.264/MPEG-4 AVC Reference Software [Electronic resource]. – Mode of access : http://iphome.hhi.de/suehring/tml. – Date of access : 31.05.2013.

18. Novel Pipeline Design for H.264 CABAC Decoding / J. Zheng [et al.] // Pacific Rim Conference on Multimedia 2007. – Hong Kong, China, 2007. – P. 559–568.

Поступила 25.03.2013

Белорусский государственный университет
информатики и радиоэлектроники,
Минск, П. Бровки, 6
e-mail: {petrovsky, stankevich, palex}@bsuir.by

Al.A. Petrovsky, A.V. Stankevich, A.A. Petrovsky

PIPELINE ARCHITECTURE OF H.264/AVC STANDARD CABAC DECODER FOR MOBILE APPLICATIONS

The paper describes a three-stage pipeline architecture implementation of the CABAC decoder for mobile applications, with image resolution up to 625SD. The decoder architecture is suggested for pipeline calculations with the decoding performance of one bin per clock cycle. The decoder is compatible with profiles high profile, high 10 profile and high 4:2:2 profile and supports regime MBAFF and 8×8 blocks. It is scalable both in the resolution and in the supported decoding tools described in standard H.264. A comparison of our implementation with implementations of a prototype CABAC decoder on FPGA from the company Xilinx is given.

УДК 621.396.96

Р.Х. Садыхов, С.А. Кучук

СИСТЕМЫ ВИДЕОНАБЛЮДЕНИЯ: СОСТОЯНИЕ, ПРОБЛЕМЫ И ТЕХНИЧЕСКИЕ СРЕДСТВА ОБРАБОТКИ ИЗОБРАЖЕНИЙ

Рассматриваются системы видеонаблюдения средних и крупных объектов, используемое оборудование. Освещаются вопросы обработки изображений, тенденции развития видеосистем, очерчивается круг проблем, требующих решения.

Введение

С помощью систем видеонаблюдения решается широкий круг важных для общества задач: обеспечиваются сохранность имущества и безопасность граждан, снижаются затраты на персонал, поддерживается приток денег в бюджет государства. Видеонаблюдение применяется в казино, супермаркетах, увеселительных заведениях, клиниках, заводах, на трансформаторных подстанциях, объектах топливно-энергетического комплекса, в химической промышленности (резервуарах хранения химически активных веществ, опасных для людей и экологии), на парковках, улицах, в приемопередатчиках сотовой связи, офисных объектах, торговых центрах, крупных жилых комплексах, вузах, школах, на стадионах, нефтепроводах, заправках и т. п. [1].

Анализируя состояние дел в области видеонаблюдения в мире, авторы акцентируют внимание на возможностях и потребностях рынка, очерчивают круг решаемых и требующих решения задач в области обработки изображений, показывают направления перспективных научных исследований.

1. Технические средства

К техническим средствам систем видеонаблюдения можно отнести видеокамеры, средства передачи данных и технического оснащения мест операторов и др., устройства видеозаписи и сигнализации.

1.1. Видеокамеры

По способу передачи данных камеры делятся на аналоговые и цифровые [2].

Цифровые камеры опережают аналоговые [3] по качеству изображения, стоимости их интеграции в крупные системы видеонаблюдения. С ростом популярности цена цифровых камер уменьшается и становится сравнимой с ценой аналоговых [4].

В новых системах видеонаблюдения среднего и крупного масштаба установлены цифровые камеры, в старых используются аналоговые. В смешанных системах присутствуют как унаследованные аналоговые, так и новые цифровые видеокамеры. Широко распространенными системами, в состав которых могут входить камеры видеонаблюдения, являются Орион (Болид), Vista 50P, NAC.

Уличные камеры устанавливают на стенах зданий или столбах. Их оснащают водонепроницаемыми термокожухами, системами подогрева [5], защиты от вандалов, беспроводной или оптоволоконной связью.

Распространенные модели камер (ACTi Corp., ACUMEN Int. Corp., Arecont Vision, AVtech, Axis Communications, BOSCH Security Systems, D-Link, Merit Li-Lin, MOBOTIX AG, Pelco, Samsung Electronics, SANYO, Sony, Verint) выполняют, как правило, схожий набор функций [6–8]:

- улучшают изображения;
- вырезают зоны маскирования (например, зону ввода ПИН-кода);
- сжимают данные;

– выявляют движение, звук, дым и огонь;

– сигнализируют при неисправностях, закрытии и загрязнении объектива, поломке осветителя, засветке, расфокусировке, изменении угла обзора, изменении положения камеры;

– выделяют лица;

– предоставляют веб-доступ;

– выполняют специальные функции: подсчет количества людей, определение направления их движения, распознавание автомобильных номеров.

К недостаткам камер, использующих протокол IP (IP-камер), относят пропадание сигнала в сети, задержки при передаче данных, необходимость настройки, потерю качества изображения при сжатии. К настоящему времени большинство этих недостатков преодолены в той или иной степени [9]. Прогноз востребованности кабельного телевидения высокой четкости (HDcctv) как альтернативы IP-камерам низкий [9].

IP-камера улучшает получаемые изображения, затем сжимает их и передает в сеть (рис. 1). Потребители данных могут получать их как изнутри, так и вне сети видеонаблюдения.

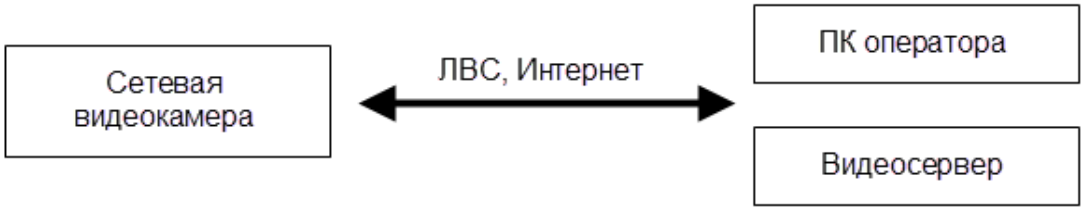


Рис. 1. Схема передачи данных потребителям от IP-видеокамеры

Все сетевые камеры улучшают изображение [6–8]: проводят стабилизацию изображений (при дрожании, вибрации), подавление и удаление шумов, цифровое масштабирование, компенсацию дефектных пикселов и пиковой яркости.

Хотя в цифровых камерах не применяются аналоговые стандарты телевизионной передачи, изображения с большим разрешением практически не используются по ряду причин: цена на камеры повышается из-за качественной оптики; требуются дополнительные затраты на хранение, передачу и обработку данных; во многих системах нужно фиксировать лица людей, поэтому камеры устанавливают с разных углов. Таким образом, наиболее востребованы 1,3–5-мегапиксельные камеры (таблица) [9]. Видеокамеры с более высоким разрешением используют на входах в здание, на открытых площадях, где не стоит задача идентификации людей [10, 11].

Популярные разрешения форматов видеоданных IP-камер [12]

Соотношение сторон	Разрешение
4:3	1,3 Мпкс: 1280x960, 2 Мпкс: 1600x1200
16:9	0,9 Мпкс HD720: 1280x720, 2 Мпкс FullHD1080: 1920x1080
Ближе к 16:9	1 Мпкс: 1280x800

Для определения требуемых параметров камеры исходят также из ширины целевой зоны и целей обработки изображений [13]. Допустим, что ширина целевой зоны, которую нужно охватить камерой, – 5 м. Цель системы определяет требуемое качество входного изображения (фиксация событий 100 пкс/м, распознавание знаков автомобилей 170–190 пкс/м, идентификация личности 250–270 пкс/м), например 270 пкс/м. Произведение нужного качества входного изображения на ширину зоны обзора дает требуемое разрешение по горизонтали IP-камеры, например 1350 пкс. Если выбрана камера с разрешением, меньшим рассчитанного, используют две и более камеры. Далее выбирается матрица в зависимости от требуемого качества картинки: отличное (на базе ПЗС-матриц) или среднее (на базе КМОП-матриц). Камера устанавливается от зоны наблюдения на некотором отдалении, например 20 м. Отдаление и ширина зоны – парамет-

ры для расчета фокусного расстояния (или угла обзора камеры по горизонтали). Так получается вторая характеристика требуемой камеры.

Угол обзора характеризует охватываемую камерой зону и качество различимых деталей объектов, позволяющих их идентифицировать. Он зависит от расстояния и требований к видимым деталям охватываемых объектов [5]. Угол обзора бюджетных видеокамер составляет 43–87 º по горизонтали и 33–71 º – по вертикали.

Чувствительность – минимальный уровень света, необходимый для получения приемлемого изображения. Требуемая чувствительность камеры зависит от наличия источников света, необходимости работы ночью. Для наблюдения освещенных автомагистралей в сумерки подходят камеры с чувствительностью 10 лк, для условий безлунной ночи без освещения требуются камеры с чувствительностью 0,01 лк [5].

Камеры, которые должны работать при низкой, переменной освещенности, оснащают режимом «день/ночь», механическим оптическим инфракрасным фильтром, автоматической регулировкой диафрагмы. Для участков с затемненными и светлыми участками используют камеры с уровнем динамического диапазона более 100 дБ [12]. Инфракрасная подсветка позволяет камере видеть 3–10 м пространства перед собой.

Развивается стандарт питания камер по сетевому кабелю IEEE 802.3af Power over Ethernet (PoE) [14]. PoE не применяется на особо опасных производствах, поскольку недостаточно взрывобезопасен [15]. В связи с тем что не все оборудование целиком соответствует PoE, перед покупкой оно проверяется на совместимость.

Для того чтобы соответствовать стандартам на энергопотребление и удешевить стоимость камер, цифровая обработка изображений осуществляется посредством микросхем, сжатие – систем на кристаллах, сетевые функции – встроенной ОС Linux [16].

Учитывая высокую ресурсоемкость видеоаналитики на высокопроизводительных видеосерверах, ширину каналов связи, необходимость оператора следить за показаниями нескольких камер, нет необходимости эксплуатировать камеры с высокой кадровой частотой. Кадровая частота камер должна составлять минимум 6 Гц, в среднем – 6-25 Гц.

1.2. Обмен данными

Для передачи данных потребителям применяют протоколы VLAN или VPN. Потребителями являются клиентское ПО постов охраны, видеосерверы, серверы обработки данных, мобильные устройства охраны, охранно-пожарные сигнализации, системы контроля и управления доступом, удаленные рабочие места [2]. Иногда доступ к камерам осуществляется через Интернет.

Устаревшие аналоговые камеры подключаются к сети через медиаконвертеры и видеорегистраторы с сетевым выходом, ПК с платой видеозахвата (до 16 камер) [2, 17] (рис. 2 и 3). Когда подключение происходит через ПК с платой видеозахвата, ПК может выступать в роли видеосервера.

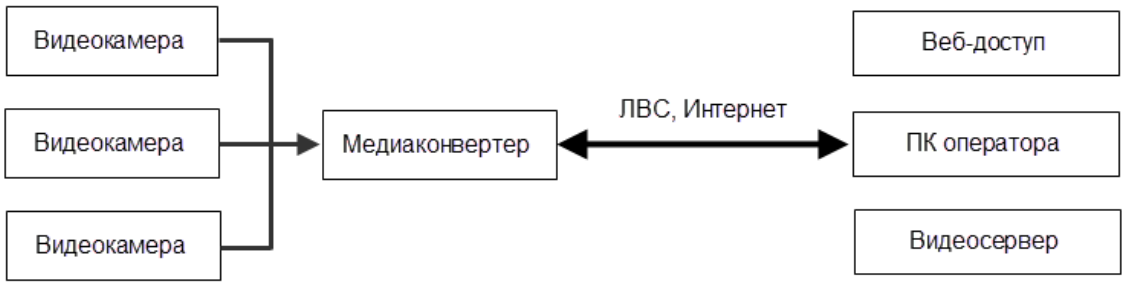


Рис. 2. Структурная схема центрально-распределенной системы
на базе аналоговых камер с медиаконвертером

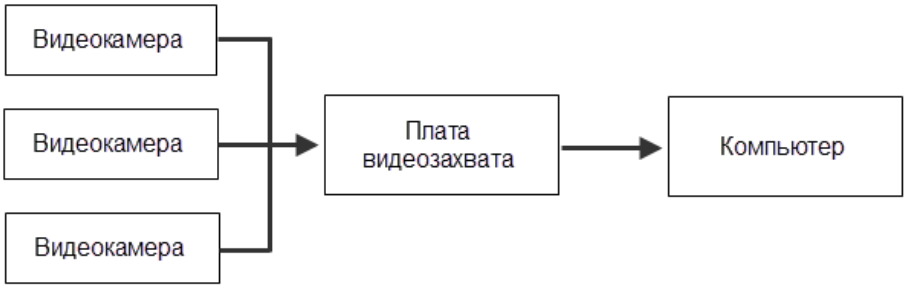


Рис. 3. Схема подключения аналоговых камер к сети посредством платы видеозахвата

В уличных камерах на столбах для защиты от грозы используются оптоволокно и беспроводные технологии: Wi-Fi, Bluetooth (дальность 50 м), IrDA OBEX, HomeRF (рис. 4) [2, 18, 19]. Наиболее популярна технология Wi-Fi несмотря на то, что скорость передачи данных в ней в значительной степени зависит от расстояния, погодных условий, наличия препятствий на пути сигнала, числа подключенных клиентов [18].

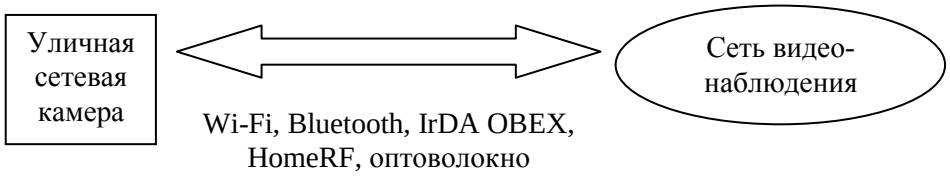


Рис. 4. Канал передачи данных от уличной камеры

Настройка камеры как сетевого устройства проводится квалифицированными администраторами, что гарантирует аутентичность данных, их защиту от несанкционированного доступа и подмены.

Для уменьшения объема видеокамера сжимает данные до их отправки в сеть. Популярные кодеки сжатия видеоданных (H.264, MPEG-4, JPEG, M-JPEG, Wavelet), как правило, реализованы аппаратно и позволяют уменьшить объем передаваемых данных по сети [16]. Существуют две проблемы: распаковки (на монитор оператора требуется одновременно выводить уменьшенные в разрешении видеопотоки порядка 16 камер) и упаковки (на видеосервере необходимо вести запись порядка 64 камер).

В настоящее время происходит постепенный переход на более эффективную реализацию сжатия – H.264/AVC/MPEG-4 Part 10 с расширением SVC [16], которая поддерживает быструю декомпрессию, предоставление одновременного доступа к данным устройствам с различным разрешением и скоростью соединения. Это позволяет уменьшить нагрузку на сеть, так как камере больше не нужно передавать данные в двух форматах одновременно: JPEG для ПО ПК операторов и мобильных устройств и MPEG-4 для видеосервера. Однако пока еще обратная совместимость сохраняется: большое количество моделей камер могут передавать данные параллельно в JPEG и MPEG-4 [20].

Разрабатывается интеллектуальное сжатие: движущиеся объекты передаются в высоком качестве, а статические объекты сжимаются с большими потерями [6]. Когда нет движения, изображение либо не передается, либо передается в ухудшенном качестве и реже, чтобы снизить объем передаваемых и обрабатываемых данных.

Хотя данные и сжимаются, объем их в средних и крупных системах настолько велик, что применяются специальные гигабитные сети, автономные от основных сетей организаций, обеспечивающие непрерывное функционирование этих систем. В таких специальных сетях используются гигабитные управляемые коммутаторы, которые обеспечивают скорость обмена данными с камерами 100 Мбит/с и имеют гигабитовый up-link на оптоволокно. Как правило, серверы, проводящие обработку данных, размещают территориально рядом с группой камер.

По мере появления стандартов все большее количество встроенных в камеры функций по обработке видеоданных унифицируется [6].

1.3. Устройства видеозаписи

Устройства видеозаписи представлены видеомагнитофонами, видеорегистраторами, видеосерверами.

Видеомагнитофоны исторически появились первыми. Сейчас они устарели и практически не используются. Эти устройства можно встретить в старых аналоговых системах видеонаблюдения, в которых они позволяют записывать порядка 40 суток видео на кассеты.

Видеорегистраторы – устройства, предназначенные для записи, хранения и воспроизведения видеоданных. Регистраторы используются в средних и небольших системах видеонаблюдения, а также в системах с аналоговыми камерами. Основные функции регистратора: предварительная обработка, запись на жесткий диск, флэш-память, воспроизведение видео, запись при наличии движения, включение сирены при движении, поддержка сетевых технологий (в том числе веб-доступа), удобный доступ к различным частям видео.

Регистраторы среднего ценового диапазона ($210–500) записывают видео в разрешениях 720 x 576 (6–26 Гц), 640 x 272 (25 Гц), более дорогие модели – 720 x 576 (25 Гц) [21]. Общей тенденцией развития регистраторов стала специализация по функциональному назначению, в том числе потребности в сети и веб-интерфейсе. Общие функции этих устройств включают возможность сохранять, обеспечивать поиск и воспроизводить видеоданные с флэш-памяти, DVD-RW, жестких дисков большого объема (до 36 ТБ); передавать данные в сеть, в том числе гигабитную; предоставлять доступ к данным через веб-интерфейс; управлять поворотными камерами; применять стандарт сжатия H.264.

Хотя многие регистраторы и поддерживают аналоговые камеры, замена вышедших из строя видеорегистраторов на новые в будущем будет связана с высокой вероятностью замены аналоговых на IP-камеры. Это связано с невозможностью ремонта регистраторов, ограничениями в предоставляемых функциях, высокой стоимостью программирования для регистраторов, низкой масштабируемостью в крупных системах на базе аналоговых камер.

Видеосерверы обрабатывают (в том числе пересжимают) и хранят данные. К каждому серверу по сети подключают группу камер (до 64, в среднем 24–32 0,3–1,3-мегапиксельные камеры) [12]. Обычно высокопроизводительный сервер размещают рядом с группой камер, которые он обслуживает в связи с большими объемами передаваемых по сети данных.

Если камера не передает данные в формате, который можно непосредственно записывать, их требуется преобразовывать из одного формата в другой. Это ресурсоемкая операция, поэтому используют производительные серверы, а запись при необходимости конвертации ведут со сниженной кадровой частотой 6 Гц (а не 25 Гц) [12]. Существует также необходимость сжатия изображений с применением ведеоаналитики, чтобы для решения задач видеонаблюдения не требовалось передавать большие объемы данных.

Для 16 мегапиксельных камер, передающих данные с кадровой частотой 6 Гц, за две недели накапливается 7 ТБ видеозаписей [22]. Перед интеллектуальными системами хранения стоит задача фиксировать происходящие события в более высоком качестве и с большей кадровой частотой, чем фиксируется отсутствие событий. Локально данные сохраняются на RAID-массивах (обычно это RAID-6 с автоматической заменой вышедших их строя дисков без остановки системы) [23]. В серверах используются наборы массивов жестких дисков, выбранных из разных партий, и специализированный высокопроизводительный контроллер, что в целом дорого, но обеспечивает надежное хранение и параллельное чтение данных [23]. Чтобы злоумышленники не могли уничтожить видеоданные, их могут хранить в других местах, в том числе на арендованном хостинге.

Хотя до сих пор формат для хранения данных не специфицирован, различные организации (например, CameraWatch) следят за тем, чтобы не хранилась конфиденциальная информация, некоторые фрагменты изображений и звук.

ПО серверов по возможности использует результаты обработки изображений камерами. Серверное ПО может отправлять в сеть результаты видеоанализа.

1.4. Места операторов

В небольших системах оператор оснащается специальным монитором или ПК, на который в готовом виде поступают данные от видеорегистратора.

Так как обычный ПК позволяет оператору видеть изображения только четырех камер одновременно при использовании сжатия MPEG-4, камеры передают данные для ПО ПК оператора в формате JPEG, что позволяет снижать затраты на оборудование и отображать на одном мониторе данные с 16 камер. Популярна также технология LED, при которой места операторов оборудуют 32–42-дюймовыми мониторами [4].

В крупных территориально распределенных системах ПК оператора оснащается четырехъядерным процессором и большими мониторами, чтобы оператор мог видеть данные с многих камер. Процессорная мощность необходима для преобразования данных на клиентском компьютере.

Важную роль играет видеоаналитика. Крупные охранные системы могут решать, на что обратить внимание оператора при наличии движения или срабатывании иных датчиков безопасности. Так повышается эффективность работы оператора: ему требуется следить только за теми камерами, где движение выделено с помощью видеоаналитики. Системы разбиваются территориально: оператор, который находится рядом с зоной наблюдения, следит за изображениями 10–100 камер. В крупных системах присутствует централизованный единый кризисный центр с различными специалистами [24].

Неудачные решения показывают, что без аналитики крупные системы неэффективны. Так, например, при обеспечении видеонаблюдения улиц Москвы в 2000-х гг. была создана система без аналитики, с использованием устаревшего оборудования, с неквалифицированным персоналом, высокой нагрузкой на каждого оператора (16 камер). Эффективность этой системы составила около 1 %.

1.5. Сигнализация

Камеры и регистраторы имеют тревожные выходы и могут активировать некоторые связанные с ними устройства тревоги. Локальные видеосерверы со специальным ПО также могут поднять тревогу при наступлении определенных условий. Серверное ПО, как правило, специализированное и обеспечивает более гибкую комплексную защиту, поддерживает больше устройств, обеспечивает специальные сценарии реагирования на угрозы.

2. Охранное телевидение

С помощью систем видеонаблюдения достигаются различные цели [15, 25–28] в областях сохранности имущества, защиты граждан, наблюдения за персоналом, доступа в помещения, а также в транспортном и домашнем секторе, медицине, при мониторинге оборудования.

Сохранность имущества – защита имущества от кражи и пожара, автоматическое пожаротушение. Фиксация правонарушений (нанесение повреждений объектам собственности, кражи) позволяет быстрее найти преступника, доказать правонарушение.

Защита граждан – комплексное понятие, включающее в себя видеонаблюдение в транспорте, учреждениях образования, на улице. Раннее выявление неадекватного поведения людей, драки, а также падающего, лежащего, бегущего, остановившегося в «тревожной» зоне, перелезающего ограду, поднимающегося вверх по пожарной лестнице дома человека позволяет решать эту задачу.

В России вкладываются деньги в программу «Безопасный город», в рамках которой отдельные города оборудуются камерами видеонаблюдения.

В чрезвычайных ситуациях комплексные системы видеонаблюдения не только оповещают, но и управляют эвакуацией: определяют тип чрезвычайной ситуации, ищут безопасный путь эвакуации, разблокируют двери выходов, включают индикаторы выхода, подают сигнал на динамики.

Для противодействия террористическим угрозам широко используется видеонаблюдение в системах охраны и безопасности для поиска оставленных под лестницами, в подвалах, на плат-

формах, в общественных оживленных местах, в метро предметов, для детекции переброшенных предметов.

С помощью видеонаблюдения (за кассами, из машин милиции, скорой помощи, такси) можно доказать надлежащее или ненадлежащее оказание услуг, обеспечить правовую защиту, следить за производством, ходом работ, предотвращать нецелевое использование имущества.

Охранное телевидение позволяет контролировать доступ людей в помещение посредством идентификации на контрольно-пропускном пункте.

Видеонаблюдение за грузоперевозками гарантирует сохранность имущества. Фиксация правонарушений, таких как угон или нанесение повреждений машине, позволяет быстрее найти преступника и доказать правонарушение. Контроль дорожного движения (распознавание номерных знаков) ускоряет возврат угнанных машин, оптимизирует движение (уменьшает число пробок, корректируя работу светофоров на перекрестках). Актуальна также задача защиты граждан в транспорте.

Повышение безопасности движения приносит существенную прибыль за счет автоматизированного сбора штрафов. Так, в России (Красногорск) система за 1,3 млн рос. руб. собирает ежегодно штрафы на сумму 3,6 млн рос. руб. за следующие правонарушения: превышение скорости на 50 км/ч и на 20 км/ч, проезд на красный свет, наличие мобильного телефона в руках водителя, неправильную парковку. Похожую систему стоимостью 250 млн рос. руб. собираются установить в Новгороде (окупаемость – 8 месяцев).

В медицинских учреждениях ведется видеонаблюдение за больными и детьми. Это мотивирует персонал к более качественной работе, пациентов – к более культурному поведению, регулирует отношения с пациентами в спорных случаях.

Мониторинг состояния оборудования, лифтов, показателей потребления услуг может происходить дистанционно, что сокращает расходы на обслуживание приёмопередатчиков сотовой связи, трансформаторных подстанций.

Комплексные системы обслуживания зданий «Умный дом» используют видеонаблюдение и для энергосбережения.

3. Задачи обработки видеоизображений

От современной IP-камеры ожидается хорошее качество видеоданных: изображения должны быть стабилизированы, шумы удалены, дефектные пикселы и пиковая яркость компенсированы [15, 29, 30].

3.1. Детекция наличия движения и типа объектов

Наиболее типичные задачи – определение наличия движения или остановки человека в «тревожной» зоне. В некоторых системах видеонаблюдения для определения движения используют маски [2] и вводят желаемый уровень детектирования угроз. Система обводит контурами движущиеся объекты на мониторе оператора.

3.2. Исключение влияния среды

При предварительной обработке изображения требуется исключать перепады освещенности постепенные (посредством динамической коррекции экспозиции) и резкие (отсвет, засвет фарами машины). Для исключения влияния резких перепадов освещенности используют пространственные характеристики: размер, скорость, направление движения и длительность нахождения объекта в зоне видеонаблюдения [29].

Чтобы не адаптировать алгоритмы обработки изображений к таким изменениям внешней среды, как качание камеры от ветра, вибрация, дрожание при установке камер на столбы, решается задача стабилизации изображения.

Помимо стабилизации, необходимо также эффективно исключать влияние теней от облаков, колыханий деревьев, кустов, травы и листвы, дождя, снега, пыли, туманов, песчаных бурь, насекомых на объективе и в поле камеры (в том числе при записи), пролетающих птиц, мусора, отражений в воде [15, 22, 29, 30].

3.3. Сжатие

При сжатии данных теряются детали движущихся, удаленных и привнесенных мелких объектов, если не использовать специальные алгоритмы. В связи с этим [2] камера детектирует движение до сжатия видеоданных, результаты детекции включаются в метаинформацию, движущиеся объекты передаются в более высоком качестве. Возникает необходимость в решении задач:

1) разработки алгоритмов на базе wavelet, которые бы обеспечивали лучшую передачу деталей для важных (движущихся) объектов, худшую – для фона, а также быструю декомпрессию, требуемое качество, разрешение и кадровую частоту видеоданных в зависимости от требований и технических возможностей пользователя;

2) разработки алгоритмов эффективной компрессии и быстрой декомпрессии (предлагается хранить не изображение, а происходящие события, используя семантическое сжатие информации, в которой заинтересован оператор [31]).

Сжатие, распаковка, преобразование медиаданных – ресурсоемкие процессы, зависящие от возможностей устройства (процессора, памяти, дисплея), количества обрабатываемых камер и канала связи.

Начиная с 2008 г. производители устройств стали добавлять поддержку стандарта сжатия H.264 High Profile в IP-камеры [20]. Расширение стандарта сжатия SVC позволяет использовать один видеопоток устройствам, различающимся разрешением и кадровой частотой экрана, производительностью процессора, шириной канала связи, и передавать интересующие детали изображений в более высоком качестве по сравнению с фоном.

3.4. Уменьшение использования ширины канала

Для эффективного использования канала связи до появления высокоэффективных методов сжатия было принято отправлять данные в двух потоках: в высоком разрешении для записи и в низком для отображения на дисплеях операторов.

На камере можно настраивать зоны маскирования, по которым данные не будут отправляться. Многие видеокамеры могут отправлять данные реже и в худшем качестве при отсутствии событий [30].

3.5. Развитие интеллектуальности аналитических систем

Видеоаналитика помогает персоналу охраны своевременно и быстро реагировать на ситуации, требующие внимания. На экране ПК или мобильного устройства оператора может отображаться схема объекта с маркерами событий «срабатывание датчиков», «появление целей», «огонь», «дым», «оставленный предмет» и др. Когда оператор отмечает курсором место на карте, на экране монитора появляются изображения с близлежащих камер.

3.6. Детекция и сопровождение целей

При охране периметра поворотными камерами стоит задача поворачивать камеры и сопровождать ими цели. При чтении автомобильных знаков камера должна поворачиваться так, чтобы отражение знака в определенное время суток не засвечивало камеру.

Для разгрузки оператора система может самостоятельно обводить контурами людей или их лица в видеопотоке, поступающем оператору, и показывать событийно-ориентированным образом нужную информацию (оператор следит за камерой только во время событий), в том числе траекторию движения. Одна из важных функций системы – не показывать повторно сигналы тревоги по движущемуся или не интересующему оператора объекту.

Востребован быстрый доступ к видеозаписям, например, при поиске определенного человека. В оживленных зонах иногда требуется подсчитывать количество людей, выделять траектории и направления их движения.

3.7. Защита от вандализма

Большое значение имеет самодиагностика всех элементов системы и защита от вандализма. На стороне видеокамеры также должны опознаваться ситуации закрытия и загрязнения

объектива, поломки осветителя, засветки, расфокусировки, изменения угла обзора, изменения положения камеры [15, 29, 30].

3.8. Распознавание номеров автомобилей

К системам распознавания номеров предъявляются жесткие требования: скорость движения машины до 150 км/ч, расстояние от камеры до машин до 75 м, угол наклона камеры к знаку до 40 º.

3.9. Удаление, привнесение, переброс объектов

Распознавание оставления, перебрасывания, удаления предмета – пока еще недостаточно хорошо решенная задача. Детекция оставленных предметов в оживленных зонах (например, метро) пока еще остается нерешенной задачей.

3.10. Идентификация лиц

Существующие на сегодняшний день системы идентификации лиц в их рабочих вариантах до сих пор еще считаются ненадежным способом аутентификации.

3.11. Детекция дыма и огня

Востребована задача детекции дыма и огня. Задача пока еще не решена достаточно хорошо.

3.12. Производительность

Краеугольный камень видеоаналитики – снижение стоимости вычислений. Видеосервер должен одновременно обрабатывать данные порядка 64 камер. Поэтому создаются графические библиотеки эффективной обработки изображений. Ставится задача создания эффективных алгоритмов сжатия изображений, которые бы позволяли устройствам, различающимся по своим техническим характеристикам, получать видеоданные, учитывали пропускную способность каналов связи, передавали интересующие фрагменты изображений в высоком качестве, удаляли мелкие изменения фона [15, 30, 39].

4. Перспективы систем видеонаблюдения

Перед разработчиками систем видеонаблюдения ставятся комплексные задачи усиления интеллектуализации видеоаналитики, обеспечения модульности компонент систем, облегчения их интеграции в более крупные системы, повышения мобильности.

4.1. Развитие серверной видеоаналитики

Несмотря на то что обработка изображений стала распределенной, программирование под конкретные модели камер мало востребовано ввиду зависимости от оборудования, высоких затрат на сопровождение, непереносимости кода на другое оборудование, ограниченности вычислительных возможностей видеокамеры, которая должна быть дешевой и следовать некоторым стандартам на энергопотребление [6, 16]. В настоящее время востребована серверная видеоаналитика, создаются платформы для упрощения разработки ПО обработки изображений, ведется поиск эффективных алгоритмов обработки данных, которые бы позволили снизить требования по производительности на видеосерверы. Обработка изображений – важный, но не единственный компонент сложной интегрированной системы, решающей несколько задач.

Системы создаются многофункциональными и интегрированными: состав обычной системы включает в себя подсистему распознавания автомобильных номеров, радиолокационную подсистему, систему контроля и управления доступом, подсистему обработки сигналов датчиков [30].

Комплексные системы делают модульными для того, чтобы у заказчика была возможность покупать фрагменты готовой системы, а не заказывать их разработку.

4.2. Упрощение установки и развертывания видеосистем

С развитием стандартизации, увеличением числа инсталляций сложность установки видеосистем на базе IP-камер постепенно снижается: многие системы настраиваются сами оптимальным способом, распознают, настраивают и задействуют подключенные устройства, обеспечивая минимальное участие пользователя в процессе установки. Развивается партнерская сеть, происходит обмен опытом на конференциях. Хотя системы и становятся проще, персонал все еще должен проходить предварительное обучение для работы с ними. В России известны системы ITV, SecurOS, VideoNet, Macroscop, среди них наиболее популярны ITV и VideoNet. В ITV реализованы детекторы движения, поворота камеры, потери качества изображения, оставленных предметов, пересечения линии в выбранном направлении, движения, остановки и длительного пребывания в зоне, входа и выхода объекта из зоны.

4.3. Децентрализация обработки данных

Обработка данных децентрализуется. Улучшение, упаковка, иная предварительная обработка, защита данных происходят внутри камеры, дальнейшая обработка – на локальных видеосерверах.

4.4. Стандартизация и сертификация

Во многих системах алгоритмы обработки изображений не тестируются на данных местности, где система будет работать, отсутствуют требования к алгоритмам и методики их тестирования. Поэтому для сравнения и тестирования универсальных систем МВД Великобритании предоставляет по запросу тестовые данные i-Lids (Imagery Library for Intelligent Detection Systems) и методику тестирования детекторов по пяти сценариям: «мониторинг стерильной зоны», «парковка», «обнаружение оставленного багажа», «наблюдение за дверьми», «построение траектории по нескольким камерам» (http://www.homeoffice.gov.uk/science-research/hosdb/i-lids/). Каждый сценарий имеет ряд записей при различных погодных условиях, освещенности, времени дня.

Максимальная дальность обнаружения движения камерой – это максимальное расстояние между камерой и движущимся человеком, при котором камера может фиксировать движение [29].

В настоящее время происходит отказ от стандартов телевизионной передачи данных PAL и NTSC и переход на новые стандарты, разрабатываемые организациями ONVIF и PSIA (в ONVIF состоит наибольшее число производителей аппаратуры). Новые стандарты регламентируют формат обмена видеоданными и ряд других аспектов видеосистем, облегчая интеграцию устройств в системы видеонаблюдения. Хотя в Европе и России популярнее стандарты ONVIF, разработчики видеокамер планируют поддерживать оба стандарта. ONVIF построены на технологии веб-сервисов (WSDL), протоколах RTP/RTSP, SOAP (xml), стандартах сжатия H.264, MPEG-4, MJPEG. С помощью стандартов ONVIF решаются следующие интеграционные проблемы: конфигурирования сетевого интерфейса, обнаружения устройств по протоколу WS-Discovery, управления профилями работы камеры, настройки поточной передачи медиаданных, обработки событий, управления поворотными камерами, видеоаналитики, защиты, хранения данных, настройки беспроводных интерфейсов, аутентификации, NVR [16]. Ход развития стандартов ONVIF позволяет говорить об обеспечении совместимости между различными видеоустройствами и простоте их подключения в будущем.

Таким образом, на сегодняшний день следование стандартам ONVIF или PSIA становится приоритетом для производителей видеокамер, поскольку позволяет создавать системы на базе видеокамер различных производителей [32].

По информации МВД Великобритании, более 80 % изображений, полученных полицией, все еще недостаточного качества, а в Англии порядка 90 % систем видеонаблюдения не соответствуют основным требованиям к записи и хранению изображений.

Один из форматов передачи данных – CIF [33]. Он предписывает несколько разрешений изображений: SQCIF (128 × 96), QCIF (176 × 144), CIF (352 × 288), 2CIF (Half D1) (704 × 288),

4CIF (D1) (704 × 576) и 16CIF (1408 × 1152). В сетевых камерах разрешающая способность может быть выше.

При работе в условиях низкой освещенности может применяться стандарт CEA-639 «Работа бытовых видеокамер и телекамер в условиях низкой освещенности». В методе измерения минимального уровня освещенности обычно используются четыре параметра телекамер: яркость, уровень черного, отношение сигнал-шум, разрешение.

Для защиты данных практически везде используется стандарт 802.1X.

Заключение

В настоящее время типичной конфигурацией, единой для крупной и средней систем видеонаблюдения, является территориально разделенная система, состоящая из ряда 1,3 мегапиксельных IP-видеокамер, транслирующих данные с кадровой частотой 6 Гц, сетевого оборудования (беспроводного для уличных камер), локального видеосервера для хранения и обработки данных, поста охраны. Эффективные системы – комплексные, которые решают несколько задач, имеют видеоаналитику, помогающую оператору увидеть динамику происходящего события и принять решение.

Общей тенденцией развития систем видеонаблюдения следует признать их интеллектуализацию, стандартизацию и унификацию оборудования, создание расширяемых многофункциональных модульных систем, на базе которых легко создать требуемую систему.

Список литературы

1. Система видеонаблюдения с применением протокола обмена данными SSP [Электронный ресурс]. – Режим доступа : http://www.akvilona.ru/serv/cctv.htm. – Дата доступа : 10.01.2012.

2. Система видеонаблюдения VideoInspector Professional [Электронный ресурс]. – Режим доступа : http://www.videomodul.ru/. – Дата доступа : 17.02.2012.

3. Шумейко, М. Передовые технологии для систем видеонаблюдения / М. Шумейко // F+S: технологии безопасности и противопожарной защиты. – 2009. – № 4 (40). – С. 4–6.

4. Алтуев, М.К. Рынок видеонаблюдения: бизнес-среда / М.К. Алтуев, К. Ванг, А.А. Кураков // Системы безопасности. – 2011. – № 2 (98). – С. 24–25.

5. Камеры видеонаблюдения, уличные камеры скрытого наружного наблюдения, миникамера слежения [Электронный ресурс]. – Режим доступа : http://www.video-vision.ru. – Дата доступа : 21.02.2012.

6. Белоусов, А.С. Какой «интеллект» нужен в IP-камерах? / А.С. Белоусов [Электронный ресурс]. – Режим доступа : http://ip-kamera.ru/articles609.html. – Дата доступа : 16.02.2012.

7. Савельев, М.А. Аналоговые и IP-камеры: конкуренция усиливается / М.А. Савельев // Каталог «CCTV» [Электронный ресурс]. – 2009. – Режим доступа : http://ip-kamera.ru/articles759.html. – Дата доступа : 20.02.2012.

8. Ерошин, Е.В. Интеллектуальные IP-камеры: что они умеют сейчас / Е.В. Ерошин // Системы безопасности. – 2009. – № 5 (89). – С. 136–136.

9. Васильев, А.Н. Мегапиксельное видеонаблюдение: IP и HDcctv. Опрос рынка и мнения экспертов / А.Н. Васильев, Р.Р. Шарифуллин, А.Н. Снегирев // Системы безопасности. – 2011. – № 2 (98). – С. 74–83.

10. Стрельцов, Р.В. Мегапиксельные видеокамеры: как извлечь максимальную выгоду / Р.В. Стрельцов // Системы безопасности. – 2010. – № 4 (94). – С. 34–35.

11. Птицын, Н.В. Мегапиксельная видеоаналитика для сложных систем видеонаблюдения / Н.В. Птицын, В.Б. Булычева // Там же. – С. 66–68.

12. Щербаков, И.А. Когда нужна высокая детализация? / И.А. Щербаков // Системы безопасности. – 2011. – № 1 (97). – С. 46–47.

13. Торубаров, А.А. Объективы для мегапиксельных камер: индивидуальный подбор / А.А. Торубаров // Системы безопасности. – 2010. – № 4 (94). – С. 52–55.

14. Питание через Ethernet (PoE) для видеонаблюдения [Электронный ресурс]. – Режим доступа : http://ip-kamera.ru/articles658.html. – Дата доступа : 20.02.2012.

15. Андрианов, Е.Ю. Системы видеонаблюдения с функциями видеоанализа для удаленных объектов / Е.Ю. Андрианов, С.Ю. Исправников, В.В. Старцев // Системы безопасности. – 2011. – № 2 (98). – С. 88–93.

16. Птицын, Н.В. Интеллект в IP-камере: горизонты возможностей / Н.В. Птицын // Системы безопасности. – 2009. – № 5 (89). – С. 130–134.

17. Аналоговые и цифровые системы видеонаблюдения [Электронный ресурс]. – Режим доступа : http://elites-montage.com.ua/svanalog.php.htm. – Дата доступа : 19.02.2012.

18. Беспроводное видеонаблюдение [Электронный ресурс]. – Режим доступа : http://ip-kamera.ru/articles340.html. – Дата доступа : 19.02.2012.

19. ИСО «ОРИОН»: Основные аспекты реализации функций охранной сигнализации // F+S: технологии безопасности и противопожарной защиты. – 2009. – № 3 (39). – С. 56–59.

20. Стандарт HD в CCTV [Электронный ресурс]. – Режим доступа : http://ip-kamera.ru/articles717.html. – Дата доступа : 14.02.2012.

21. Сравнительный обзор видеорегистраторов [Электронный ресурс]. – Режим доступа : http://security-ua.com. – Дата доступа : 20.02.2012.

22. Чижов, А.С. Видеоанализ в регионах / А.С. Чижов // Системы безопасности. – 2011. – № 3 (99). – С. 76–77.

23. Егоров, А. RAID 0, RAID 1, RAID 5, RAID 10 или что такое уровни RAID? / А. Егоров [Электронный ресурс]. – Режим доступа : http://ip-kamera.ru/articles647.html. – Дата доступа : 18.02.2012.

24. Организация рабочего места оператора видеонаблюдения при большом количестве IP камер [Электронный ресурс]. – Режим доступа : http://ip-kamera.ru/articles832.html. – Дата доступа : 20.02.2012.

25. Ермолаев, Е. Комплексные решения для контроля и управления высотными зданиями / Е. Ермолаев // Высотные здания. – 2008, июнь–июль. – С. 118–121.

26. Ерошин, Е.В. Новые рынки для сетевого видеонаблюдения / Е.В. Ерошин // Каталог CCTV [Электронный ресурс]. – 2010. – Режим доступа : http://ip-kamera.ru/articles813.html. – Дата доступа : 20.11.2011.

27. Горяченков, М. Задача дистанционного мониторинга и управления группой распределенных аппаратных объектов / М. Горяченков // Технологии безопасности и противопожарной защиты [Электронный ресурс]. – 2010. – № 1 (49). – Режим доступа : http://www.bolid.ru/netcat_files/monitoring_art.pdf. – Дата доступа : 03.01.2012.

28. Системы видеонаблюдения на транспорте в вопросах и ответах / Е.В. Ерошин [и др.] // Системы безопасности. – 2010. – № 3 (93). – С. 136–142.

29. Романович, Д. Лучшие способы применения видеоаналитики для наружного наблюдения / Д. Романович [Электронный ресурс]. – Режим доступа : http://dsslnews.com/ru/articles/articles-secyrity/1751-2011-06-01-00-31-51. – Дата доступа : 02.02.2012.

30. Пименов, А.В. Универсальные платформы видеонаблюдения на основе IP-видеосерверов / А.В. Пименов [Электронный ресурс]. – Режим доступа : http://ip-kamera.ru/articles983.html. – Дата доступа : 14.02.2012.

31. Хинкель, Р. Видеонаблюдение на транспорте: безопасность и бизнес / Р. Хинкель, У. Бартхелмес // Системы безопасности. – 2010. – № 3 (93). – С. 62–63.

32. Программное обеспечение для систем IP-видеонаблюдения [Электронный ресурс]. – Режим доступа : http://ip-kamera.ru/articles833.html. – Дата доступа : 17.02.2012.

33. Common Intermediate Format [Электронный ресурс]. – Режим доступа : http://ru.wikipedia.org/wiki/Common_Intermediate_Format. – Дата доступа : 21.02.2012.

Поступила 15.05.2012

Белорусский государственный университет
информатики и радиоэлектроники,
Минск, П. Бровки, 6
e-mail: rsadykhov@bsuir.by,
Cuchuk.Sergey@gmail.com

R.Kh. Sadykhov, S.A. Kuchuk

VIDEO SURVEILLANCE SYSTEMS: STATUS, PROBLEMS AND HARDWARE OF IMAGE PROCESSING

Video systems for surveillance of medium and large objects and these systems hardware (street IP-cameras, servers, data storage devices, operator devices) are surveyed. The issues of image processing, surveillance systems development tendencies and tasks requiring solution are covered.

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

УДК 534.26

Г.Ч. Шушкевич, Н.Н. Киселева

ПРОНИКНОВЕНИЕ ЗВУКОВОГО ПОЛЯ ЧЕРЕЗ МНОГОСЛОЙНУЮ СФЕРИЧЕСКУЮ ОБОЛОЧКУ

Рассматривается аналитико-численный алгоритм решения граничной задачи, описывающей процесс проникновения звукового поля сферического излучателя, который расположен внутри тонкой незамкнутой сферической оболочки, через проницаемую многослойную сферическую оболочку. Численно исследуется влияние некоторых параметров задачи на значение коэффициента ослабления (экранирования) звукового поля внутри сферической оболочки.

Введение

Исследование распространения звуковых волн в многослойных средах имеет большое количество практических приложений в электроакустике, гидроакустике, медицинской диагностике (дефектоскопии), биоакустике, конструировании многослойных звукопоглощающих панелей для защиты от шума и вибрации [1–6]. Библиография по решению задач рассеяния весьма обширна. Рассмотрим лишь некоторые работы, имеющие отношение к данной теме исследования.

Рассеяние звукового поля на двух разнесенных сферах (акустически мягких либо жестких) одинаковых или разных радиусов исследовано в работах [7–11]. Методом разделения переменных решена задача рассеивания плоской звуковой волны на пористой сфере [12] и сфере, покрытой эластичным пористым слоем [13]. В работах [14, 15] рассматривается рассеяние плоской звуковой волны на двух упругих сферических оболочках. В работе [16] рассмотрена дифракция звука на радиально-слоистой изотропной термоупругой сферической оболочке. Обратная плоская задача дифракции плоской звуковой волны на многослойном теле, состоящем из конечного числа однородных слоев, изучена в [17, 18]. Методом гибридной матрицы решена задача о прохождении плоской звуковой волны через плоскую многослойную структуру из пьезоэлектрических материалов [19]. Построено аналитическое решение задачи о прохождении ультразвуковой волны через плоскопараллельную пористую пьезоэлектрическую многослойную среду [20, 21]. В случае плоской акустической волны, распространяющейся в произвольном направлении, получены двухсторонние нелокальные граничные условия, связывающие акустические поля по обе стороны упругого слоя [22]. Эти граничные условия могут быть использованы для моделирования процессов проникновения акустических волн через тонкостенные упругие оболочки произвольной формы.

В данной работе построено точное осесимметричное решение задачи о проникновении звукового поля через систему проницаемых сферических оболочек. В качестве источника поля рассматривается сферический излучатель, расположенный внутри тонкой незамкнутой сферической оболочки. Используя соответствующие теоремы сложения [23], решение поставленной краевой задачи сведено к решению парных сумматорных уравнений по полиномам Лежандра, которые преобразуются к бесконечной системе линейных алгебраических уравнений второго рода с вполне непрерывным оператором. Исследуется влияние некоторых параметров задачи на значение коэффициента ослабления (экранирования) звукового поля внутри сферической оболочки.

1. Постановка задачи и представление решения

Пусть все пространство R^3 разделено концентрическими сферами $S_j\left(r_j = a_j\right)$, $j = 1, ..., s$, $s > 2$, с центром в точке O_1 на $s+1$ область: $D_0(a_1 < r_1 < \infty)$, $D_j(a_j < r_1 < a_{j+1})$,

$j = 1, 2, ..., s-1$, $D_s (0 \le r_1 < a_s)$ (рис.1). В области D_0 находится идеально тонкая незамкнутая сферическая оболочка Γ_1 с углом раствора θ_0, расположенная на сфере Γ радиуса a с центром в точке O. Область пространства, ограниченную сферой Γ, обозначим $D_0^{(0)} (0 \le r < a)$ и $D_0 = D_0^{(0)} \bigcup \Gamma \bigcup D_0^{(1)}$. Расстояние между точками O и O_1 обозначим через h.

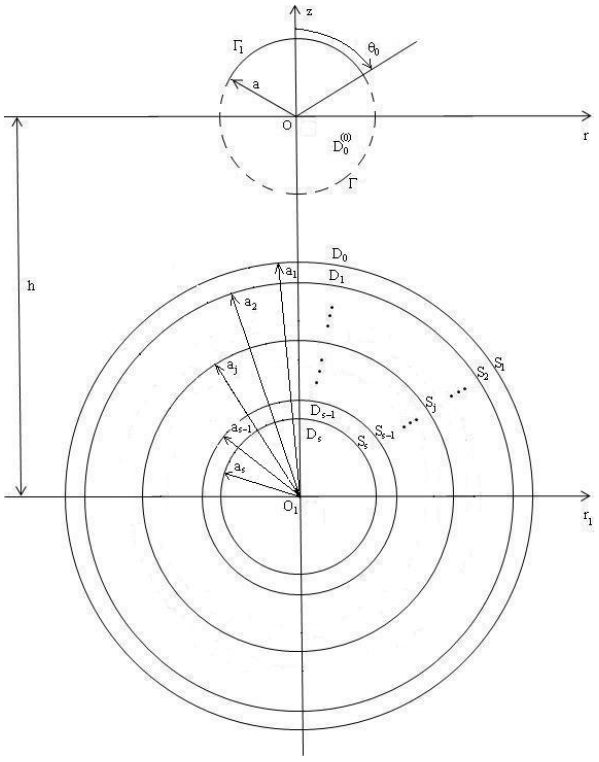


Рис. 1. Геометрия задачи

В точке O расположен точечный излучатель звуковых волн, колеблющихся с круговой частотой ω. Области D_j, $j = 0, 1, ..., s$, заполнены материалом, в котором не распространяются сдвиговые волны. Плотность среды и скорость звука в области D_j обозначим соответственно через ρ_j, c_j, $j = 0, 1, ..., s$.

Для решения задачи свяжем с точками O, O_1 сферические координаты $Or\theta\varphi$ и $O_1 r_1 \theta_1 \varphi$ соответственно. Сферическая оболочка Γ_1 описывается следующим образом:

$$\Gamma_1 = \{ r = a,\ 0 \le \theta \le \theta_0 < \pi,\ 0 \le \varphi \le 2\pi \}.$$

Обозначим через p_c давление звукового поля источника; $p_0^{(0)}$ – давление звукового поля, отраженного от границы Γ_1 в области $D_0^{(0)}$; $p_0^{(1)}$ – давление звукового поля, отраженного от границы Γ_1 в области $D_0^{(1)}$; $p_0^{(2)}$ – давление звукового поля, отраженного от границы S_1 в области $D_0^{(1)}$; $p_0 = p_c + p_0^{(0)}$ – суммарное давление звукового поля в области $D_0^{(0)}$; $p_0 = p_0^{(1)} + p_0^{(2)}$ – суммарное давление звукового поля в области $D_0^{(1)}$; p_j – давление звукового поля в области D_j, $j = 1, 2, ..., s$.

Решение дифракционной задачи сводится к нахождению давлений $p_0^{(0)}$, $p_0^{(1)}$, $p_0^{(2)}$, p_j, $j = 1, 2, ..., s$, удовлетворяющих:

– уравнению Гельмгольца

$$\Delta p_0^{(0)} + k_0^2 p_0^{(0)} = 0 \quad \text{в} \quad D_0^{(0)}; \quad \Delta p_0^{(1)} + k_0^2 p_0^{(1)} = 0, \quad \Delta p_0^{(2)} + k_0^2 p_0^{(2)} = 0 \quad \text{в} \quad D_0^{(0)};$$
$$\Delta p_j + k_j^2 p_j = 0 \quad \text{в} \quad D_j,$$

где $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ – оператор Лапласа; $k_j = \omega / c_j$ – волновое число;

– граничному условию на поверхности сферической оболочки Γ_1 (акустически жесткой оболочки)

$$\frac{\partial}{\partial \vec{n}} \left(p_c + p_0^{(0)} \right) \Big|_{\Gamma_1} = 0, \tag{1}$$

где $\vec{n}$ – нормаль к поверхности Γ_1;

– граничным условиям на поверхности сферы S_j, $j = 1, 2, ..., s$,

$$p_{j-1}\big|_{S_j} = p_j\big|_{S_j}, \qquad \frac{1}{\rho_{j-1}} \frac{\partial p_{j-1}}{\partial \vec{n}}\bigg|_{S_j} = \frac{1}{\rho_j} \frac{\partial p_j}{\partial \vec{n}}\bigg|_{S_j}, \tag{2}$$

где $\vec{n}$ – нормаль к поверхности S_j;

– условию на бесконечности [24, 25]

$$\lim_{M \to \infty} r \cdot \left(\frac{\partial p_0(M)}{\partial r} - ik_0 p_0(M) \right) = 0, \tag{3}$$

где M – произвольная точка пространства.

Потребуем также выполнения условия непрерывности давлений на открытой части сферической оболочки $\Gamma \setminus \Gamma_1$:

$$\left(p_c + p_0^{(0)} \right) \Big|_{\Gamma \setminus \Gamma_1} = \left(p_0^{(1)} + p_0^{(2)} \right) \Big|_{\Gamma \setminus \Gamma_1} \tag{4}$$

и нормальной производной на поверхности сферы Γ:

$$\frac{\partial}{\partial \vec{n}} \left(p_c + p_0^{(0)} \right) \Big|_{\Gamma} = \frac{\partial}{\partial \vec{n}} \left(p_0^{(1)} + p_0^{(2)} \right) \Big|_{\Gamma}, \tag{5}$$

где $\vec{n}$ – нормаль к поверхности Γ.

Реальные звуковые давления вычисляются по формуле

$$P_j = Re\left(p_j e^{-i\omega t} \right),$$

где i – мнимая единица, $j = 0, 1, ..., s$.

Давление исходного звукового поля представим в виде ряда по сферическим волновым функциям [23, 24]:

$$p_c(r, \theta) = P \frac{e^{ik_0 r}}{r} = ik_0 P h_0^{(1)}(kr) = P \sum_{n=0}^{\infty} f_n h_n^{(1)}(k_0 r) P_n(\cos\theta), \quad f_n = ik_0 \delta_{0n}, \tag{6}$$

где $h_n^{(1)}(kr)$ – сферические функции Ханкеля; $P_n(\cos\theta)$ – полиномы Лежандра [26]; δ_{0n} – символ Кронекера; P – const [25].

Представим давление p_j рассеянного звукового поля в области D_j, $j = 0, 1, ..., s$, в виде суперпозиции базисных решений уравнения Гельмгольца, принимая во внимание условие на бесконечности (3):

$$p_0^{(0)}(r,\theta) = P\sum_{n=o}^{\infty} c_n j_n(k_0 r) P_n(\cos\theta),\ r < a; \tag{7}$$

$$p_0^{(1)}(r,\theta) = P\sum_{n=o}^{\infty} x_n h_n^{(1)}(k_0 r) P_n(\cos\theta),\ r > a\ ;\ p_0^{(2)}(r_1,\theta_1) = P\sum_{n=o}^{\infty} y_n h_n^{(1)}(k_0 r_1) P_n(\cos\theta_1),\ r_1 > a_1; \tag{8}$$

$$p_j(r_1,\theta_1) = P\sum_{n=0}^{\infty} a_n^{(j)} j_n(k_j r_1) P_n(\cos\theta_1) + P\sum_{n=0}^{\infty} b_n^{(j)} h_n^{(1)}(k_j r_1) P_n(\cos\theta_1) \text{ в } D_j,\ j = 1, 2, \ldots, s-1; \tag{9}$$

$$p_s(r_1,\theta_1) = P\sum_{n=0}^{\infty} d_n j_n(k_s r_1) P_n(\cos\theta_1) \text{ в } D_s, \tag{10}$$

где $j_n(kr)$ – сферические функции Бесселя первого рода [26].

Неизвестные коэффициенты c_n, x_n, y_n, $a_n^{(j)}$, $b_n^{(j)}$, d_n подлежат определению из граничных условий.

2. Выполнение граничных условий

Выполним граничные условия (1), (4), (5). Для этого представим функцию $p_0^{(2)}(r_1,\theta_1)$ через сферические волновые функции в системе координат с началом в точке O, используя формулу [23, 24]

$$h_n^{(1)}(k_0 r_1) P_n(\cos\theta_1) = \sum_{l=0}^{\infty} A_{nl}^0(h)\, j_l(k_0 r) P_l(\cos\theta),\ r < h,$$

где $A_{nl}^0(h) = (2l+1)\sum_{\sigma=|l-n|}^{l+n} i^{\sigma+l-n} b_\sigma^{(n\,0\,0)} h_\sigma^{(1)}(k_0 h)$; $b_\sigma^{(n0q0)} = (nq00|\sigma 0)^2$, $nq00|\sigma 0$ – коэффициенты Клебша – Гордона [24].

Тогда

$$p_0^{(2)}(r,\theta) = P\sum_{n=o}^{\infty} T_n j_n(k_0 r) P_n(\cos\theta),\quad T_n = \sum_{k=o}^{\infty} y_k A_{kn}^0(h). \tag{11}$$

Принимая во внимание представления (6), (7), (11), условие непрерывности (5) с учетом условия ортогональности полиномов Лежандра на отрезке $[0;\pi]$ примет вид

$$f_n \frac{\mathrm{d}}{\mathrm{d}\xi} h_n^{(1)}(\xi_0) + c_n \frac{d}{d\xi} j_n(\xi_0) = x_n \frac{d}{d\xi} h_n^{(1)}(\xi_0) + T_n \frac{d}{d\xi} j_n(\xi_0),\ \xi = k_0 a,\ n = 0, 1, \ldots. \tag{12}$$

Выполним граничное условие (1) на поверхности сферической оболочки Γ_1 и условие непрерывности (4). В полученных уравнениях исключим коэффициенты c_n с помощью представления (12) и найдем парные сумматорные уравнения по полиномам Лежандра вида

$$\begin{cases} \sum_{n=o}^{\infty} x_n \dfrac{d}{d\xi_0} h_n^{(1)}(\xi_0) P_n(\cos\theta) = -\sum_{n=o}^{\infty} T_n \dfrac{d}{d\xi_0} j_n(\xi_0) P_n(\cos\theta),\quad 0 \le \theta < \theta_0; \\ \sum_{n=o}^{\infty} \dfrac{x_n - f_n}{\dfrac{d}{d\xi_0} j_n(\xi_0)} P_n(\cos\theta) = 0,\quad \theta_0 < \theta \le \pi. \end{cases} \tag{13}$$

Для преобразования парных сумматорных уравнений (13) введем в рассмотрение новые коэффициенты X_n по формуле

$$x_n = X_n \frac{d}{d\xi_0} j_n(\xi_0) + f_n,\ n = 0, 1, \ldots, \tag{14}$$

и малый параметр g_n по формуле

$$g_n = 1 + \frac{4i\xi_0^3}{2n+1}\frac{d}{d\xi_0} j_n(\xi_0)\frac{d}{d\xi_0} h_n^{(1)}(\xi_0),\ g_n = O\left(n^{-2}\right). \tag{15}$$

В результате парные сумматорные уравнения (13) преобразуются к виду

$$\begin{cases} \sum\limits_{n=0}^{\infty}(2n+1)(1-g_n)X_n P_n(\cos\theta) = \sum\limits_{n=o}^{\infty}(2n+1)(\tilde{f}_n+\tilde{T}_n)P_n(\cos\theta), \quad 0 \le \theta < \theta_0; \\ \sum\limits_{n=0}^{\infty} X_n P_n(\cos\theta) = 0, \quad \theta_0 < \theta \le \pi, \end{cases} \tag{16}$$

где

$$\tilde{T}_n = 4i\xi_0^3 T_n \frac{d}{d\xi_0} j_n(\xi_0)/(2n+1), \quad \tilde{f}_n = 4i\xi_0^3 f_n \frac{d}{d\xi_0} h_n^{(1)}(\xi_0)/(2n+1). \tag{17}$$

Используя интегральные представления для полиномов Лежандра, парные сумматорные уравнения (16) преобразуем к бесконечной системе линейных алгебраических уравнений (СЛАУ) второго порядка с вполне непрерывным оператором [27, 28]

$$X_n - \sum_{k=0}^{\infty} g_k R_{nk} X_k = \sum_{k=0}^{\infty}\left(\tilde{T}_k + \tilde{f}_k\right)R_{nk},\ n = 0, 1, \ldots, \tag{18}$$

$$R_{nk} = \frac{2}{\pi}\int_0^{\theta_0} \sin(n+0{,}5)t \sin(k+0{,}5)t\, dt,$$

$$R_{nk} = \frac{1}{\pi}\left[\frac{\sin(n-k)\theta_0}{n-k} - \frac{\sin(n+k+1)\theta_0}{n+k+1}\right],\ \left.\frac{\sin(n-k)\theta_0}{n-k}\right|_{n=k} = \theta_0.$$

Для выполнения граничных условий (2) представим функцию $p_0^{(1)}(r,\theta)$ через сферические волновые функции в системе координат с началом в точке O_1, используя формулу [23, 24]

$$h_n^{(1)}(k_0 r)P_n(\cos\theta) = \sum_{l=0}^{\infty} B_{nl}^0(h)\, j_l(k_0 r_1)P_l(\cos\theta_1),\ r_1 < h,$$

где

$$B_{nl}^0(h) = (2l+1)\sum_{\sigma=|l-n|}^{l+n}(-1)^{\sigma} i^{\sigma+l-n} b_{\sigma}^{(n0l0)} h_{\sigma}^{(1)}(k_0 h).$$

Тогда

$$p_0^{(1)}(r_1, \theta_1) = P\sum_{n=0}^{\infty} Z_n j_n(k_0 r_1)P_n(\cos\theta_1),\ Z_n = \sum_{p=0}^{\infty} x_p B_{pn}^0(h). \tag{19}$$

Принимая во внимание представления давлений (9), (10), (19) и выполняя граничные условия (2) с учетом ортогональности полиномов Лежандра на отрезке $[0;\ \pi]$, получим

$$M^{(j-1,j)}(n)V^{(j-1)}(n) = M^{(j,j)}(n)V^{(j)}(n),\ V^{(j)}(n) = P^{(j-1,j)}(n)V^{(j-1)}(n),\ j = 1, 2, \ldots, s-1; \tag{20}$$

$$M^{(s-1,s)}(n)V^{(s-1)}(n) = d_n E^{(s)}(n), \tag{21}$$

где $V^{(j)}(n)$, $E^{(s)}(n)$ – векторы-столбцы равномерности 2; $M^{(j-1,j)}(n)$, $M^{(j,j)}(n)$, $P^{(j-1,j)}(n)$ – матрицы размерности 2×2;

$$V^{(0)}(n)=\begin{pmatrix} Z_n \\ y_n \end{pmatrix};\ V^{(j)}(n)=\begin{pmatrix} a_n^{(j)} \\ b_n^{(j)} \end{pmatrix};\ P^{(j-1,j)}(n)=\left(M^{(j,j)}(n)\right)^{-1} M^{(j-1,j)}(n);$$

$$M^{(j-1,j)}(n)=\begin{pmatrix} m_{11}^{(j-1,j)}(n) & m_{12}^{(j-1,j)}(n) \\ m_{21}^{(j-1,j)}(n) & m_{22}^{(j-1,j)}(n) \end{pmatrix};\ M^{(j,j)}(n)=\begin{pmatrix} m_{11}^{(j,j)}(n) & m_{12}^{(j,j)}(n) \\ m_{21}^{(j,j)}(n) & m_{22}^{(j,j)}(n) \end{pmatrix};$$

$$m_{11}^{(j-1,j)}(n)=j_n\left(\xi_{j-1,\,j}\right);\ m_{12}^{(j-1,j)}(n)=h_n^{(1)}\left(\xi_{j-1,\,j}\right);$$

$$m_{21}^{(j-1,j)}(n)=\frac{k_{j-1}}{\rho_{j-1}}\frac{d}{d\xi_{j-1,\,j}}j_n\left(\xi_{j-1,\,j}\right);\ m_{22}^{(j-1,j)}(n)=\frac{k_{j-1}}{\rho_{j-1}}\frac{d}{d\xi_{j-1,\,j}}h_n^{(1)}\left(\xi_{j-1,\,j}\right);\ \xi_{j-1,\,j}=k_{j-1}a_j;$$

$$m_{11}^{(j,j)}(n)=j_n\left(\xi_{j,j}\right);\ m_{12}^{(j,j)}(n)=h_n^{(1)}\left(\xi_{j,j}\right);$$

$$m_{21}^{(j,j)}(n)=\frac{k_j}{\rho_j}\frac{d}{d\xi_{j,j}}j_n\left(\xi_{j,j}\right);\ m_{22}^{(j,j)}(n)=\frac{k_j}{\rho_j}\frac{d}{d\xi_{j,j}}h_n^{(1)}\left(\xi_{j,j}\right);\ \xi_{j,j}=k_j a_j;$$

$$E^{(s)}(n)=\begin{pmatrix} e_1^{(s)}(n) \\ e_2^{(s)}(n) \end{pmatrix};\ e_1^{(s)}(n)=j_n\left(\xi_{s,s}\right);\ e_2^{(s)}(n)=\frac{k_s}{\rho_s}\frac{d}{d\xi_{s,s}}j_n\left(\xi_{s,s}\right);\ \xi_{s,s}=k_s a_s.$$

Итерационно из (20), (21) получим

$$M^{(s-1,s)}(n)P^{(s-2,s-1)}(n)P^{(s-3,s-2)}(n)\cdot\ldots\cdot P^{(0,1)}(n)V^{(0)}(n)=d_nE^{(s)}(n)$$

или

$$C(n)V^{(0)}(n)=d_nE^{(s)}(n), \tag{22}$$

где

$$C(n)=M^{(s-1,s)}(n)P^{(s-2,s-1)}(n)P^{(s-3,s-2)}(n)\cdot\ldots\cdot P^{(0,1)}(n)=\begin{pmatrix} c_{11}(n) & c_{12}(n) \\ c_{21}(n) & c_{22}(n) \end{pmatrix}.$$

Из (22) исключим коэффициенты d_n. Для этого умножим обе части (22) на вектор-строку

$$\bar{E}^{(s)}(n)=\left(e_2^{(s)}(n),\quad -e_1^{(s)}(n)\right)$$

и получим

$$W(n)V^{(0)}(n)=0,\ W(n)=\bar{E}^{(s)}(n)C(n)=\left(w_1(n),\quad w_2(n)\right),$$

или

$$w_1(n)Z_n+w_2(n)y_n=0,\ y_n=-\frac{w_1(n)}{w_2(n)}Z_n. \tag{23}$$

Из представлений (11), (14), (17), (19), (23) следует связь между коэффициентами $\tilde{T}_k$ и X_p:

$$\tilde{T}_k=-\sum_{p=0}^{\infty}S_{pk}X_p+\tilde{\tilde{f}}_k,\ k=0,1,2,\ldots, \tag{24}$$

где

$$S_{pk}=4i\xi_0^3\frac{d}{d\xi_0}j_p(\xi_0)\frac{d}{d\xi_0}j_k(\xi_0)\sum_{m=0}^{\infty}\frac{w_1(m)}{w_2(m)}B_{pm}(h)A_{mk}(h)/(2k+1);$$

$$\tilde{\tilde{f}}_k=4k_0\xi_0^3\frac{d}{d\xi_0}j_k(\xi_0)\sum_{m=0}^{\infty}\frac{w_1(m)}{w_2(m)}B_{0m}(h)A_{mk}(h)/(2k+1).$$

Преобразуем правую часть системы (18). Исключим из правой части коэффициенты $\tilde{T}_k$ с помощью представления (24) и получим бесконечную СЛАУ второго порядка:

$$X_n - \sum_{k=0}^{\infty}(g_k R_{nk} - \alpha_{nk})X_k = \sum_{k=0}^{\infty}\left(\tilde{f}_k + \tilde{\tilde{f}}_k\right)R_{nk}\ ,\ n = 0, 1, 2, \ldots , \qquad (25)$$

где

$$\alpha_{nk} = \sum_{p=0}^{\infty} R_{np} S_{kp}\ .$$

Найдем связь между коэффициентами d_n, входящими в представление давления в области D_s, и решением системы (25). Для этого (22) рассмотрим как систему вида

$$\begin{cases} d_n e_1^{(s)}(n) - c_{12}(n)\, y_n = c_{11}(n) Z_n; \\ d_n e_2^{(s)}(n) - c_{22}(n)\, y_n = c_{21}(n) Z_n, \end{cases}$$

из которой следует, что

$$d_n = \frac{\bar{\Delta}_1(n)}{\Delta(n)} Z_n, \qquad (26)$$

где

$$\Delta(n) = c_{12}(n) e_2^{(s)}(n) - e_1^{(s)}(n) c_{22}(n),\ \bar{\Delta}_1(n) = c_{12}(n) c_{21}(n) - c_{11}(n) c_{22}(n).$$

Согласно представлениям (14), (19) из (26) получим связь между коэффициентами d_n и X_p:

$$d_n = \frac{\bar{\Delta}_1(n)}{\Delta(n)} \sum_{p=0}^{\infty}\left(X_p \frac{d}{d\xi_0} j_p(\xi_0) + f_p \right) B_{pn}^0(h).$$

Коэффициент ослабления звукового поля в области D_s вычислим по формуле

$$K(r_1, \theta_1) = \left| p_s(r_1, \theta_1) \right| / \left| p_c(r_1, \theta_1) \right|,\ \ 0 \le r_1 \le a_s,$$

где

$$p_c(r_1, \theta_1) = Pik_0 \sum_{n=0}^{\infty} B_{0n}^0(h)\, j_n(k_0 r_1) P_n(\cos\theta_1).$$

3. Вычислительный эксперимент

С помощью системы компьютерной математики Mathcad [29] были проведены вычисления коэффициента ослабления $K(r_1, \theta_1)$ звукового поля в области D_s для некоторых параметров задачи.

Сферические функции $j_n(x)$, $h_n^{(1)}(x) = j_n(x) + iy_n(x)$ вычислялись с помощью встроенных функций $js(n, x)$ и $ys(n, x)$ [29, с. 268]. Здесь $y_n(x)$ – сферическая функция Бесселя второго рода [26]. Производные сферических функций вычислялись с помощью формулы [26, с. 258]

$$\frac{d}{dx} f_n(x) = n f_n(x) / x - f_{n+1}(x), \qquad n = 0, 1, 2, \ldots$$

Коэффициенты Клебша – Гордона $b_{\sigma}^{(n0q0)}$ вычислялись по формуле (3.4.17) [24, с. 127].

Бесконечная система (25) решалась методом усечения [24, 30]. Вычислительный эксперимент показал, что порядок усечения для рассмотренных параметров задачи можно взять равным 75. Это обеспечивает решение системы (25) с точностью 10^{-4}. Все бесконечные суммы вычислялись с точностью 10^{-10} [31].

Графики коэффициента ослабления (экранирования) звукового поля $K(r_1,\theta_1), 0 \le \theta_1 \le 180^\circ$, трехслойным сферическим экраном для некоторых значений r_1 и $\theta_0 = 90^\circ$, $a = 0{,}3$ м, $a_1 = 1$ м, $a_2 = 0{,}99$ м, $a_3 = 0{,}97$ м, $a_4 = 0{,}96$ м, $h = 2$ м, $f = 50$ Гц, $\omega = 2\pi f$, если области D_0, D_4 заполнены воздухом ($\rho_0 = \rho_4 = 1{,}225$ кг/м^3, $c_0 = c_4 = 343$ м/с), области D_1, D_3 – органическим стеклом ($\rho_1 = \rho_3 = 1200$ кг/м^3, $c_1 = c_3 = 2565$ м/с), показаны на рис. 2, *а* (область D_2 заполнена воздухом), на рис. 2, *б* (область D_2 заполнена пресной водой: $\rho_2 = 1000$ кг/м^3, $c_2 = 1483$ м/с), на рис. 2, *в* (область D_2 заполнена льдом: $\rho_2 = 900$ кг/м^3, $c_2 = 3980$ м/с), на рис. 2, *г* (область D_2 заполнена гелием: $\rho_2 = 0{,}18$ кг/м^3, $c_2 = 970$ м/с).

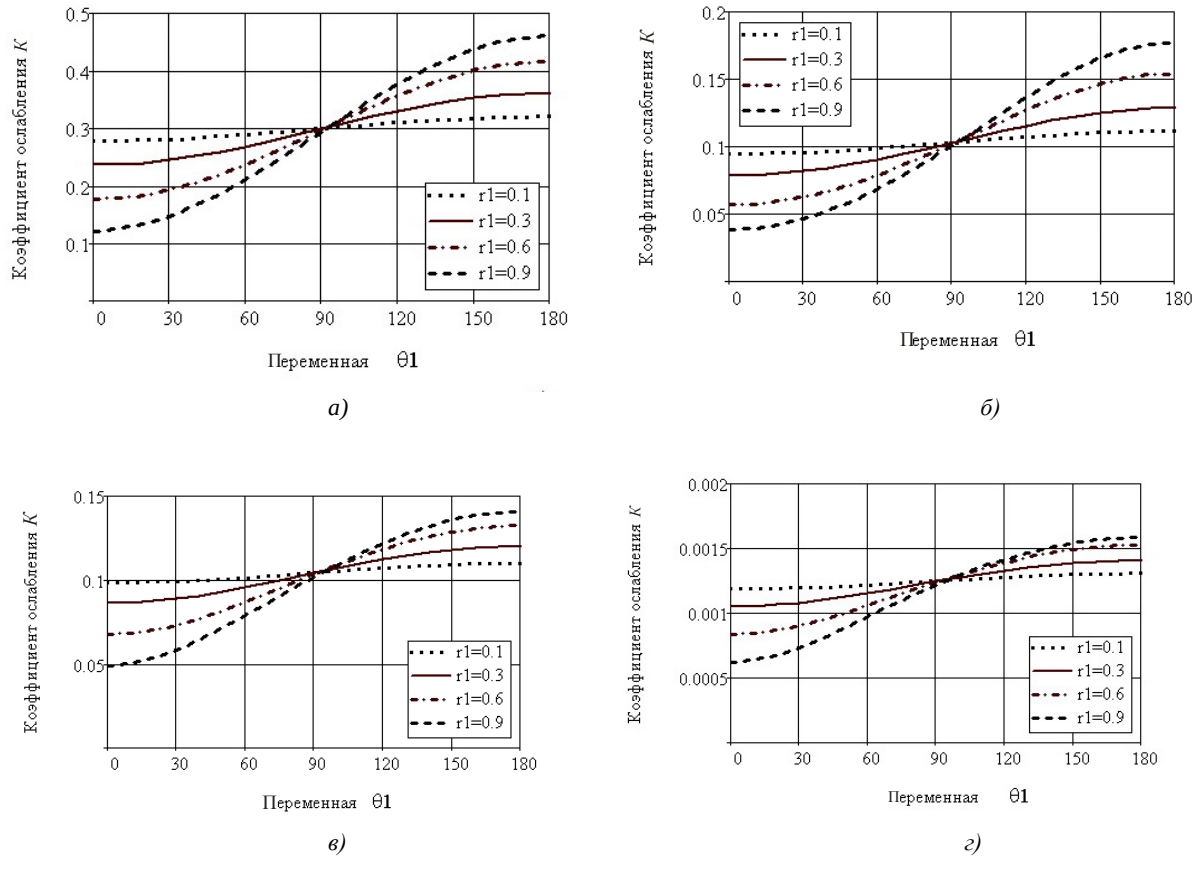
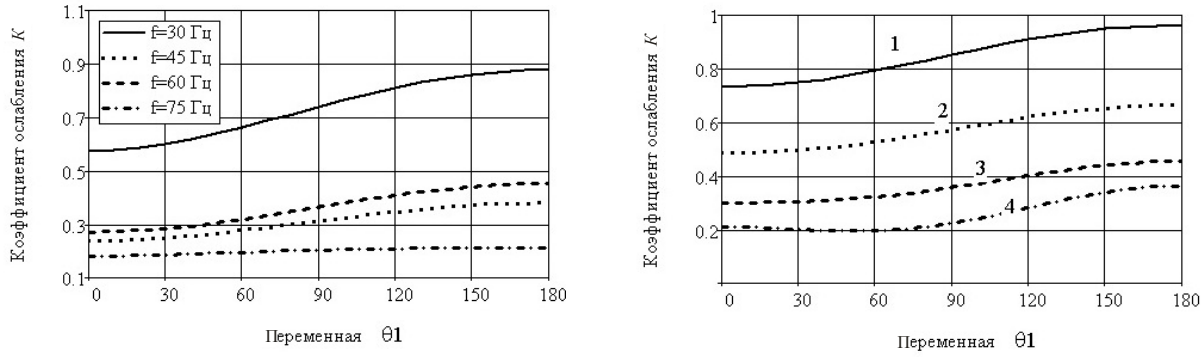


Рис. 2. Графики коэффициента ослабления звукового поля $K(r_1,\theta_1)$ для некоторых значений r_1

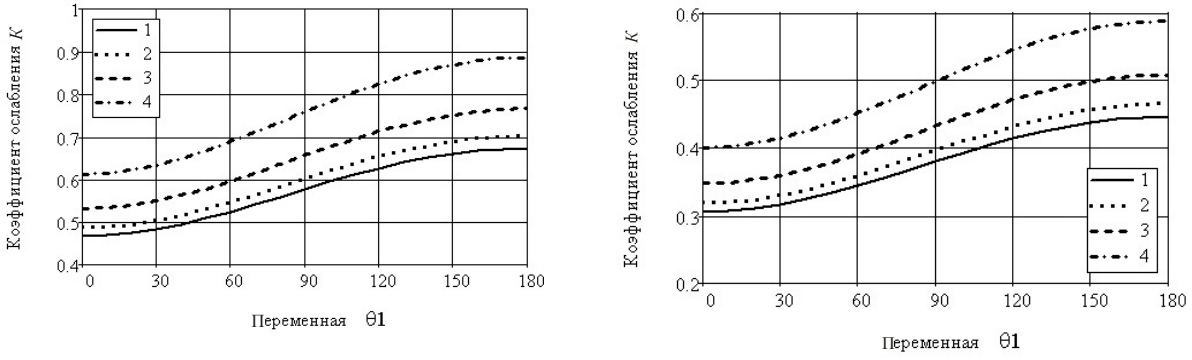
Графики коэффициента ослабления (экранирования) звукового поля $K(0{,}5;\theta_1)$, $0 \le \theta_1 \le 180^0$, трехслойным сферическим экраном для различных значений частоты звука f, если области D_0, D_2, D_4 заполнены воздухом, области D_1, D_3 – органическим стеклом, $\theta_0 = 45^0$, $a = 0{,}1$ м, $a_1 = 1$ м, $a_2 = 0{,}98$ м, $a_3 = 0{,}96$ м, $a_4 = 0{,}95$ м, $h = 2{,}5$ м, показаны на рис. 3, *а*.



а) *б)*

Рис. 3. Графики коэффициента ослабления звукового поля $K(0,5;\theta_1)$ для различных значений частоты звука f

Графики коэффициента ослабления (экранирования) звукового поля $K(0,5;\theta_1)$, $0 \le \theta_1 \le 180^0$, трехслойным сферическим экраном для различных значений частоты звука f: f =10 Гц (1), f =20 Гц (2), f =30 Гц (3), f =40 Гц (4), если области D_0, D_2 заполнены воздухом, область D_1 – льдом, D_3 – органическим стеклом, D_4 – пресной водой, $\theta_0 = 30^0$, $a = 0,1$ м, $a_1 = 1$м, $a_2 = 0,98$ м, $a_3 = 0,97$ м, $a_4 = 0,96$ м, h=5м, показаны на рис. 3, *б*.



а) *б)*

Рис. 4. Графики коэффициента ослабления звукового поля $K(0,5;\theta_1)$ для различных значений угла раствора θ_0 сферической оболочки Γ_1

Графики коэффициента ослабления (экранирования) звукового поля $K(0,5;\theta_1)$, $0 \le \theta_1 \le 180^0$, трехслойным сферическим экраном для различных значений угла раствора сферической оболочки θ_0: 1 – $\theta_0 = 10^0$, 2 – $\theta_0 = 60^0$, 3 – $\theta_0 = 90^0$, 4 – $\theta_0 = 120^0$, если области D_0, D_2, D_4 заполнены воздухом, области D_1, D_3 – органическим стеклом, $f = 30$ Гц, $a = 0,5$ м, $a_1 = 1$ м, $a_2 = 0,98$ *м*, $a_3 = 0,97$ м, $a_4 = 0,96$ м, изображены на рис. 4, *а*; графики коэффициента ослабления (экранирования) звукового поля $K(0,5;\theta_1)$ в случае, когда область D_2 заполнена водой, – на рис. 4, *б*.

Заключение

С помощью теоремы сложения для сферических волновых функций решение задачи о проникновении звукового поля через проницаемую многослойную сферическую оболочку сведено к решению парных сумматорных уравнений по полиномам Лежандра. Парные уравнения

преобразованы к бесконечной системе линейных алгебраических уравнений второго рода с вполне непрерывным оператором. В качестве источника звукового поля рассматривается сферический излучатель, расположенный внутри тонкой незамкнутой сферической оболочки.

Численно исследовано влияние геометрических параметров задачи, плотности сред и скорости звука на значение коэффициента ослабления поля для трехслойного сферического экрана. Вычислительные эксперименты показали, что если второй сферический слой экрана заполнен веществом с малой плотностью, эффективность экранирования значительно увеличивается. Коэффициент ослабления поля зависит от круговой частоты звукового источника ω и угла раствора сферической оболочки θ_0: с увеличением частоты ω коэффициент экранирования уменьшается, с увеличением угла раствора θ_0 – увеличивается. Разработанная методика и программное обеспечение могут найти практическое использование при проектировании многослойных звуковых экранов.

Список литературы

1. Бреховских, Л.М. Волны в слоистых средах / Л.М. Бреховских. – М. : Изд-во АН СССР, 1957. – 502 с.

2. Иванов, Н.И. Инженерная акустика. Теория и практика борьбы с шумом / Н.И. Иванов. – М. : Логос, 2008. – 424 с.

3. 6th National Conference ACOUSTICS 2012 [Electronic resource]. – Mode of access : http://conferences.ionio.gr/acoustics2012/en. – Date of access : 08.01.2013.

4. International Conference on Noise and Vibration Engineering [Electronic resource]. – Mode of access : http://www.isma-isaac.be. – Date of access : 09.01.2013.

5. Second International Conference of Acoustics and Vibration, ISAV2012, Engineering [Electronic resource]. – Mode of access : http://isav.ir/2012/index.php. – Date of access : 11.01.2013.

6. III Всерос. науч.-практ. конф. с междунар. участием «Защита населения от повышенного шумового воздействия» [Электронный ресурс]. – Режим доступа : http://onlinereg.ru/ noise2011. – Дата доступа : 11.01.2013.

7. Шебеко, Г.А. Дифракция скалярной сферической волны на нескольких шарах, расположенных в полупространстве / Г.А. Шебеко // Вестник БГУ. Сер. 1. – 1970. – № 3. – С. 5–10.

8. Марневская, Л.А. Решение некоторых задач дифракции звуковых волн на сферах и сферических приемниках : автореф. дис. ... канд. физ.-мат. наук : 01.01.02 / Л.А. Марневская; Бел. гос. ун-т. – Минск, 1979. – 18 с.

9. Acoustic scattering by a pair of spheres / C.G. Gaunaurd [et al.] // J. Acous. Soc. Amer. – 1995. – Vol. 98. – P. 495–507.

10. Gabrielli, P. Acoustic scattering by two spheres: Multiple scattering and symmetry considerations / P. Gabrielli, M. Mercier-Finidori // J. of Sound and Vibration. – 2001. – Vol. 241. – P. 423–439.

11. Румелиотис, Дж.А. Рассеяние звуковых волн на двух сферических телах, одно из которых имеет малый радиус / Дж. А. Румелиотис, А.Д. Котсис // Акустический журнал. – 2007. – Т. 5, № 1. – С. 38–49.

12. Huang, L.N. Trapping and absorption of sound waves I a screened sphere / L.N. Huang // Wave Motion. – 1990. – Vol. 12, № 1. – P. 1–13.

13. Huang, L.N. Trapping and absorption of sound waves II a sphere covered with a porous layer / L.N. Huang // Wave Motion. – 1990. – Vol. 12, № 5. – P. 401–414.

14. Huang, H.H. Acoustic scattering of a plane wave by two spherical elastic shells / H.H. Huang, G.C. Gaunaurd // J. Acous Soc. Amer. – 1995. – Vol. 98. – P. 2149–2156.

15. Huang, H.H. Acoustic scattering of a plane wave by two spherical elastic shells above the coincidence frequency / H.H. Huang, G.C. Gaunaurd // J. Acous Soc. Amer. – 1997. – Vol. 101. – P. 2659–2668.

16. Ларин, Н.В. Рассеяние звука неоднородным термоупругим сферическим слоем / Н.В. Ларин, Л.А. Толоконников // Прикладная математика и механика. – 2010. – Т. 74, № 4. – С. 645–654.

17. Guozheng, Y. The Far Field Operator for a Multilayered Scatterer / Y. Guozheng, Z. Huijiang // J. Computers and Mathematics with Applications. – 2002. – Vol. 43. – P. 631–639.

18. Guozheng, Y. Inverse Scattering by a Multilayered Obstacle / Y. Guozheng // J. Computers and Mathematics with Applications. – 2004. – Vol. 48. – P. 1801–1810.

19. Acoustic wave transmission through piezoelectric structured materials / M. Lam [et al.] // J. Ultrasonics. – 2009. – Vol. 49. – P. 424–431.

20. Vashishth, A. K. Ultrasonic wave's interaction at fluid-porous piezoelectric layered interface / A. K. Vashishth, V. Gupta // J. Ultrasonics. – 2013. – Vol. 53. – P. 479–974.

21. Acoustic waves in solid and fluid layered materials / E.H. El Boudouti [et al.] // J. Surface Science Reports. – 2009. – Vol. 64. – P. 471–594.

22. Ерофеенко, В.Т. Моделирование двухсторонних граничных условий для акустических волн на упругом экране / В.Т. Ерофеенко // Весці НАН Беларусі. – 2010. – № 4. – С. 76–84.

23. Ерофеенко, В.Т. Теоремы сложения / В.Т. Ерофеенко. – Минск : Наука и техника, 1989. – 240 с.

24. Иванов, Е.А. Дифракция электромагнитных волн на двух телах / Е.А. Иванов. – Минск : Наука и техника, 1968. – 584 с.

25. Шендарев, Е.Л. Излучение и рассеяние звука. / Е.Л. Шендарев. – Л. : Судостроение, 1989. – 304 с.

26. Справочник по специальным функциям с формулами, графиками и таблицами / под ред. М. Абрамовича, И. Стиган. – М. : Наука, 1979. – 830 с.

27. Шушкевич, Г.Ч. Расчет электростатических полей методом парных, тройных уравнений с использованием теорем сложения / Г.Ч. Шушкевич. – Гродно : ГрГУ, 1999. – 238 с.

28. Резуненко, В.А. Дифракция плоской звуковой волны на сфере с круговым отверстием / В.А. Резуненко // Вісник Харків. нац. універ. ім. В.Н. Каразіна. Сер. Мат., прик. мат. і мех. – 2009. – № 850. – С. 71–77.

29. Шушкевич, Г.Ч. Компьютерные технологии в математике. Система Mathcad 14. Ч. 1. / Г.Ч. Шушкевич, С.В. Шушкевич. – Минск : Изд-во Гревцова, 2010. – 287 с.

30. Каханер, Д. Численные методы и программное обеспечение / Д. Каханер, К. Моулер, С. Нэш. – М. : Мир, 1998. – 576 с.

31. Вержбицкий, В.М. Основы численных методов / В.М. Вержбицкий. – М. : Высшая школа, 2002. – 848 с.

Поступила 22.01.2013

Гродненский государственный
университет им. Янки Купалы,
Гродно, Ожешко, 22
e-mail: g_shu@tut.by

G.Ch. Shushkevich, N.N. Kiselyova

PENETRATION OF A SOUND FIELD THROUGH A MULTILAYERED SPHERICAL SHELL

An analytical solution of the boundary problem describing the process of penetration of the sound field of a spherical emitter located inside a thin unclosed spherical shell through a permeable multilayered spherical shell is considered. The influence of some parameters of the problem on the value of the sound field weakening (screening) coefficient is studied via a numerical simulation.

УДК 537.311.322

О.А. Козлова, В.В. Нелаев

AB INITIO МОДЕЛИРОВАНИЕ ЭЛЕКТРОННЫХ СВОЙСТВ ДВУХМЕРНОГО МОЛИБДЕНИТА

Посредством ab initio, из первых принципов, моделирования исследуются электронные свойства трехмерной и двухмерной структур молибденита, MoS_2, сформированных вдоль <001>, <010> и <100> кристаллографических направлений. Проводятся расчеты электронной плотности и зонной структуры. Показывается, что зонные структуры 2D-<010> и -<100> MoS_2 идентичны. Зонная структура 2D-<001> MoS_2 отличается наличием прямозонного перехода и отсутствием дополнительных энергетических уровней. Обнаруженные особенности подтверждают возможность наличия в MoS_2, как и у графена, исключительных электронных и магнитных свойств.

Введение

Ab initio моделирование электронных и спин-зависимых свойств материалов является эффективным теоретическим инструментом исследований [1, 2]. Основная особенность *ab initio* приближения, предназначенного для описания эволюции и энергетических свойств электронно-ядерной подсистемы, состоит в том, что в нем не требуется, в отличие от большинства других теоретических методов, наличие эмпирических или полуэмпирических «подгоночных» под эксперимент параметров. Все параметры в *ab initio* теории основаны на фундаментальных физических константах, таких как заряд электрона, масса электрона, скорость света и т. д. С использованием *ab initio* моделирования детально исследованы, например, анизотропные свойства таких материалов микроэлектроники и солнечной энергетики, как форсит [3], пирит [4], литиофилит [5], оксид титана [6], деформированный кремний и германий [7], ферромагнитные материалы и активные металлы для наношнуров [8–10]. Модельный кристаллит в большинстве этих работ представлял собой трехмерную структуру.

Вместе с тем и двухмерные структуры сегодня демонстрируют целый ряд исключительных физических свойств и соответственно возможностей практического применения. Примером такой двухмерной структуры является графен [11, 12], который уже сегодня находит широкое применение в различных сферах – от наноэлектроники до приборов и устройств космической техники. Возможно, существуют и многокомпонентные соединения, в двухмерных структурах которых полезные электронные и магнитные свойства будут проявляться в еще большей степени, чем в трехмерных. Так, дисульфид молибдена (MoS_2), содержащий в своей объемной кристаллической структуре низкоразмерные гетерометаллические фрагменты, проявляет ярко выраженную анизотропию электрофизических и магнитных свойств; в нем могут быть достигнуты высокие значения магнитострикции и может резко проявляться магнитокалорический эффект [13]. Экспериментальные и теоретические исследования показывают [14–18], что MoS_2 является перспективным материалом наноэлектроники и солнечной энергетики. В ряде работ подтверждена возможность практического применения различных структурных модификаций дисульфида молибдена в виде нанотрубок различных форм, интеркаляций, фуллеридов и однослойных структур [14, 16, 18]. Указанные особенности физических свойств обычной трехмерной структуры молибденита стимулируют теоретические исследования свойств его двухмерной структуры, тем более что его однослойная модификация уже находит практическое применение в качестве материала структурного элемента транзистора [14].

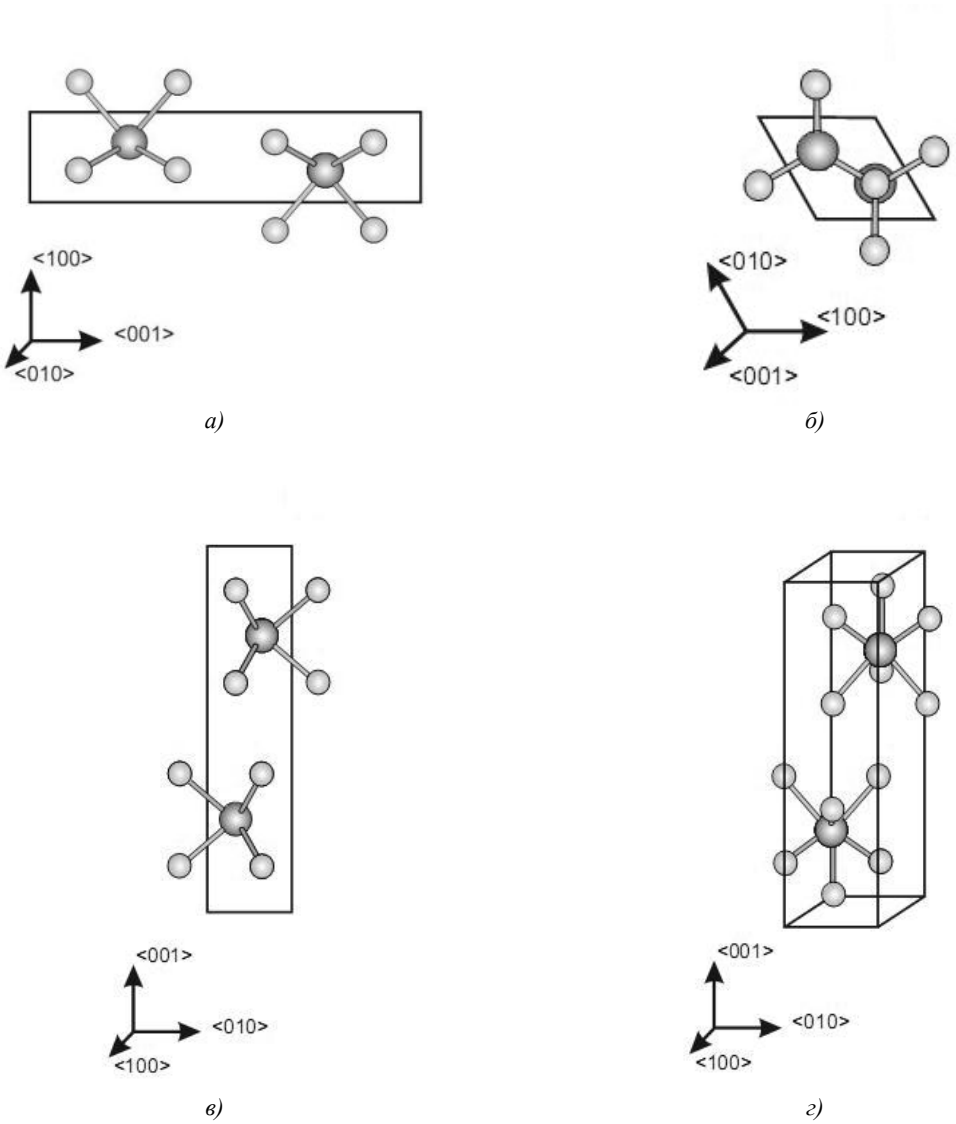
Ab initio моделирование электронных свойств двухмерного (2D-) MoS_2 по трем основным кристаллографическим направлениям <001>, <010> и <100> позволяет детально описать анизотропные электронные свойства этого соединения (структуру зонной диаграммы, значения и форму волновых функций), установить их зависимость от кристаллографической ориентации и атомной конфигурации элементарной ячейки.

Моделирование на квантово-механическом уровне представляет собой в конечном счете описание исследуемой электронно-ядерной системы на языке волновых функций и заданного гамильтониана системы посредством численного решения уравнения Шредингера. Реализация этой задачи требует колоссальных вычислительных ресурсов и может быть осуществлена только с использованием многопроцессорных вычислительных кластеров в среде грид-технологий. Представленные результаты моделирования были получены с помощью суперкомпьютера СКИФ.

1. Кристаллография MoS_2

Межатомные взаимодействия в кристалле MoS_2 определяются кристаллографическими параметрами и типом химических связей. Природный MoS_2 (α-MoS_2) имеет гексагональную структуру (тригональная призма) слоистого типа, подобную по форме структуре графита.

Объемная элементарная ячейка MoS_2 (рис. 1) представляет собой гексагональную структуру пространственной группы $P6_3/mmc$ (№ 194). Ячейка состоит из четырех прямоугольных граней со сторонами 0,316 и 1,272 нм, параллельных оси <010> и пересекающих оси <100> и <001> на разных расстояниях. В основании ячейки лежит ромб с углом, равным 120 º [19, 20].



а) *б)* *в)* *г)*

Рис. 1. Элементарные ячейки MoS_2: *а)* в направлении <010>; *б)* в направлении <001>; *в)* в направлении <100>; *г)* в объемной структуре

2. Методология

В качестве инструмента *ab initio* расчетов использовался программный комплекс VASP [1, 21], предназначенный для моделирования атомно-молекулярных и электронно-ядерных систем методами квантовой механики и молекулярной динамики. Взаимодействие между ионами и электронами моделируемой системы описывается посредством псевдопотенциального подхода и метода присоединенных плоских волн (PAW-метода).

Координаты атомов исходного кристаллита MoS_2 были выбраны в соответствии с кристаллографическими ячейками, представленными на рис. 1. *Ab initio* моделирование проводилось в рамках теории функционала электронной плотности с использованием обобщенного градиентного приближения (GGA-PBE). Для ускорения сходимости в расчетах выбрана величина энергии «обрезания» кинетической энергии E_{cut}=500 эВ подобно расчетной процедуре, изложенной в работе [15]. Разбиение обратного пространства на сетку 15x15x1 осуществлялось посредством использования Gamma-схемы (gamma-центрированной сетки для гексагональных систем) [21].

С целью физически адекватного представления двухмерности структуры использовалась процедура введения «вакуумного промежутка» – методического приема, предназначенного для исключения межатомного взаимодействия при определенном расстоянии между отдельными слоями кристалла молибденита. Значение этого промежутка выбиралось равным расстоянию между слоями, при котором химические связи полностью разрываются. Для 2D<001>, <010>, <100>-структур MoS_2 эти величины составили соответственно 9,1; 2,34 и 2,34 Å.

Перед расчетом электронных свойств исследуемой структуры проводилась обычная для *ab initio* методологии процедура релаксации, заключающаяся в определении координат атомов, при которых потенциальная энергия системы минимальна.

3. Результаты моделирования электронных характеристик MoS_2

Выполнены расчеты электронной плотности и зонной диаграммы моделируемой системы. Расчет плотности электронных состояний (Density Of States, DOS) осуществлялся при заданных минимальной и максимальной потенциальных энергий системы (9,5 и 9,0 эВ соответственно).

Рассчитаны электронные плотности для 2D<001>, <010>, <100>-структур MoS_2 (рис. 2).

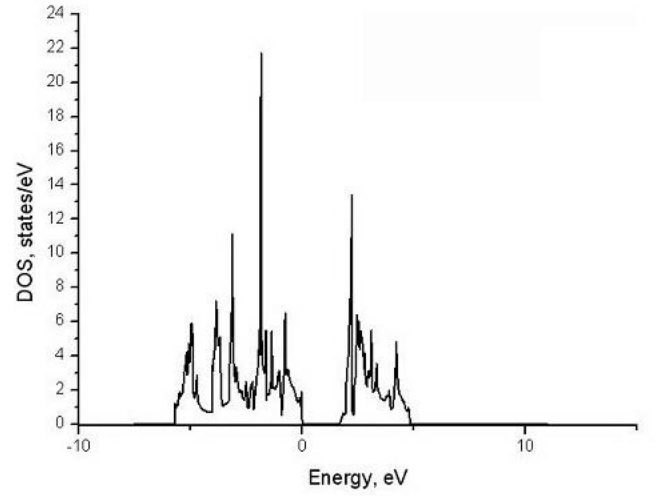
На основании результатов расчетов электронной плотности определены значения ширины запрещенной зоны для всех исследованных 2D-структур MoS_2. Эти величины оказались равными 1,71; 1,24 и 1,24 эВ при значениях уровня Ферми 1,97; 0,21 и 0,21 эВ соответственно для <001>-, <010>-, <100>-структур MoS_2. Следует отметить, что для <001>-структуры MoS_2 переход из валентной зоны в зону проводимости носит прямозонный характер.

Незначительные пики вблизи уровня Ферми на зависимостях электронной плотности для 2D<010>, <100>-структур MoS_2 (рис. 2, *б* и *в*), соответствующие расщеплению энергетических уровней (рис. 3), представляют собой вклады, обусловленные изменением (по сравнению с объемным кристаллитом) положений атомов в «отрелаксированной» двухмерной структуре.

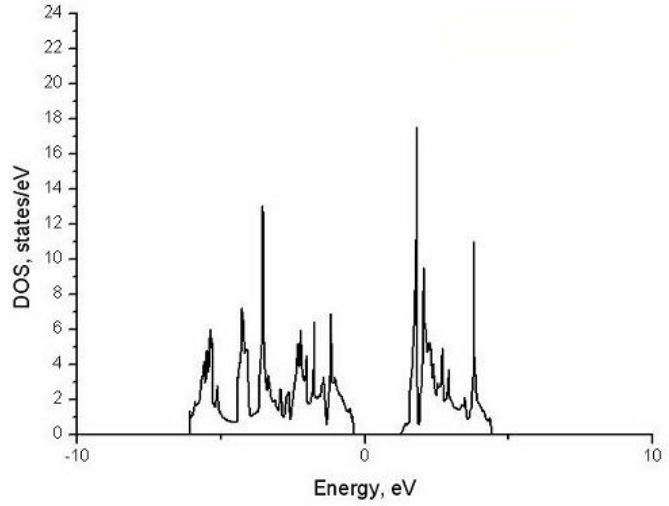
Зонные диаграммы для 2D-структур MoS_2, сформированных по трем исследованным кристаллографическим направлениям, представлены на рис. 3.

Анализ результатов, представленных на рис. 3, показывает, что зонные диаграммы 2D<010>- и 2D<100>-структур, а также для 3D-структуры одинаковы. Отмеченная особенность объясняется симметричным расположением атомов относительно граней ячейки и однотипностью связей в этих структурах. Отметим образование дополнительных разрешенных состояний для электронов в валентной зоне и в зоне проводимости, вызванное разрывом связей на поверхностях 2D<010>, <100>-структур, что приводит к уменьшению ширины запрещенной зоны и увеличению спектров обменного взаимодействия.

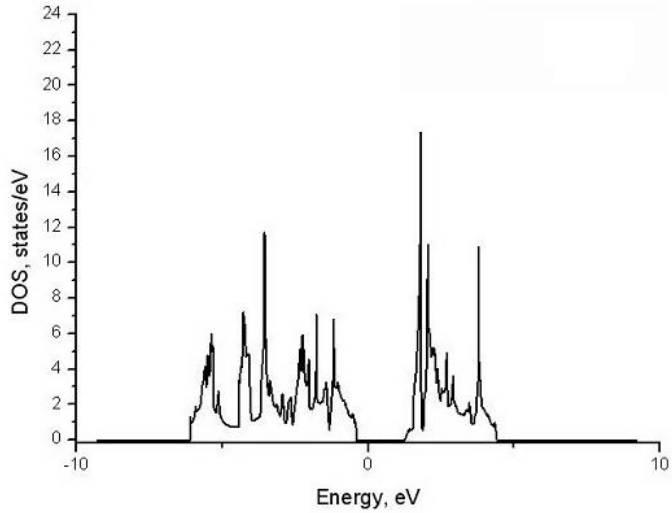
Для зонной диаграммы 2D<001>-структуры выявлен прямозонный переход, что, возможно, объясняется восстановлением разорванных связей на поверхности моделируемой структуры.



а)

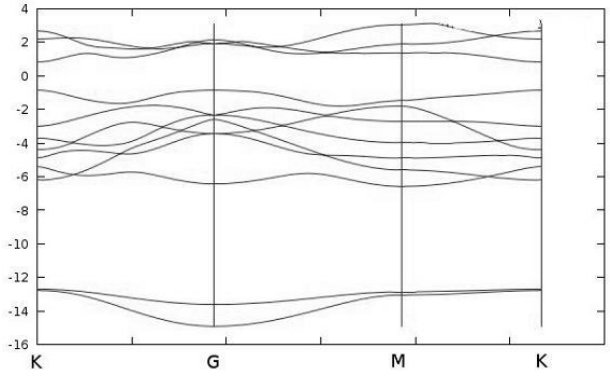


б)

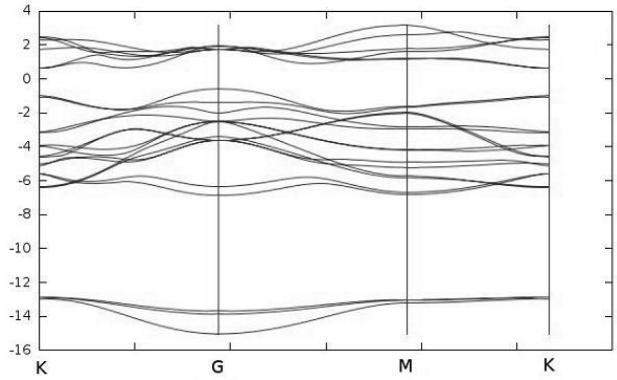


в)

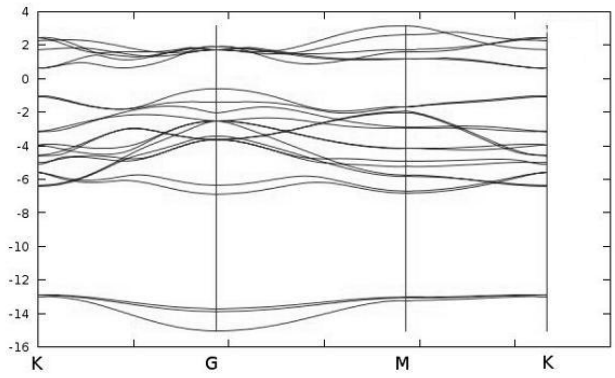
Рис. 2. Электронные плотности в 2D-структуре MoS_2 для различных кристаллографических направлений: *а)* <001>; *б)* <010>; *в)* <100>. Энергия 0 эВ соответствует уровню Ферми



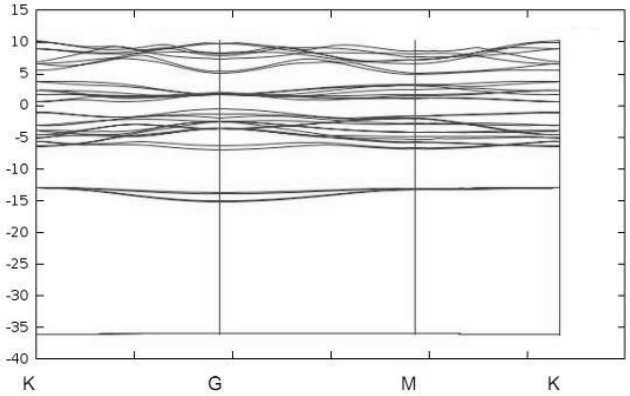
а)



б)



в)



г)

Рис. 3. Зонные диаграммы исследованных 2D-структур MoS_2: *а)* <001>; *б)* <010>; *в)* <100>; *г)* 3D-структуры MoS_2

Заключение

Посредством *ab initio* моделирования исследованы электронные свойства 3D- и 2D-структур молибденита, MoS_2, сформированных вдоль кристаллографических направлений <001>, <010> и <100>.

Результаты расчетов распределения электронной плотности и зонных диаграмм показали, что трехмерные и двухмерные <010>, <100>-структуры MoS_2 по указанным характеристикам идентичны. Рассчитанные значения ширины запрещенной зоны исследованных двухмерных структур MoS_2 по трем основным кристаллографическим направлениям оказались равными 1,71; 1,24 и 1,24 эВ для 2D<001>, <010>, <100> соответственно. В наибольшей степени электронные анизотропные свойства проявились в 2D<001>-структуре. Межзонный переход в этой структуре оказался прямозонным, что является важным свойством для использования 2D<001>-структуры MoS_2 в сенсорике, оптике и наноэлектронике. Спиновая поляризация носителей заряда в исследованных структурах не обнаружена. Проявление этого эффекта может ожидаться в 2D-структурах MoS_2 при введении в них комплексов точечных дефектов, как показано для графена [12].

Работа выполнялась при поддержке фондов государственных программ научных исследований Республики Беларусь (подпрограммы «Кристаллические и молекулярные структуры» ГПНИ «Функциональные и машиностроительные материалы, наноматериалы» и подпрограммы «Информатика и космос» ГПНИ «Научные основы и инструментальные средства информационных и космических технологий»).

Список литературы

1. Kresse, G. From ultrasoft pseudopotentials to the projector augmented-wave method / G. Kresse, D. Joubert // Physical Review. B. – 1999. – Vol. 54. – P. 1758–1775.

2. Kresse, G. Efficiency of ab initio total energy calculations for metals and semiconductors using a plane-wave set / G. Kresse, J. Furthmuller // Comput. Mat. Sci. – 1996. – Vol. 6. – P. 15–50.

3. Durinck, J. Influence of crystal chemistry on ideal plastic shear anisotropy in forsterite: first principle calculations / J. Durinck, A. Legris, P. Cordier // American Mineralogist. – 2005. – Vol. 90. – P. 1072–1077.

4. Nair, N.N. Glycine at the pyrite/water interface: an *ab initio* metadynamics study / N.N. Nair, E. Schreiner, D. Marx // Proceedings of NIC Symposium 2008. John von Neumann Institute for Computing. – Jülich, Germany, 2008. – Vol. 39. – P. 101–108.

5. Wang, L. *Ab initio* study of the surface properties and nanoscale effects of $LiMnPO_4$ / L. Wang, F. Zhou, G. Ceder // Electrochemical and Solid-State Letters. – 2008. – Vol. 11. – P. A94–A96.

6. Lei, Y. First principles study of the size effect of TiO_2 anatase nanoparticles in dye-sensitized solar cell / Y. Lei, H. Liu, W. Xiao // Modell. Simul. Mater. Sci. Eng. – 2010 – Vol. 18. – P. 025004–025011.

7. Double-gate strained-Ge heterostructure tunneling FET (TFET) with record high drive currents and 60mV/dec subthreshold slope / T. Krishnamohan [et al.] // Proceedings of Electron Devices Meeting (IEDM). IEEE International. – San Francisco, USA, 2008. – P. 1–3.

8. Conroy, M. Anisotropic constitutive relationships in energetic materials: PETN and HMX / M. Conroy, I.I. Oleynik, C.T. White // AIP Conf. Proc. – 2007. – Vol. 955. – P. 361–364.

9. Friak, M. *Ab initio* calculation of tensile strength in iron / M. Friak, M. Sob, V. Vitek // Phil. Mag. – 2003. – Vol. 83. – P. 3529–3537.

10. Tung, J.C. An *ab initio* study of the magnetic and electronic properties of Fe, Co, and Ni nanowires on Cu(001) surface / J.C. Tung, G.Y. Guo // Computer Physics Communications. – 2011. – Vol. 182(1). – P. 84–86.

11. The rise of graphene / S. Novoselov [et al.] // Nature Mater. – 2007. – Vol. 6. – P.183–191.

12. Nelayev, V. Magnetism of graphene with vacancy clusters / V. Nelayev, A. Mironchik // Mater. Phys. Mech. – 2010. – Vol. 9. – P.26–34.

13. Исаева, А.А. Создание новых материалов для микро- и наноэлектроники на основе наноблочных смешанных халькогенидов переходных (Ni, Fe)-непереходных металлов /

А.А. Исаева, А.Н. Кузнецов // Международный форум по нанотехнологиям Rusnanotech. – М., 2008. – С. 322–324.

14. Single-layer MoS_2 transistors / B. Radisavljevic [et al.] // Nature Nanotechnology. – 2011. – Vol. 6. – P. 147–150.

15. Lebegue, S. Electronic structure of two-dimensional crystals from ab initio theory / S. Lebegue, O. Ericsson // Physical Review. B. – 2009. – Vol. 79. – P. 115409–115412.

16. Fabrication of inorganic molybdenum disulfide fullerenes by arc in water / N. Sano [et al.] // Chemical Physics Letters. – 2003. – Vol. 368. – P. 331–337.

17. An alternative route to molybdenum disulfide nanotubes / W-K Hsu [et al.] // American Chemical Society. – 2000. – Vol. 122. – P. 10155–10158.

18. Lithium dynamics in molybdenum disulfide intercalation compounds studied by nuclear magnetic resonance / J.P. Donoso // Brazilian Journal of Physics. – 2006. – Vol. 36 (1A) – P. 55–60.

19. Schoenfeld, B. Anisotropic mean-square displacements (MSD) in single crystals of 2H- and 3R-MoS_2 / B. Schoenfeld, J.J. Huang, S.C. Moss // Acta Cryst. B. – 1983. – Vol. 39. – P. 404–407.

20. Y_2O_3 and MoS_2 electronic properties simulation / A. Gulay [et al.] // Proceedings of MEMSTECH'2011. – Polyana, Ukraine, 2011. – P. 111–113.

21. VASP the GUIDE [Электронный ресурс] / G. Kresse, J. Furthmuller // University of Vienna. – 2007. – P. 40. – Mode of access : http://wolf.ifj.edu.pl./workshop/work2008/tutorial/vasp.pdf. – Date of access : 12.01.2013.

Поступила 01.02.2013

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

O.A. Kozlova, V.V. Nelayev

AB INITIO MODELLING OF ELECTRONIC PROPERTIES OF TWO-DIMENSIONAL MOLYBDENUM DISULFIDE

The electronic properties of the three- and two-dimensional structures of molybdenum disulfide, formed along <001>, <010>and<100> crystallographic directions, are studied by means of *ab initio*, from first principles, modeling methods. Electron density and zone structure calculations are performed. It is shown that the zone diagrams of 2D-<010> and -<100> MoS_2 are identical. The zone structure of 2D-<001> MoS_2 distinguishes by the direct band gap transition and the absence of the additional energetic levels. The detected peculiarities confirm the possibility that MoS_2, as well as graphene, possess unique electronic and magnetic properties.

ИНФОРМАТИКА

2013 **июль-сентябрь** **№ 3**

ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ

УДК 004.33.054

С.В. Ярмолик, В.Н. Ярмолик

КВАЗИСЛУЧАЙНОЕ ТЕСТИРОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Анализируются причинно-следственные связи при возникновении неисправностей вычислительных систем. Даются определения понятий «неисправность», «ошибка» и «неисправное поведение вычислительных систем», показывается их общность для программной и аппаратной частей вычислительных систем. Рассматривается классификация обобщенных входных тестовых воздействий на три категории: точечные тестовые наборы, узкополосные тестовые наборы и блочные тестовые наборы. Приводится анализ методов тестирования вычислительных систем по методике черного ящика, показывается эффективность использования квазислучайного тестирования. Анализируются и предлагаются методы формирования квазислучайных тестовых воздействий.

Введение

В настоящее время актуальной является проблема тестирования современных вычислительных систем, таких как встроенные системы (Embedded System), системы на кристалле (System-on-a-Chip) и сети на кристалле (Net-on-a-Chip) [1, 2]. Характерной особенностью подобных вычислительных систем является тесное взаимодействие их аппаратных и программных средств, причем подавляющую часть аппаратных средств представляют собой запоминающие устройства различного уровня иерархии системы [2, 3]. Согласно последним прогнозным исследованиям менее чем через десять лет встроенная память будет занимать более 95 % площади кристалла вычислительных систем [3].

Архитектурные особенности современных вычислительных систем, многообразие физических дефектов их аппаратной части и ошибок в программном обеспечении, а также многочисленные подходы для тестирования таких систем предопределяют необходимость применения универсальных методов для совместного тестирования аппаратной и программной частей систем [2, 4]. В настоящее время практически отсутствуют методы совместного тестирования аппаратной и программной частей вычислительных систем, в большей части из-за отсутствия точных моделей их неисправного поведения и различной содержательной и терминологической их интерпретации [1, 2, 4, 5].

1. Основные определения и классификация

Большинство терминов и понятий в области тестирования и диагностики вычислительных систем имеют достаточно общий смысл и характеризуются отсутствием точности и однозначности. В основном это связано с разнообразием типов современных вычислительных систем, большим количеством различных методов их тестирования, различными уровнями абстракции их описания, а также с формулировкой различных целей и задач тестирования. Поэтому для последующего обсуждения очень важно однозначно определить следующие термины: физический дефект (physical defect), ошибка проектирования (mistake), неисправность (fault), ошибка системы (error) и неисправное поведение системы (failure).

Обычно на самом низком уровне абстракции используется термин «неисправность» как математическая модель, описывающая физические дефекты системы и ошибки при ее проектировании, изготовлении и использовании. Основными источниками неисправностей являются: ошибки спецификации (specification mistakes) из-за неправильных или неправильно интерпретированных проектных требований к системе и/или ее спецификации; ошибки проектирования (implementation mistakes) в основном из-за ошибок кодирования (bags); физические дефекты аппаратных компонентов системы (component physical defects); внешние факторы (external

factors), такие как условия окружающей среды (температура, электромагнитные и радиационные излучения и др.) или человеческий фактор (ошибки оператора) [1, 4–6]. Тогда неисправность как математическую модель неисправного состояния вычислительной системы определим следующим образом:

Определение 1. Неисправностью вычислительной системы является такая ее функциональность, которая может приводить к новым выходным значениям (наблюдаемому поведению) и/или к новому состоянию системы, не соответствующим спецификации системы и требованиям, предъявляемым к ней.

Согласно определению 1 неисправности вычислительной системы могут приводить к ее неверным наблюдаемым выходным значениям и ошибочным внутренним состояниям или только к ошибочным внутренним состояниям. Отметим, что не для всего множества входных значений наличие неисправности приводит к ошибочным состояниям системы и ошибочным выходным значениям.

Определение 2. Ошибкой вычислительной системы из-за наличия в ней неисправности является формирование для некоторого множества входных значений системы одного или более неверных результатов, отличающихся от ожидаемых значений.

Ошибки вычислительной системы, по сути, являются результатом активизации ее неисправностей. Они могут привести к неисправному поведению системы (системному сбою), которое, в свою очередь, может быть наблюдаемым признаком неисправного поведения (состояния) вычислительной системы.

Определение 3. Неисправным поведением (сбоем) вычислительной системы называется формирование наблюдаемых выходных значений системы, отличных от ожидаемых, для некоторого множества входных значений.

Неисправное поведение вычислительной системы возникает в результате решения задачи транспортировки ошибочного значения (ошибки) системы на ее наблюдаемые выходы.

Приведенные определения в равной степени применимы как к программному, так и аппаратному обеспечению вычислительных систем. Обоснованность и применимость этих определений проиллюстрируем примером из программного обеспечения [6]. Рассмотрим программу, которая вычисляет остаток от деления на три квадрата (*data***2) входного значения *data* (рис. 1).

```
begin
        read (data);
                data: = 2*data;   (← неисправность)
                data: = data mod 3;
        write (data)
end.
```

Рис. 1. Пример программы, содержащей неисправность

В связи с наличием неисправности из-за ошибки при кодировании в третьей строке кода программы эта программа фактически вычисляет остаток от деления на три удвоенного значения *data*. При рассмотрении программы, представленной выше для исходного значения *data* = 3, выходной результат оператора *data*: = 2**data*, содержащего неисправность, составляет 6, тогда как ожидаемое значение, при отсутствии неисправности, должно быть 9. Таким образом, возникает вычислительная ошибка. В то же время для *data* = 2 ошибочного значения не возникает, так как неисправность не активизируется. Кроме того, для обоих значений 3 и 2 входных данных *data* программа вычисляет ожидаемо правильные результаты, а именно 0 и 1. Это означает, что значения *data*, равные 3 и 2, не вызывают неисправного состояния (сбоя программы), так как значение *data* = 2 даже не вызывает ошибку, а значение *data* = 3 инициирует ошибочный промежуточный результат – 6 вместо 9. Однако в обоих случаях формируется правильное выходное значение программы, равное 0. В то же время для *data* = 4 программа, содержащая неисправность в третьем операторе, формирует ошибочное значение (вычислительную ошибку).

Действительно, в этом случае выходное значение равняется 2 вместо ожидаемой величины 1, что свидетельствует о неисправном состоянии программы и приводит к ее сбою.

В следующем примере рассмотрим аппаратную составляющую вычислительной системы, представленную оперативным запоминающим устройством (ОЗУ) [7]. Предположим, что ОЗУ содержит четырехразрядные запоминающие ячейки, которые используются для хранения данных, состоящих из четырех бит и представляющих собой шестнадцатеричную цифру. В одной из ячеек ОЗУ возникла константная неисправность ≡0, которая приводит к тому, что, например, во втором разряде ячейки постоянно находится значение 0 [7]. Что касается остальных разрядов ячейки, то их содержимое может быть 0 или 1. Рассмотрим случай шестнадцатеричных данных, равных 4, 3 и 2, которые должны быть записаны в ОЗУ и затем считаны. Для значения данных 4 наличие неисправности во втором бите ячейки ОЗУ не приводит к ошибке, несмотря на наличие неисправности ≡0, так как $4_{(16)} = 0100_{(2)}$. Оба значения данных 3 и 2 в двоичном представлении (0011, 0010) содержат 1 во втором бите, что является причиной возникновения ошибки выборки после выполнения операции чтения. Это, в свою очередь, может привести к неисправному состоянию вычислительной системы в целом.

Из рис. 2 видно, что первопричиной неисправностей являются: ошибки спецификации, ошибки реализации, физические дефекты аппаратных компонентов системы и внешние факторы [4–6]. Наличие неисправностей может вызывать ошибки как программных, так и аппаратных средств вычислительных систем. Ошибки возникают в результате активизации неисправностей вычислительной системы, как это было показано в приведенных выше примерах. И, наконец, активизированная ошибка вычислительной системы может привести к наблюдаемому неисправному поведению вычислительной системы.

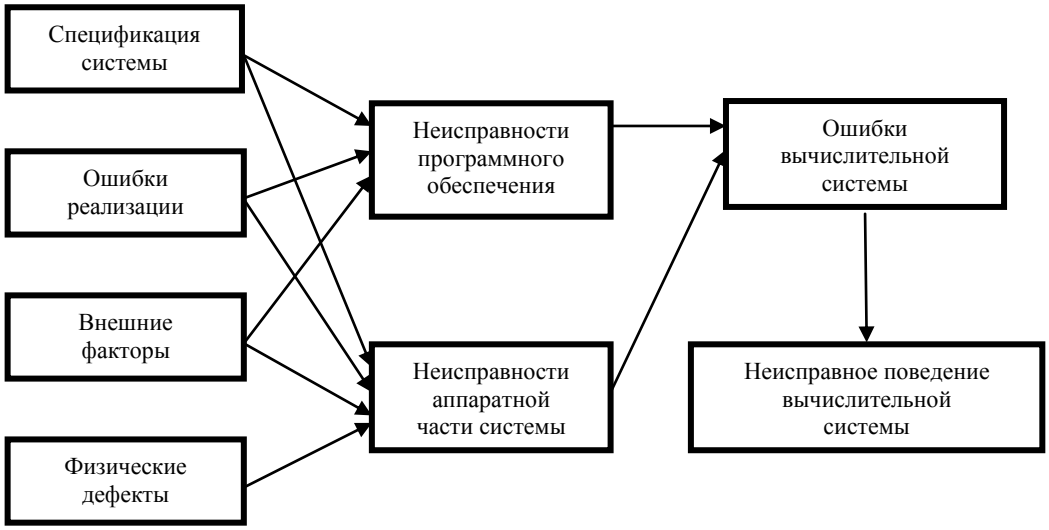


Рис. 2. Причинно-следственная связь между неисправностями, ошибками и неисправным поведением вычислительной системы

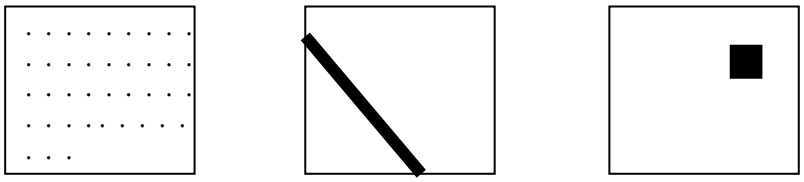
Существующие методологии построения тестовых последовательностей решают задачу нахождения такого подмножества входных тестовых воздействий (failure patterns), которые приводили бы к наблюдаемому неисправному поведению вычислительной системы в случае возникновения в ней неисправностей на всех стадиях жизненного цикла системы [4–6].

2. Обобщенные входные тестовые воздействия

Для того чтобы оценить эффективность различных методов тестирования вычислительных систем, рассмотрим множества их входных тестовых воздействий, которые приводят к наблюдаемому неисправному поведению систем. Очевидно, что данные множества являются уникальными для каждого из объектов тестирования и зависят от многих факторов (см. рис. 2),

а также от его архитектуры и функциональности. Однако структура входных тестовых воздействий характеризуется рядом закономерностей, отмеченных в последних публикациях [8–14].

Большой объем эмпирических и экспериментальных исследований различного рода программного обеспечения показал обобщенность (структурированность) входных воздействий (наборов), которые инициируют неисправное поведение систем, т. е. являются тестовыми воздействиями [8–11]. В основополагающей работе [10] была представлена классификация обобщенных входных тестовых воздействий на три категории: точечные тестовые наборы (point patterns), узкополосные тестовые наборы (strip patterns) и блочные тестовые наборы (block patterns) [10]. Для иллюстрации данной классификации пространство входных наборов рассматривается как двухмерное. Тогда три вида входных тестовых наборов имеют простую геометрическую интерпретацию (рис. 3) [8–11].



а) *б)* *в)*

Рис. 3. Типовые входные тестовые воздействия

На рис. 3 области, выделенные черным цветом, соответствуют входным воздействиям, приводящим к наблюдаемому неисправному поведению программного обеспечения, а фигуры квадратов соответствуют областям входных воздействий [8, 10, 11]. Точечные тестовые наборы (рис. 3, *а*) определяют уникальные входные наборы, распределенные по всему пространству входных наборов, причем их распределение в большинстве случаев имеет регулярную структуру. Узкополосные тестовые наборы (рис. 3, *б*) характеризуются непрерывными множествами входных воздействий имеющих вид узких полос. Тестовые наборы блочного типа (рис. 3, *в*) представляют собой одну либо несколько непрерывных областей входных воздействий, приводящих к наблюдаемому неисправному поведению систем. Как правило, такие типовые входные воздействия выбираются в качестве моделей входных тестовых воздействий, поскольку многочисленные эмпирические исследования подтверждают целесообразность их использования в качестве приближения к реальным входным тестовым воздействиям [8–11]. Несмотря на то что эти модели не являются реальными, их применение, а также применение их комбинаций оказывается весьма эффективным при построении тестовых последовательностей для реальных вычислительных систем [8–11].

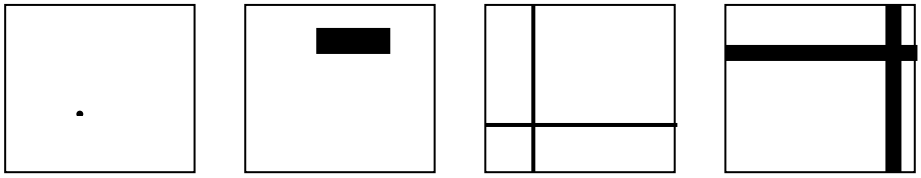
Значительная часть аппаратной составляющей вычислительных систем, особенно систем на кристалле и встроенных систем, включает в себя встроенную память, которая в соответствии с прогнозами будет занимать доминирующую часть на кристаллах вычислительных систем. Поэтому эффективное тестирование памяти и совершенная методология анализа неисправностей запоминающих устройств будет способствовать повышению надежности и выхода годных встроенных систем и систем на кристалле, особенно при ускоренных процессах их разработки и применении передовых технологий изготовления [1, 7].

Причинно-следственный подход может быть использован для прогнозирования неисправного поведения ОЗУ в случае возникновения физических дефектов. Неисправное поведение памяти представляется растровым изображением физического топологического представления результатов тестирования, показывающим расположение неисправных запоминающих ячеек. Чаще всего возникают одиночные неисправности запоминающих ячеек памяти, неисправности групп запоминающих ячеек, неисправности дешифратора адреса ячейки памяти, неисправности шин данных и другие неисправности запоминающих устройств [7, 12–14].

Согласно устоявшейся классификации неисправностей запоминающих устройств различают два их множества: простые, в которых участвуют одна либо две ячейки памяти, и сложные, включающие в себя множество ячеек [7, 14]. К неисправностям, затрагивающим одну ячейку, относятся константные неисправности и переходные неисправности, а к неисправно-

стям, в которых участвуют две ячейки, – неисправности взаимного влияния [7, 14]. Обобщающей моделью входных тестовых воздействий (адресов ячеек памяти) является точечная модель (рис. 4, *а*) либо, для случая многократных неисправностей данных типов, блоковая (рис. 4, *б*) [7, 13, 14]. Кодочувствительные неисправности, неисправности дешифратора адреса и неисправности типа «разрушающая операция чтения» и «ошибочная запись» покрываются моделью узкополосных входных воздействий (рис. 4, *в*) [7, 13, 14]. В случае комбинации физических дефектов памяти, а также их многократных разновидностей входные тестовые воздействия описываются комбинированной моделью (рис. 4, *г*), объединяющей блоковый и узкополосный типы [13].

Таким образом, все множество моделей неисправного поведения запоминающих устройств описывается четырьмя обобщенными моделями входных тестовых воздействий, представленных на рис. 4 [13].



а) *б)* *в)* *г)*

Рис. 4. Типовые входные тестовые воздействия запоминающих устройств

По сравнению с входными тестовыми наборами для программного обеспечения тестовые воздействия запоминающих устройств имеют реальную двухмерную структуру в силу физического двухмерного представления матрицы запоминающих ячеек [14]. В этом случае входными тестовыми данными являются адреса запоминающих ячеек, состоящие из адресов по горизонтальной и вертикальной осям матрицы ячеек памяти.

Как видно из приведенного анализа, типовые входные тестовые воздействия для программного обеспечения и для большей части аппаратных средств вычислительных систем описываются аналогичными моделями, что, очевидно, позволяет использовать единый подход для тестирования вычислительных систем в целом. Следующим важным выводом анализа типовых входных тестовых воздействий является их разнородность по структуре (см. рис. 3 и 4), что затрудняет выбор наиболее эффективного метода их тестирования.

3. Анализ методов тестирования вычислительных систем

В общем случае тестирование вычислительных систем направлено на повышение их надежности и заключается в выявлении максимально возможного количества неисправностей систем за приемлемый (реальный) промежуток времени. Исчерпывающее тестирование (exhaustive testing) характеризуется максимально возможной эффективностью, так как при его реализации проверяется объект тестирования на всевозможных входных воздействиях для всех возможных состояний объекта тестирования [15]. В случае двоичных входных воздействий из N бит необходимо сгенерировать 2^N двоичных комбинаций. Всевозможные двоичные комбинации генерируются для каждого состояния, которое в случае цифровых устройств и систем определяется количеством M запоминающих элементов, которые используются для построения регистров, счетчиков и других последовательностных устройств систем. Таким образом, количество состояний цифрового устройства или системы определяется величиной 2^M в силу того, что каждый запоминающий элемент может находиться в одном из двух состояний, а именно в состоянии 0 или состоянии 1. Сложность исчерпывающего теста (количество тестовых наборов) определяется астрономической величиной 2^{N+M}. Экспоненциальный рост длины исчерпывающего теста существенно ограничивает область его применения только для случаев простейших вычислительных устройств с небольшими значениями N и M [15, 16]. Локально исчерпывающее (locally exhaustive) [16] или псевдоисчерпывающее (pseudo exhaustive) [17, 18] тестирование вычислительных систем позволяет избежать ограничений на число входов тестируемого устройства или системы. Эти подходы являются реальной альтернативой для исчерпы-

вающего тестирования и позволяют существенно сократить число тестовых векторов, однако требуют большого объема исследований при построении подобных тестов [16, 18], а также детального описания вычислительных систем, что не всегда возможно.

Для эффективной аппроксимации исчерпывающего и псевдоисчерпывающего тестирования широко используется случайное (вероятностное) тестирование (random testing) [19]. В данном случае под аппроксимацией понимается формирование заведомо не исчерпывающих и не псевдоисчерпывающих тестов, а тестов, которые являются некоторым их приближением. Данный метод тестирования широко применяется в рамках модели черного ящика (black box), когда вычислительная система описывается только на уровне выполняемых ею функций, а ее внутренняя структура и конкретная реализация как программной, так и аппаратной частей не учитываются. Согласно определению случайного тестирования очередное значение тестового набора выбирается случайным образом, независимо от значений предыдущих наборов теста [19, 20]. Случайное тестирование является неэффективным, когда плотность оставшихся (необнаруживаемых) неисправностей оказывается низкой [20]. Случайное тестирование не использует информацию, которая доступна при реализации метода черного ящика, а именно информацию о предыдущих тестовых наборах, что увеличивает длину тестовой последовательности и уменьшает полноту покрытия неисправностей вычислительных систем [11, 18, 21]. Использование информации о предыдущих тестовых наборах явилось основой создания так называемого антислучайного тестирования (antirandom testing) [18, 22, 23].

Предпосылкой антислучайного тестирования является то, что для достижения более высокой полноты покрытия неисправностей вычислительных систем очередной тестовый набор выбирается из случайных тестовых наборов таким образом, чтобы он был максимально отличным от ранее сгенерированных наборов. Для этого используются различные метрики отличия, такие как расстояние Хемминга и декартово расстояние [22, 23]. При антислучайном тестировании очередной тестовый набор выбирается из множества наборов с максимальным расстоянием от ранее сгенерированных наборов, что позволяет достичь большей эффективности по сравнению со случайным тестированием [22, 23]. К сожалению, основным недостатком антислучайного тестирования является его вычислительная сложность, вызванная необходимостью вычисления метрик расстояния для каждого кандидата в тесты [22]. Даже для улучшенных вариантов антислучайного тестирования вычислительная сложность их реализации является существенным ограничением при практическом применении для реальных размерностей тестовых наборов [23]. В качестве более эффективной метрики для построения антислучайного теста с ограниченным числом наборов используется максимально минимальное расстояние Хэмминга (maximal minimal hamming distance), которое применяется для построения антислучайного теста [18, 25, 26]. К сожалению, проблема вычисления расстояния является ключевой проблемой при генерировании антислучайных тестов, что ограничивает области их применения [11, 18, 22–26].

Все перечисленные методы тестирования в своей основе используют модель черного ящика и направлены на формирование входных тестовых наборов как подмножества входных воздействий, формируемых в большинстве случаев с использованием случайных воздействий [15–26]. Случайное тестирование и все многообразие его модификаций можно описать в терминах численных методов Монте-Карло, основанных на получении большого числа реализаций стохастического (случайного) процесса. Основными недостатками методов случайного тестирования является их невысокая полнота покрытия неисправностей и большая длина тестовых последовательностей. Подобными недостатками характеризуется и метод Монте-Карло, для которого характерны значительная вычислительная погрешность и заметная временная сложность. Поэтому в качестве альтернативного решения для тестирования вычислительных систем в настоящей статье предлагается использовать идеи квазислучайного тестирования, основанного на применении квазислучайных последовательностей как тестовых последовательностей, эффективно покрывающих пространство входных воздействий систем [27–31]. Аналогично, как и в методе квази-Монте-Карло (КМК), квазислучайное тестирование, очевидно, позволит достичь большей полноты покрытия при меньших длинах тестовых последовательностей. Основой реализации квазислучайного тестирования являются квазислучайные последовательности. В следующем разделе статьи проводится анализ подобных последовательностей, обосновыва-

ется использование последовательностей Соболя и предлагаются авторские модификации для генерирования большого числа их реализаций.

4. Квазислучайные последовательности

Последовательность неслучайных чисел называется псевдослучайной последовательностью чисел, если она обладает всеми свойствами случайной последовательности [32]. Последовательность неслучайных чисел называется квазислучайной, если ее можно использовать в реализации алгоритмов Монте-Карло вместо случайной последовательности [33]. Именно такие последовательности используются на практике для различных задач метода КМК, что позволяет достичь меньших вычислительных погрешностей и более быстрой сходимости [27, 33, 34]. Это достигается не столько свойством независимости, характерным для псевдослучайных последовательностей, сколько свойством равномерности. В русскоязычной литературе такие последовательности называют согласно Соболю [27, 33, 34] ЛП$_\tau$-последовательностями, что интерпретируется как «любой последовательный участок хорошо распределен» (более равномерно по сравнению с псевдослучайными последовательностями), в англоязычной литературе – последовательностями с малым дискрепансом (low-discrepancy sequence), а их разновидности – по именам авторов, выделяя последовательности Соболя [27–31].

Для визуальной демонстрации большей равномерности квазислучайной последовательности точек по сравнению с псевдослучайной последовательностью обычно рассматривается двухмерное пространство в виде единичного квадрата. Это пространство равномерно делится на подквадраты. Так, например, квадрат на рис. 5 равномерно разбит на 64 подквадрата и на них нанесены 64 квазислучайные точки ЛП$_\tau$-последовательности (рис. 5, *а*) и 64 псевдослучайные точки (рис. 5, *б*). Из приведенных рисунков видно, что в каждый подквадрат попало ровно по одной квазислучайной точке, в то время как для псевдослучайных точек равномерное заполнение подквадратов не выполняется.

а) *б)*

Рис. 5. Множества точек в двухмерном пространстве: *а)* квазислучайных; *б)* псевдослучайных

Следует отметить, что данное свойство ЛП$_\tau$-последовательностей выполняется для пространств произвольной размерности, а не только для двухмерного пространства [33, 34].

4.1. Последовательности Корпута

Последовательности Корпута являются простейшим примером квазислучайных последовательностей [27, 28]. Для произвольного целого $n \in \{1, 2, 3, \ldots, N\}$ значение элемента x_n последовательности Корпута может быть получено для любого базиса p представления числа n, где p – простое целое число. Первоначально целое число n представляется в базисе p как

$$n = \sum_{i=0}^{I} \alpha_i(n)\, p^i,$$

где $\alpha_i(n) \in \{0, 1, 2, \ldots, p-1\}$ является значением i-й цифры p-ичного представления числа n, а I – наименьшим целым значением i, для которого $\alpha_i(n) \neq 0$, а для всех $i > I$ выполняется равенство $\alpha_i(n) = 0$. Значение I вычисляется как

$$I = \lfloor \log_p n \rfloor, \quad n = \overline{1, N}.$$

Затем для получения значения x_n цифры $\alpha_i(n)$ числа n транспонируются относительно запятой, разделяющей целую и дробную части p-ичного числа, таким образом, что первой после запятой оказывается младшая $\alpha_0(n)$ цифра p-ичного представления числа n. Следующей цифрой будет $\alpha_1(n)$ и т. д. Аналитически процедура транспонирования, в результате которой получается значение x_n, описывается формулой

$$x_n = \sum_{i=0}^{I} \frac{\alpha_i(n)}{p^{i+1}}.$$

В качестве примера рассмотрим целое число $n = 11$, которое представим в базисе $p = 3$ как $11_{(10)} = 1\times3^2 + 0\times3^1 + 2\times3^0 = 102_{(3)}$ с $\alpha_0(11) = 2$, $\alpha_1(11) = 0$ и $\alpha_2(11) = 1$. Следует отметить, что $I = \lfloor \log_3 11 \rfloor = 2$ и соответственно $\alpha_i(n)=0$ для $i>2$. Выполнив процедуру транспонирования, получим значение элемента x_{11} последовательности Корпута $x_{11} = 2/3^1 + 0/3^2 + 1/3^3 = 19/27$, которое принадлежит интервалу [0, 1]. Квазислучайной последовательностью Корпута с основанием $p = 2$ для $N = 7$ является последовательность x_n, состоящая из следующих элементов: $x_1 = 1/2_{(10)}=0{,}1_{(2)}$; $x_2 = 1/4_{(10)}=0{,}01_{(2)}$; $x_3 = 3/4_{(10)}=0{,}11_{(2)}$; $x_4 = 1/8_{(10)}=0{,}001_{(2)}$; $x_5 = 5/8_{(10)}=0{,}101_{(2)}$; $x_6 = 3/8_{(10)}=0{,}011_{(2)}$ и $x_4 = 7/8_{(10)}=0{,}111_{(2)}$.

Важным свойством последовательности Корпута является то, что после генерирования каждого множества из 2^k–1 элементов для приведенного примера (k = 1, 2 и 3) последовательность максимально равномерно распределена, т. е. самый длинный интервал, который не содержит элементов из последовательности Корпута, является минимальным.

Использование произвольного базиса p предопределяет формирование иррациональных дробных чисел, что затрудняет применение подобных последовательностей для целей тестирования вычислительных систем при $p \neq 2$.

4.2. Последовательности Халтона

Последовательности Халтона представляются последовательностью точек, задаваемых их координатами, в s-мерном ($s \geq 1$) пространстве [29]. Они являются наиболее известными многомерными квазислучайными последовательностями. Данные последовательности служат основой для построения других разновидностей квазислучайных последовательностей [29]. Первая координата точек Халтона задается последовательностью Корпута с основанием $p = 2$, вторая координата определяется последовательностью Корпута с основанием 3. В общем случае координаты точек Халтона задаются последовательностями Корпута с использованием простых чисел p как оснований систем счисления [29].

В качестве примера рассмотрим последовательность точек Халтона в двухмерном пространстве (s=2) для $p = 2$ и 3. Для построения данной последовательности необходимо использовать две последовательности Корпута для p = 2 и 3. В десятичной системе счисления эти последовательности иррациональных чисел принимают следующий вид: 1/2, 1/4, 3/4, 1/8, 5/8, 3/8, 7/8, ... и 1/3, 2/3, 1/9, 4/9, 7/9, 2/9, 5/9, ... Тогда последовательность Халтона будет представляться последовательностью точек (1/2, 1/3), (1/4, 2/3), (3/4, 1/9), (1/8, 4/9), (5/8, 7/9), (3/8, 2/9), (7/8, 5/9), ... двухмерного пространства. Пример последовательности точек Халтона x_n для трехмерного случая приведен в табл. 1.

Таблица 1

Последовательность Халтона

x_n	p=2	p=3	p=5
n=1	1/2	1/3	1/5
n=2	1/4	2/3	2/5
n=3	3/4	1/9	3/5
n=4	1/8	4/9	4/5
n=5	5/8	7/9	1/25
n=6	3/8	2/9	6/25
n=7	7/8	5/9	11/25
n=8	1/16	8/9	16/25

Отмечается, что последовательности Халтона характеризуются низким качеством для размерностей *s*, больших чем 14 [29]. На практике в силу корреляционных зависимостей последовательности Халтона используются для значений *s*, не больших чем 6 [29].

Так же, как и в случае последовательностей Корпута, применение последовательностей Халтона для целей технической диагностики практически ограничивается только одномерными последовательностями, когда $s = 1$ и соответственно $p = 2$.

4.3. Последовательности Соболя

Последовательности Соболя используют двоичную систему счисления для формирования координат точек в *s*-мерном пространстве и в силу данного обстоятельства являются широко востребованными для современных приложений [27, 29–31]. Последовательность точек Соболя в *s*-мерном пространстве строится на основании одномерной последовательности Корпута, когда первая координата точек формируется последовательностью Корпута для $p = 2$, а остальные – путем процедуры перестановок [27]. При этом перестановки зависят от так называемых *направляющих чисел* (*direction numbers*) v_i^j, применяемых для всех измерений $j = \overline{1, s}$ *s*-мерного пространства. Направляющие числа $\mathrm{v}_i^j = \frac{m_i^j}{2^i}, i = \overline{1, w}$, образуют последовательность дробных двоичных чисел с *w* двоичными битами после запятой, где m_i^j представляет собой целое нечетное число, удовлетворяющее неравенству $0 < m_i^j < 2^i$. Для описания конкретной последовательности Соболя необходимо задать направляющие числа v_i^j для всех измерений *s*-мерного пространства.

Значение *n*-го элемента (точки) x_n последовательности Соболя определяется его координатами x_n^j для всех измерений *s*, которые вычисляются согласно выражению

$$x_n^j = \alpha_0(n)\mathrm{v}_1^j \oplus \alpha_1(n)\mathrm{v}_2^j \oplus \ldots \oplus \alpha_{w-1}(n)\mathrm{v}_w^j, \quad j = \overline{1, s}.$$

Здесь $\alpha_i(n) \in \{0, 1\}$, $i = \overline{0, w-1}$, являются значениями цифр двоичного представления $\sum_{i=0}^{\lfloor \log_2 n \rfloor} \alpha_i(n) 2^i$ числа *n*, а символ $\oplus$ означает операцию поразрядного сложения по модулю два. Например, если $m_1^j = 1$, $m_2^j = 3$, $m_3^j = 7$ (w=3) и соответственно v_1^j =0,100, v_2^j= 0,110 и v_3^j= 0,111, можно получить значение произвольного элемента последовательности Соболя для $n \in \{1, 2, \ldots, 2^w - 1\} = \{1, 2, \ldots, 7\}$. Предположим, что $n = 7_{(10)}$, которое в двоичном представлении записывается в виде $111_{(2)}$, тогда значение *j*-й координаты x_7^j седьмого элемента x_7 последовательности Соболя вычисляется как

$$x_7^j = \alpha_0(7)\mathrm{v}_1^j \oplus \alpha_1(7)\mathrm{v}_2^j \oplus \alpha_2(7)\mathrm{v}_3^j = 0{,}100 \oplus 0{,}110 \oplus 0{,}111 = 0{,}101.$$

Все элементы последовательности Соболя длиной $2^w - 1 = 2^3 - 1 = 7$ приведены в табл. 2.

Таблица 2

Одномерная классическая последовательность Соболя

$n_{(10)}$	$n_{(2)} = \alpha_2(n)\alpha_1(n)\alpha_0(n)$	x_n	$2^w \times x_n$
1	0 0 1	v_1	1 0 0
2	0 1 0	v_2	1 1 0
3	0 1 1	$\mathrm{v}_1 \oplus \mathrm{v}_2$	0 1 0
4	1 0 0	v_3	1 1 1
5	1 0 1	$\mathrm{v}_1 \oplus \mathrm{v}_3$	0 1 1
6	1 1 0	$\mathrm{v}_2 \oplus \mathrm{v}_3$	0 0 1
7	1 1 1	$\mathrm{v}_1 \oplus \mathrm{v}_2 \oplus \mathrm{v}_3$	1 0 1

Здесь для одномерного случая $v_i^1 = v_i$, а $x_n^1 = x_n$.

Последовательности Соболя, основанные на использовании двоичной системы счисления ($p = 2$), являются широко востребованной разновидностью квазислучайных последовательностей чисел в основном из-за удобства реализации на ЭВМ алгоритмов их генерирования. Основными недостатками классической последовательности Соболя являются заметная вычислительная сложность, зависящая от значения номера элемента (количества операций сложения по модулю два), а также ограниченное множество последовательностей, определяемое выбором направляющих чисел [29–31, 33, 34].

5. Модифицированные последовательности Соболя

Из табл. 2 видно, что значение координаты n-го элемента x_n последовательности Соболя вычисляется как поразрядная сумма по модулю два до $w=\lfloor \log_2 n \rfloor$ операндов в зависимости от количества ненулевых компонентов двоичного представления $\alpha_{w-1}(n)\alpha_{w-2}(n)\ldots\alpha_1(n)\alpha_0(n)$ величины n. Количество операндов может быть снижено до одного при использовании кода Грея [35]. Известно, что двоичное число $n+1$, закодированное в коде Грея, отличается от числа n только в одном бите. Представление числа n в коде Грея может быть получено согласно известному соотношению $n_g=g_{w-1}(n)g_{w-2}(n)\ldots g_1(n)g_0(n)=\alpha_{w-1}(n)\alpha_{w-2}(n)\ldots\alpha_1(n)\alpha_0(n)\oplus 0\alpha_{w-1}(n)\alpha_{w-2}(n)\ \ldots\ \alpha_1(n)$ [35]. Здесь индекс g числа n_g означает его представление в коде Грея.

В силу того что $(n+1)_g$ отличается от n_g только в одном бите, значение $x_{(n+1)g}^j$ будет отличаться от величины x_{ng}^j только значением одного направляющего числа v_i^j. Тогда значение j-й координаты $x_{(n+1)g}^j$ элемента $x_{(n+1)g}$ последовательности Соболя будет определяться как

$$x_{(n+1)g}^j = x_{ng}^j \oplus v_i^j . \qquad (1)$$

Соотношение (1) является описанием экономичного способа Антонова – Салеева для формирования последовательностей Соболя, приводимого во многих источниках, в том числе и в [28]. Процедура формирования последовательности Соболя для одномерного случая в соответствии с (1) представлена в табл. 3. Здесь рассмотрен случай последовательности, сгенерированной при условиях, которые аналогичны последовательности, приведенной в табл. 2.

Таблица 3

Одномерная последовательность Соболя, сгенерированная по способу Антонова – Салеева

n	n_g	v_i	$2^w \times x_{ng}$
0 0 1	001⊕000=001	v_1	100
0 1 0	010⊕001=011	v_2	010
0 1 1	011⊕001=010	v_1	110
1 0 0	100⊕010=110	v_3	001
1 0 1	101⊕010=111	v_1	101
1 1 0	110⊕011=101	v_2	011
1 1 1	111⊕011=100	v_1	111

Значение индекса направляющего числа v_i^j в выражении (1) зависит от так называемой последовательности переключений T_q отраженного кода Грея [35]. Для классического отраженного кода Грея переключательная последовательность задается рекурсивной процедурой следующим образом. Первоначально задается $T_1 = 1$ и, если $q > 1$, $T_q = T_{q-1}, q, T_{q-1}$. Например, для $q = 4$ получим $T_4 = 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1$.

Применение соотношения (1) позволяет существенно снизить вычислительную сложность генерирования последовательностей Соболя, что и предопределило их широкое применение на практике [28, 30, 31].

Вторым недостатком последовательностей Соболя является ограниченность их количества, которое определяется наборами направляющих чисел. С целью расширения множества последовательностей Соболя введем формализацию представления модифицированных направляющих чисел в виде нижней треугольной матрицы (унитреугольной матрицы) с единичной главной диагональю.

Первоначально отметим, что в силу ранее приведенных ограничений направляющие числа для всех измерений $v_i=0,\beta_{-1}(i)\beta_{-2}(i)\ldots\beta_{-w}(i)$ во всех случаях имеют определенные значения $\beta_{-i}(i)=1$ и $\beta_{-j}(i)=0$ для $j>i$, так же, как и произвольные значения $\beta_{-j}(i)\in\{0,1\}$ для $j<i$. Это означает, что для всех возможных последовательностей Соболя и всех их координат $v_1 = 0{,}100\ldots0$, а соответственно $2^w\times v_1=100\ldots00$ в силу того, что для m_1 существует только одно безальтернативное значение $m_1=1<2^1$. Так как m_2 есть нечетное целое, меньшее чем 2^2, оно может принимать два значения: 1 или 3. Соответственно $2^w\times v_2=\beta_{-1}(2)10\ldots00$, где $\beta_{-1}(2)=0$ для $m_2=1$ и $\beta_{-1}(2)=1$ для $m_2=3$. Для m_3 имеем $2^w\times v_3=\beta_{-1}(3)\beta_{-2}(3)10\ldots00$ и т. д.

Далее рассмотрим случай одномерных последовательностей Соболя и обозначим значения $2^w\times v_i$ новой переменной μ_i, которую будем рассматривать как значения модифицированных направляющих чисел. Отметим, что результаты, полученные для одномерного случая, легко обобщаются на многомерные последовательности Соболя.

Числа μ_i можно представить в виде нижней треугольной матрицы (унитреугольной матрицы) с единичной главной диагональю (табл. 4).

Таблица 4

Значения модифицированных направляющих чисел

μ_i	$\beta_{-1}(i)$	$\beta_{-2}(i)$	$\beta_{-3}(i)$	…	$\beta_{-w+1}(i)$	$\beta_{-w}(i)$
μ_1	1	0	0	…	0	0
μ_2	$\beta_{-1}(2)$	1	0	…	0	0
μ_3	$\beta_{-1}(3)$	$\beta_{-2}(3)$	1	…	0	0
…	…	…	…	…	…	…
μ_{w-1}	$\beta_{-1}(w-1)$	$\beta_{-2}(w-1)$	$\beta_{-3}(w-1)$	…	1	0
μ_w	$\beta_{-1}(w)$	$\beta_{-2}(w)$	$\beta_{-3}(w)$	…	$\beta_{-w+1}(w)$	1

Согласно процедуре генерирования $x_{(n+1)g} = x_{ng}\oplus\mu_i$ последовательности Соболя (1) для получения очередного значения используется только одно направляющее число. Индекс i направляющего числа μ_i определяется последовательностью переключений отраженного кода Грея, в которой на каждой второй позиции используется индекс 1, на каждой четвертой – индекс 2, на каждой восьмой – 3 и т. д. Это следует из определения переключательной последовательности и видно из ранее приведенного примера для T_4 [35]. Таким образом, каждое второе значение последовательности Соболя получается с использованием μ_1, каждое четвертое с использованием μ_2 и т. д. Для μ_1 значение $\beta_{-1}(1)=1$, что обеспечивает максимальную частоту изменения старшего бита кода элемента x_{ng} последовательности Соболя. Максимальная частота изменения следующего бита кода x_{ng} в два раза меньше и т. д. (см. табл. 3).

Следует отметить, что наиболее важным элементом последовательности Соболя являются направляющие числа μ_i, $i=\overline{1,w}$, приведенные в табл. 4, которые должны принимать уникальные значения для каждой координаты точек в s-мерном пространстве. Наиболее значимым свойством направляющих чисел, вытекающим из ограничений на эти числа и подтверждающимся видом введенной авторами матрицы модифицированных порождающих чисел, является их линейная независимость.

Для получения полного множества модифицированных направляющих чисел μ_i, $i=\overline{1,w}$, на основании $m<w$ исходных, так же, как и в оригинальном методе Соболя, будем использовать примитивные порождающие полиномы. В общем виде примитивный полином имеет вид $\varphi(x)=1\oplus\lambda_1x^1\oplus\lambda_2x^2\oplus\ \ldots\ \oplus\lambda_{m-1}x^{m-1}\oplus x^m$, где $m=\deg\varphi(x)$, а двоичные коэффициенты $\lambda_i\in\{0,1\}$, $i=\overline{1,m-1}$, определяют конкретный вид полинома [32]. Подобный подход, когда на основании

m исходных направляющих чисел и порождающего полинома $\varphi(x)$ генерируются остальные $w–m$ числа, представлен в оригинальном методе Соболя и его классических модификациях, в том числе и в экономичном способе Антонова – Салеева.

Рассмотрим алгоритм формирования модифицированных направляющих чисел, представленных в виде нижней треугольной матрицы с единичной главной диагональю (см. табл. 4).

При реализации каждой итерации значение предыдущих (ранее полученных) направляющих чисел будем модифицировать с целью формирования двоичных кодов, разрядность которых увеличивается от первоначальных m бит до результирующих w бит. Формально это достигается сдвигом направляющего числа на один разряд влево с одновременным приписыванием в младшем разряде двоичного нуля. Подобная модификация математически записывается как

$$\mu_j = \mu_j \times 2^1, \quad j = \overline{1,i}, \quad m \le i < w.$$

Значение μ_j в левой части равенства есть новое значение направляющего числа μ_j, полученное на основании его предыдущего значения. Само рекуррентное соотношение примет вид

$$\mu_{i+1} = \lambda_1\mu_i \oplus \lambda_2\mu_{i-1} \oplus \ldots \oplus \lambda_{m-1}\mu_{i-m+2} \oplus \mu_{i-m+1} \oplus (\mu_{i-m+1} / 2^m). \qquad (2)$$

Значение $\mu_{i-m+1}/2^m$ представляет собой сдвинутую копию вправо на m позиций двоичного кода μ_{i-m+1}.

Например, для случая последовательности Соболя длиной $2^w–1=2^5–1=31$, где $w=5$, используем примитивный полином $\varphi(x)=1\oplus x^1\oplus x^3$ степени $m=3$, а для первых $m=3$ направляющих чисел возьмем целые нечетные числа $m_1 = 1$, $m_2 = 3$ и $m_3 = 5$. Тогда v_1 =0,100, v_2 = 0,110, v_3 = 0,101 и соответственно $\mu_1 = v_1 2^3 = 100$, $\mu_2 = v_2 2^3 = 110$ и $\mu_3 = v_3 2^3 = 101$. Так как $w=5$, остальные $(w–m)$ направляющие числа, в данном случае 5–3=2 числа, генерируются с использованием рекуррентного соотношения (2) для заданных значений m и w:

$$\mu_j = \mu_j \times 2^1, \quad j = \overline{1,i}, \quad 3 \le i < 5;$$

$$\mu_{i+1} = \mu_i \oplus \mu_{i-2} \oplus (\mu_{i-2} / 2^3).$$

Для $i=3$ получим $\mu_1=\mu_1\times2=1000$, $\mu_2=\mu_2\times2=1100$ и $\mu_3=\mu_3\times2=1010$, кроме того, $\mu_1/2^3=0001$. Тогда $\mu_4=\mu_3\oplus\mu_1\oplus(\mu_1/2^3)=1010\oplus1000\oplus0001=0011$, и далее для $i=4$ получим $\mu_5=\mu_4\oplus\mu_2\oplus(\mu_2/2^3)$ = = 00110 ⊕ 11000 ⊕ 00011 = 11101.

Последовательность Соболя, соответствующая полученным направляющим числам μ_1=10000, μ_2=11000, μ_3=10100, μ_4=00110 и μ_5=11101, приведена в табл. 5.

Таблица 5

Последовательность Соболя

n	T_q	x_{ng}	n	T_q	x_{ng}	n	T_q	x_{ng}	n	T_q	x_{ng}
1	1	10000	9	1	00010	17	1	01011	25	1	11001
2	2	01000	10	2	11010	18	2	10011	26	2	00001
3	1	11000	11	1	01010	19	1	00011	27	1	10001
4	3	01100	12	3	11110	20	3	10111	28	3	00101
5	1	11100	13	1	01110	21	1	00111	29	1	10101
6	2	00100	14	2	10110	22	2	11111	30	2	01101
7	1	10100	15	1	00110	23	1	01111	31	1	11101
8	4	10010	16	5	11011	24	4	01001			

Второй модификацией последовательностей Соболя является использование в качестве направляющих чисел w последовательных значений М-последовательности, формируемых в соответствии с примитивным порождающим полиномом степени w. В отличие от классического метода формирования направляющих чисел в данном случае генерируется все множество w чисел, а не только $w–m$, на основании m исходных чисел.

Для соблюдения всех свойств последовательностей Соболя необходимо, чтобы μ_1 всегда равнялось w-разрядному коду 100…0 (см. табл. 4). Для получения всего множества модифицированных направляющих чисел двоичный код 100…0 используем в качестве начального состояния генератора M-последовательности и сформируем w–1 последующих состояний генератора, которые будем использовать как $\mu_2, \mu_3, \ldots, \mu_w$.

Например, для полинома $\varphi(x) = 1 \oplus x^1 \oplus x^4$ степени $m = 4$ направляющие числа принимают значения $\mu_1 = 1000$, $\mu_2 = 1100$, $\mu_3 = 1110$ и $\mu_4 = 1111$. Изменив порождающий полином на $\varphi(x) = 1 \oplus x^3 \oplus x^4$, получим новое множество модифицированных направляющих чисел, а именно $\mu_1 = 1000$, $\mu_2 = 0100$, $\mu_3 = 0010$ и $\mu_4 = 1001$. Для двух приведенных примеров соответствующие треугольные матрицы V_1 и V_2 с единичной главной диагональю, построенные на основании модифицированных направляющих чисел, имеют вид

$$V_1 = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{vmatrix}; \qquad V_2 = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{vmatrix}.$$

Для каждого множества направляющих чисел, описываемых матрицами V_1 и V_2, может быть сформирована уникальная последовательность Соболя.

Очевидно, что количество множеств модифицированных порождающих чисел определяется количеством примитивных порождающих полиномов степени w. Для общего случая это количество вычисляется как $\Phi(2^w–1)/w$, где Ф есть функция Эйлера от целого числа $2^w–1$ [32].

Следующей модификацией последовательностей Соболя является использование для формирования произвольных значений $\beta_{–j}(i) \in \{0,1\}$ для $j < i$ последовательных $(w^2–w)/2$ бит произвольной псевдослучайной последовательности, в том числе и M-последовательности. Возможность формирования подобным образом модифицированных направляющих чисел μ_i следует из того факта, что равенство единице младшего бита двоичного представления целого числа свидетельствует о его нечетности. Это отражено, например, в табл. 4 в виде равенства $\beta_{–i}(i) = 1$, $i = \overline{1,w}$. Очевидно, что старшие биты нечетного целого числа могут принимать произвольные значения.

В случае модифицированных направляющих чисел, как отмечалось ранее, значение μ_1 всегда равняется w-разрядному коду 100…00. Соответственно $\mu_2 = \beta_{–1}(2)10\ldots00$, где $\beta_{–1}(2) \in \{0, 1\}$, $\mu_3 = \beta_{–1}(3)\beta_{–2}(3)10\ldots00$, где $\beta_{–1}(3)$ и $\beta_{–2}(3) \in \{0, 1\}$ и т. д. Значение $\mu_w = \beta_{–1}(w)\beta_{–2}(w)\beta_{–3}(w) \ldots \beta_{-w+1}(w)1$, где $\beta_{–j}(w) \in \{0,1\}$ для $j < w$. Таким образом, $\beta_{–j}(i) \in \{0,1\}$ при $j<i$ для модифицированных направляющих чисел μ_i могут принимать произвольные двоичные значения. Количество возможных вариантов значений $\beta_{–j}(i) \in \{0,1\}$ при $j < i$ определяется их числом $(w^2–w)/2$ и вычисляется как $2^{(w^2-w)/2}$. Это подтверждается анализом табл. 4 и матриц V_1 и V_2.

В качестве примера рассмотрим случай, когда w=4. Тогда число значений $\beta_{–j}(i) \in \{0,1\}$ при $j < i$, где $i = \overline{1,4}$, равняется $(w^2–w)/2=(4^2–4)/2=6$. Это означает, что шесть элементов матрицы модифицированных направляющих чисел, находящихся под главной диагональю, могут принимать произвольные двоичные значения. Количество подобных матриц и соответственно множеств модифицированных направляющих чисел равняется $2^{(w^2-w)/2} = 2^6 = 64.$ Две из указанных матриц V_1 и V_2 приводились ранее.

Максимально возможное количество уникальных множеств из w модифицированных направляющих чисел μ_i равняется $2^{(w^2-w)/2}$, а их элементы могут формироваться с использованием различных алгоритмов генерирования равновероятных двоичных цифр. В случае использования для этих целей M-последовательностей единственным ограничением на степень m порождающего полинома $\varphi(x)$ является равенство $\deg\varphi(x)=(w^2–w)/2$. Выполнение данного равенства обеспечивает формирование различных вариантов значений $\beta_{–j}(i) \in \{0,1\}$ при $j<i$ и соответственно всевозможных последовательностей Соболя для заданного значения w. В случае когда

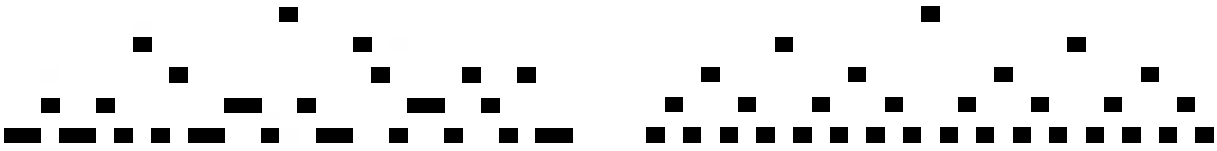
$w = 4$, для порождающего полинома $\varphi(x)$ необходимо, чтобы $\deg\varphi(x) = 6$, что позволит получить все возможные матрицы вида V_1 и V_2, которые будут отличаться значениями элементов, находящихся ниже главной диагонали.

6. Применение последовательностей Соболя в качестве тестовых воздействий

Основой для применения последовательностей Соболя в качестве тестовых воздействий служит их «более равномерное» распределение по сравнению с псевдослучайными последовательностями, которые являются базовыми для реализации практически всех современных методов тестирования вычислительных систем [5, 7, 9, 11, 12, 18, 22–26].

Распределение последовательных псевдослучайных значений (рис. 6, *а*), полученных на основании полинома $\varphi(x)=1\oplus x^2\oplus x^5$, и значений последовательности Соболя (рис. 6, *б*), представленных в табл. 5, характеризуется различной степенью равномерности.

На рис. 6 показана временная последовательность генерирования псевдослучайных значений {16, 8, 20, 10, 21, 26, 29, 14, 23, 27, 13, 6, 3, 17, 24, 28, 30, 31, 15, 7, 19, 25, 12, 22, 11, 5, 18, 9, 4, 2, 1} и элементов последовательности Соболя {16, 8, 24, 12, 28, 4, 20, 18, 2, 26, 10, 30, 14, 22, 6, 27, 11, 19, 3, 23, 7, 31, 15, 9, 25, 1, 17, 5, 21, 13, 29} на интервале целых чисел от 1 до 31 в зависимости от длины последовательностей 2^k-1, $k = \overline{1, w}$. Из рис. 6 видно, что для $k = 3$ последовательность значений 16, 8, 20, 10, 21, 26, 29 из $2^3-1 = 7$ псевдослучайных чисел распределена «менее равномерно» по сравнению с числами Соболя 16, 8, 24, 12, 28, 4, 20. В общем случае указанное свойство «большей равномерности» выполняется для всех значений k и пространств различной размерности [27, 33, 34].



а) *б)*

Рис. 6. Примеры последовательностей: *а)* псевдослучайной; *б)* Соболя

Структура квазислучайных последовательностей Соболя максимально равномерно распределяет тестовые воздействия по всему пространству входных наборов, что позволит достичь большей их покрывающей способности по отношению к типовым входным тестовым наборам (рис. 3 и 4). Напомним, что, как и в известных методах тестирования вычислительных систем [5, 7, 9, 11, 12, 18, 22–26], в данном случае также используется принцип черного ящика. Факт максимально равномерного распределения тестовых воздействий позволяет увеличить значение традиционных мер оценки качества тестов, а именно вероятности обнаружения тестом хотя бы одной неисправности (P-measure) и вероятности обнаружения ожидаемого количества неисправностей (E-measure) [21–24].

В качестве экспериментального подтверждения эффективности кавазислучайного тестирования вычислительных систем в рамках данной статьи были проведены исследования применимости данного подхода для тестирования запоминающих устройств. Исследовались запоминающие устройства, состоящие из матрицы 1024×1024 запоминающих ячеек. В качестве их неисправностей рассматривались многократные неисправности, представляющие собой блоки из 2×2, 3×3 и 4×4 ячеек, содержащие константные неисправности. В качестве теста использовался MATS+, который характеризуется 100%-й полнотой обнаружения константных неисправностей. Рассматривались два случая формирования адресных последовательностей, а именно псевдослучайные адресные последовательности и квазислучайные двухмерные последовательности Соболя. Отмечено существенное уменьшение времени тестирования до обнаружения факта неисправного поведения запоминающего устройства. Во всех случаях длина теста в среднем уменьшается на величины от 44 до 53 % по отношению к псевдослучайным адресным последовательностям. Данные исследования подтверждают результаты,

ранее полученные для программных приложений. Так же, как и в случае управляемых вероятностных тестов [24–26], квазислучайные тесты позволяют существенно уменьшить количество тестовых наборов для обнаружения первой неисправности (F-measure) в программном продукте [21–23]. Результаты, приведенные в [31], свидетельствуют о 50 %-м улучшении значений данной метрики даже в случае использования простейших классических квазислучайных последовательностей.

Заключение

В статье рассмотрена причинно-следственная зависимость между неисправными состояниями вычислительных систем и типовыми тестовыми воздействиями для их обнаружения. Показана идентичность типовых тестовых воздействий как для программной, так и для аппаратной части вычислительных систем, что позволяет использовать единую методологию для тестирования вычислительных систем в целом. В качестве последовательностей тестовых воздействий вычислительных систем, адекватно покрывающих множество типовых входных значений (рис. 3 и 4), обосновывается использование квазислучайных последовательностей. Основное внимание уделено последовательностям Соболя как наиболее эффективным для формирования бинарных тестовых воздействий. Проведенный анализ последовательностей Соболя позволил предложить ряд модификаций, позволяющих существенно увеличить их количество.

Список литературы

1. Rajsuman, R. System-On-A-Chip: Design and Test / R. Rajsuman. – London : Artech House Publishers, 2000. – 303 p.

2. Иванюк, А.А. Проектирование встраиваемых цифровых устройств и систем / А.А. Иванюк. – Минск : Бестпринт, 2012. – 338 с.

3. Semiconductor Intellectual Property: Continuing on the Path to Grow. ASIC IP report / Semiconductor Research Corp. Research Triangle Park. – NC, USA, 2007. – 100 p.

4. Harris, I.G. Hardware-Software Co-validation: Fault Models and Test Generation / I.G. Harris // IEEE Design & Test of Computers. – 2003. – Vol. 20, № 1. – P. 40–47.

5. Hamill, M. Common Trends in Software Fault and Failure Data / M. Hamill, K. Goseva-Popstojanova // IEEE Transaction on Software Engineering. – 2009. – Vol. 35, № 4. – P. 484–496.

6. Comparative Analysis of Hardware and Software Fault Tolerance: Impact on Software Reliability Engineering / H. Ammar [et al.] // Institute for Software Research Fairmont, WV 26554 USA [Electronic resourse] January 15, 1999. – Mode of access : www.isr.wvu.edu. – Date of access : 12.07.2012.

7. Ярмолик, С.В. Маршевые тесты для самотестирования ОЗУ / С.В. Ярмолик, А.П. Занкович, А.А. Иванюк. – Минск : Изд. центр БГУ, 2009. – 270 с.

8. White, L. A domain strategy for computer program testing / L. White, E. Cohen // IEEE Transaction on Software Engineering. – 1980. – Vol. SE–6, № 3. – P. 247–257.

9. Schnekenburger, C. Towards the determination of typical failure patterns / C. Schnekenburger, J. Mayer // In Proc. of Fourth Intern. Workshop on Software Quality Assurance: in conjunction with the 6th ESE/FSE joint meeting, ser. SOQUA'07. – N.Y., USA : ACM, 2007. – P. 90–93.

10. Proportional sampling strategy guidelines for software testing practitioner / F.T. Chan [et al.] // Information and Software Technology – 1996. – Vol. 38, № 12. – P. 775–782.

11. Shahbazi, A. Centroidal Voronoi Tessellation – a New Approach to Random Testing / A. Shahbazi, A.F. Tappenden, J. Miller // IEEE Transaction on Software Engineering. – 2013. – Vol. 39, № 2. – P. 163–183.

12. Fault Pattern Oriented Defect Diagnosis for Memories / C.W. Wang [et al.] // In Proc. of Intern. Test Conference (ITC 2003). – Charlotte, NC, USA, 2003. – P. 29–38.

13. Using electrical bitmap results from embedded memory to enhance yield / A. Segal [et al.] // IEEE Design & Test of Computers. – 2001. – Vol. 15, № 3. – P. 28–39.

14. Goor, A.J. Testing Semiconductor Memories, Theory and Practice / A.J. Goor. – UK, Chichester : John Wiley & Sons, 1991. – 536 p.

15. Barzilai, Z. Exhaustive Generation of Bit Pattern with Application to VLSI Self–Testing / Z. Barzilai, D. Coppersmith, A. Rozenberg // IEEE Transactions on Computers. – 1983. – Vol. C–31, № 2. – P. 190–194.

16. Furuya, K. A probabilistic approach to locally exhaustive testing / K. Furuya // IEEE Transactions on IEICE. – 1989. – Vol. E72, № 5. – P. 656–660.

17. Testing of embedded RAM using exhaustive random sequence / H. Maeno [et al.] // In Proc. of Intern. Test Conference (ITC 1987). – Washington, D.C., USA, 1987. – P. 105–110.

18. Mrozek, I. Antirandom Test Vector for BIST in Hardware/Software Systems / I. Mrozek, V. Yarmolik // Fundamenta Informaticae. – 2012. – Vol. 119, № 2. – P. 163–185.

19. Malaiya, Y.K. The coverage problem for random testing / Y.K. Malaiya, S. Yang // In Proc. of Intern. Test Conference (ITC 1984). – Philadelphia, PA, USA, 1984. – P. 237–242.

20. Malaiya, Y.K. An examination of fault exposure ratio / Y.K. Malaiya, A. Mayrhauser, P.K. Srimani // IEEE Transactions on Software Engineering. – 1993. – Vol. 19, № 11. – P. 1087–1094.

21. Malaiya, Y.K. Antirandom Testing: Getting the most out of Back-Box Testing / Y.K. Malaiya, S. Yang // In Proc. of Sixth Intern. Symposium on Software Reliability Engineering. – Toulouse, France, 1995. – P. 86–95.

22. Antirandom Testing: A Distance-Based Approach / S.H. Wu [et al.] // VLSI Design. – 2008. – № 2. – P. 1–9.

23. Fast Antirandom (FAR) Test Generation / A. Mayrhause [et al.] // In Proc. of the Third IEEE Intern. High-Assurance System Engineering Symposium. – Washington, D.C., USA, 1998. – P. 262–269.

24. Mrozek, I. Iterative Antirandom Testing / I. Mrozek, V.N. Yarmolik // Journal of Electronic Testing: Theory and Applications (JETTA). – 2012. – Vol. 9, № 3. – P. 251–266.

25. Ярмолик, С.В. Управляемое случайное тестирование / С.В. Ярмолик, В.Н. Ярмолик // Информатика. – 2011. – № 1(29). – С. 79–88.

26. Ярмолик, С.В. Управляемые вероятностные тесты / С.В. Ярмолик, В.Н. Ярмолик // Автоматика и телемеханика. – 2012. – № 10. – С. 142–155.

27. Sobol, I.M. Uniformly distributed sequences with an additional uniform property / I.M. Sobol // USSR Comput. Math. Math. Phys. – 1976. – Vol. 16. – P. 236–242.

28. Niederreiter, H. Point sets and sequences with small discrepancy / H. Niederreiter // Monatshefte fur Mathematik. – 1987. – Vol. 104, № 4. – P. 273–337.

29. Halton, J.H. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals / J.H. Halton // Numerische Mathemati. – 1960. – Vol. 2, № 1. – P. 84–90.

30. Chi, H. Computational investigations of quasi-random sequences in generating test cases for specification-based tests / H. Chi, E.I. Jones // Proc. of the 38th on Winter Conference, ser. WSC'06. Winter Simulation Conference. – Monterey, CA, USA, 2006. – P. 975–980.

31. Chen, T.Y. Quasi-Random Testing / T.Y. Chen, R. Merkel // IEEE Transaction on Reliability. – 2007. – Vol. 56, № 3. – P. 562–568.

32. Ярмолик, В.Н. Генерирование и применение псевдослучайных сигналов в системах испытаний и контроля / В.Н. Ярмолик, С.Н. Демиденко. – Минск : Наука и техника, 1986. – 200 с.

33. Соболь, И.М. Вычисление несобственных интегралов при помощи равномерно распределенных последовательностей / И.М. Соболь // Доклады АН СССР. – 1973. – Т. 210, № 2. – С. 278–281.

34. Соболь, И.М. Точки, равномерно заполняющие многомерный куб / И.М. Соболь. – М. : Знание, 1985. – 32 с.

35. Savage, C. A survey of combinatorial Gray code // SIAM Review / C. Savage. – 1997. – Vol. 39, № 4. – P. 605–629.

Поступила 02.06.13

Белорусский государственный университет
информатики и радиоэлектроники,
Минск, П. Бровки, 6
e-mail: yarmolik@cosmostv.by,
yarmolik10ru@yahoo.com

S.V. Yarmolik, V.N. Yarmolik

QUASI-RANDOM TESTING OF COMPUTER SYSTEMS

Various modified random testing approaches have been proposed for computer system testing in the black box environment. Their effectiveness has been evaluated on the typical failure patterns by employing three measures, namely, P-measure, E-measure and F-measure. A quasi-random testing, being a modified version of the random testing, has been proposed and analyzed. The quasi-random Sobol sequences and modified Sobol sequences are used as the test patterns. Some new methods for Sobol sequence generation have been proposed and analyzed.

УДК 681.32

А.А. Иванюк

ПРОЕКТИРОВАНИЕ КОНФИГУРИРУЕМОГО СДВИГОВОГО РЕГИСТРА С ЛИНЕЙНОЙ ОБРАТНОЙ СВЯЗЬЮ

Рассматривается методика проектирования конфигурируемого сдвигового регистра, для которого возможно задание его разрядности в различных режимах функционирования. Предложенная схема сдвигового регистра с линейной обратной связью позволяет использовать его в качестве циклического сдвигового регистра, генератора М-последовательности, счетчика Джонсона и одноканального сигнатурного анализатора. Приводится оценка аппаратурных затрат на реализацию конфигурируемого сдвигового регистра.

Введение

Среди многообразия цифровых генераторов псевдослучайных последовательностей (ГПП) наиболее широкое распространение получили генераторы М-последовательностей, построенные на основе сдвиговых регистров с линейной обратной связью (от англ. Linear Feedback Shift Register, LFSR) [1]. В задачах контроля и диагностики средств вычислительной техники LFSR используются для синтеза генераторов псевдослучайных тестовых последовательностей и сигнатурных анализаторов [2]; в криптографии – для генерирования символов псевдослучайных числовых последовательностей с дискретным равномерным распределением [3], синтеза схем шифрования в потоковых криптосистемах [4]; в системах телекоммуникаций – для аппаратной реализации схем помехоустойчивого кодирования [5], схем скремблирования [6] и т. д.

В статье рассматривается один из возможных вариантов реализации конфигурируемого LFSR, для которого предусмотрено динамическое изменение его разрядности и режимов функционирования.

1. Теоретические основы проектирования цифровых устройств на основе LFSR

В основе структуры LFSR лежит *n*-разрядный сдвиговый регистр, проектируемый, как правило, при помощи синхронных *D*-триггеров. Синтез LFSR осуществляется на основе характеристического полинома

$$\varphi(x) = 1 \oplus \alpha_1 x^1 \oplus \alpha_2 x^2 \oplus \ldots \oplus \alpha_{n-1} x^{n-1} \oplus \alpha_n x^n, \quad (1)$$

где $n = \deg(\varphi(x))$ определяет число разрядов LFSR, а коэффициенты $\alpha_i \in \{0,1\}$ ($i = \overline{1,n}$) используются для формирования значения сигнала в цепи обратной связи.

Пусть $d_i \in \{0,1\}$ ($i = \overline{1,n}$) есть значение, хранящееся на *i*-м триггере, а $D^{(k)} = \{d_1^{(k)}, d_2^{(k)}, \ldots, d_n^{(k)}\}$ – *n*-разрядное двоичное слово, являющееся состоянием LFSR в *k*-й такт функционирования. Предположим, что в (*k*+1)-й такт функционирования при наступлении фронта сигнала синхронизации, являющегося общим для всех триггеров LFSR, осуществляется операция поразрядного сдвига двоичного слова, такая, что

$$d_i^{(k+1)} = d_{i-1}^{(k)}, \forall i = \overline{2,n}. \quad (2)$$

Новое значение младшего разряда LFSR при этом вычисляется исходя из значений коэффициентов полинома (1):

$$d_1^{(k+1)} = \bigoplus_{i=1}^{n} \alpha_i d_i^{(k)}. \quad (3)$$

Соответствующий выбор полинома (1) и начального состояния $D^{(0)}$ определяет вид последовательности, вырабатываемой LFSR. Например, при $\varphi(x)=1\oplus x^n$ и $D^{(0)}=\{1,0,0,...,0\}$ последовательность вырабатываемых двоичных слов $(D^{(0)}, D^{(1)}, D^{(2)}, ..., D^{(n-1)}, D^{(0)})$ будет представлять собой циклическую двоичную последовательность типа «one hot» с периодом повторения символов, равным n. С учетом того что $\alpha_1=\alpha_2=...=\alpha_{n-1}=0$ и $\alpha_n=1$, выражение (3) принимает следующий вид:

$$d_1^{(k+1)}=d_n^{(k)}. \tag{4}$$

Выражение (4) совместно с (2) может быть использовано для синтеза структуры генератора вышеописанной циклической последовательности.

Если характеристический полином $\varphi(x)$ является примитивным, период повторения вырабатываемых символов равен 2^n-1. Такого рода двоичные последовательности называются М-последовательностями [2] и по своим вероятностным характеристикам являются псевдослучайными последовательностями, при этом LFSR называется генератором М-последовательности либо ГПП.

Для синтеза *n*-разрядного ГПП на основе LFSR необходимо выбрать соответствующий примитивный полином степени *n* ($\deg(\varphi(x))=n$). Известно, что число примитивных полиномов степени *n* над полем GF(2) можно вычислить по формуле [7]

$$M(n)=\frac{L(2^n-1)}{n}, \tag{5}$$

где L – функция Эйлера.

Например, для $n=4$ существует $M(4)=2$ примитивных полинома $\varphi_1(x)=1\oplus x\oplus x^4$ и $\varphi_2(x)=1\oplus x^3\oplus x^4$, на основе которых можно спроектировать два генератора М-последовательности. С учетом одинаковых начальных состояний такие генераторы за 15 тактов функционирования выработают 15 символов двух М-последовательностей, но с различным порядком их следования. Это свойство характерно для всех ГПП, синтезированных на основе примитивных полиномов с одинаковым значением их старших степеней.

Генераторы М-последовательностей часто применяются в качестве источников тестовых воздействий при решении задач тестирования цифровых устройств, при которых реакции на тестовые воздействия сжимаются в компактную характеристику, называемую сигнатурой [2]. Аппаратура сжатия при этом называется сигнатурным анализатором [2], структура которого может быть синтезирована по схожим принципам, что и генератор М-последовательности. В общем случае одноканальный сигнатурный анализатор (ОСА) представляет собой ГПП, в цепи обратной связи которого присутствует дополнительный элемент XOR (исключающее ИЛИ), на один из входов которого подается символ сжимаемой последовательности. При этом значение младшего разряда LFSR описывается следующим образом:

$$d_1^{(k+1)}=d_0^{(k)}\oplus\left(\bigoplus_{i=1}^{n}\alpha_i d_i^{(k)}\right), \tag{6}$$

где $d_0^{(k)}\in\{0,1\}$ – значение сжимаемого символа.

При условиях, что $D^{(0)}\neq\{0,0,0,...,0\}$, $d_0^{(k)}=0$ $(\forall k=0,1,2,...)$ и α_i есть коэффициенты примитивного полинома $\varphi(x)$, аппаратура ОСА будет функционировать в качестве генератора М-последовательности. Если $\varphi(x)=1\oplus x^n$ и $d_0^{(k)}=1$ $(\forall k=0,1,2,...)$, выражение (6) принимает следующий вид:

$$d_1^{(k+1)} = 1 \oplus d_n^{(k)} = \overline{d_n^{(k)}}, \tag{7}$$

что для случая $n = 2^r$ $(\forall r = 1, 2, 3, ...)$ является выражением для вычисления нового значения младшего разряда счетчика Джонсона [8], вырабатывающего псевдослучайную последовательность с периодом, равным $2n$. Таким образом, соответствующие значения α_i и $d_0^{(k)}$ при заданном n позволяют посредством выражений (2) и (6) описать функционирование четырех различных цифровых устройств: генератора циклической последовательности, генератора М-последовательности, одноканального сигнатурного анализатора и счетчика Джонсона.

С целью определения произвольной разрядности в пределах значения n вышеперечисленных аппаратных структур введем дополнительные коэффициенты $\beta_j \in \{0,1\}$ $(\forall j = \overline{1, n})$ (рис. 1).

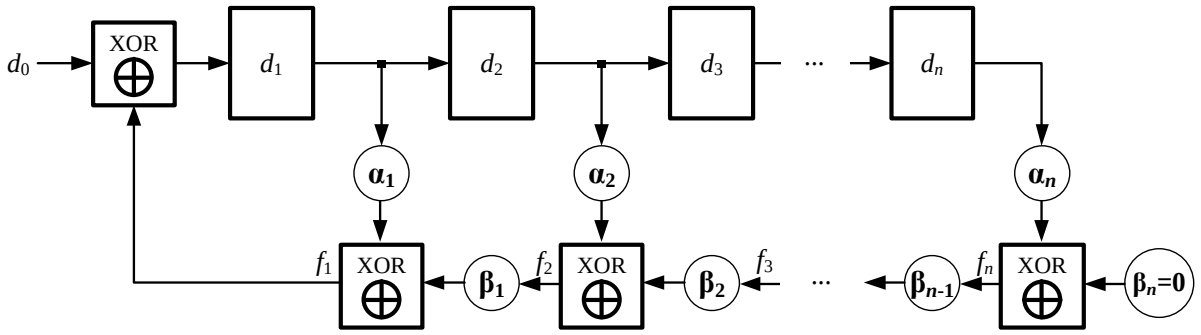


Рис. 1. Обобщенная структура конфигурируемого сдвигового регистра

Нулевое значение коэффициента β_j означает, что соответствующий ему разряд d_j является старшим разрядом в конфигурируемой структуре и все последующие разряды d_l $(j < l \le n)$ не участвуют в формировании значения сигнала f_1 в цепи обратной связи. При этом значение f_1 можно выразить следующим образом:

$$\begin{aligned} f_1 &= \alpha_1 d_1 \oplus \beta_1 f_2 = \alpha_1 d_1 \oplus \alpha_2 \beta_1 d_2 \oplus \beta_1 \beta_2 f_3 = ... = \\ &= \alpha_1 d_1 \oplus \alpha_2 \beta_1 d_2 \oplus \alpha_3 \beta_1 \beta_2 d_3 \oplus ... \oplus \alpha_n \beta_1 \beta_2 ... \beta_{n-1} f_n. \end{aligned} \tag{8}$$

С учетом того что для старшего используемого разряда $f_n = \alpha_n d_n$ $(\beta_n = 0)$, выражение (8) можно записать в более компактной форме:

$$f_1 = \alpha_1 d_1 \oplus \left(\bigoplus_{i=1}^{n-1} \alpha_{i+1} d_{i+1} \prod_{j=1}^{i} \beta_j \right). \tag{9}$$

Таким образом, единичные значения коэффициентов $\beta_1 = \beta_2 = ... = \beta_m = 1$ определяют количество разрядов $m \le n$, участвующих в конфигурации сдвигового регистра с линейной обратной связью.

В общем случае значение младшего разряда конфигурируемого сдвигового регистра описывается выражением

$$d_1^{(k+1)} = d_0^{(k)} \oplus \alpha_1 d_1^{(k)} \oplus \left(\bigoplus_{i=1}^{n-1} \alpha_{i+1} d_{i+1}^{(k)} \prod_{j=1}^{i} \beta_j \right). \tag{10}$$

Развитие идеи компактного тестирования привело к появлению конфигурируемых структур на подобие BILBO (от англ. Built-In Logic Block Observer) [9], которые, будучи построенными на LFSR, могут функционировать как в качестве генераторов тестовых последовательностей, так и в качестве сигнатурных анализаторов. Двойственное функционирование BILBO обусловлено наличием реконфигурируемых блоков, которые обеспечивают соответствующую коммутацию сигналов в зависимости от задаваемого режима. Для BILBO определены четыре основных режима: режим нормального функционирования, при котором триггеры, входящие в состав BILBO, играют роль элементов памяти устройства; режим сдвигового регистра; режим генератора тестовых последовательностей (ГТП) и режим сигнатурного анализатора (СА) [9].

В работе [10] было показано, что для увеличения достоверности встроенного самотестирования посредством BILBO необходимо применять ГТП и СА с использованием различных полиномов.

В общем случае задачу проектирования конфигурируемого LFSR можно сформулировать как задачу синтеза сдвигового регистра с различными задаваемыми коэффициентами α_i и с различным значением числа разрядов в пределах n.

Для решения данной задачи было предложено множество подходов [10–14]. Так, в работе [10] предлагается структура LFSR с фиксированным параметром n и возможностью задания различных коэффициентов α_i. В работе [11] предлагается 128-разрядный сдвиговый регистр с множеством фиксированных коэффициентов α_i для возможности реализации LFSR произвольной разрядности в пределах от $n = 8$ до $n = 128$. Дальнейшим развитием работы [11] стала публикация [12], предлагающая идею реконфигурируемого LFSR с целью обеспечения различных базовых операций для программно-определяемых радиосистем (от англ. Software-Defined Radio, SDR). В работе [13] была предложена архитектура LFSR, состоящая из 64 базовых реконфигурируемых элементов, каждый из которых содержит настраиваемый 8-разрядный сдвиговый регистр данных и 32 8-разрядных конфигурационных регистра, позволяющих настраивать структуру LFSR на произвольную разрядность для осуществления различных операций над элементами полей GF(2), GF(2^8), GF(2^{16}) либо GF(2^{32}). В работе [14] рассматривается задача аппаратной реализации генераторов псевдослучайных последовательностей с перестраиваемой структурой на основе теории клеточных автоматов.

В настоящей работе рассмотрим методику проектирования конфигурируемого сдвигового регистра посредством языка VHDL с дальнейшей его реализацией для программируемых логических интегральных схем типа FPGA.

2. Проектирование конфигурируемого сдвигового регистра

Модульность и высокий уровень абстракции языка VHDL позволяют описывать схемотехнические элементы цифровых устройств произвольной сложности [15]. Кроме того, VHDL позволяет составлять параметризированные проектные описания для случая итерационных цифровых структур. В связи с этим для составления VHDL-описаний LFSR-структур могут быть применены следующие подходы:

1. Составление непараметризированного описания с фиксированными значениями n и α_i.

2. Составление параметризированного описания с произвольно задаваемыми значениями n и α_i.

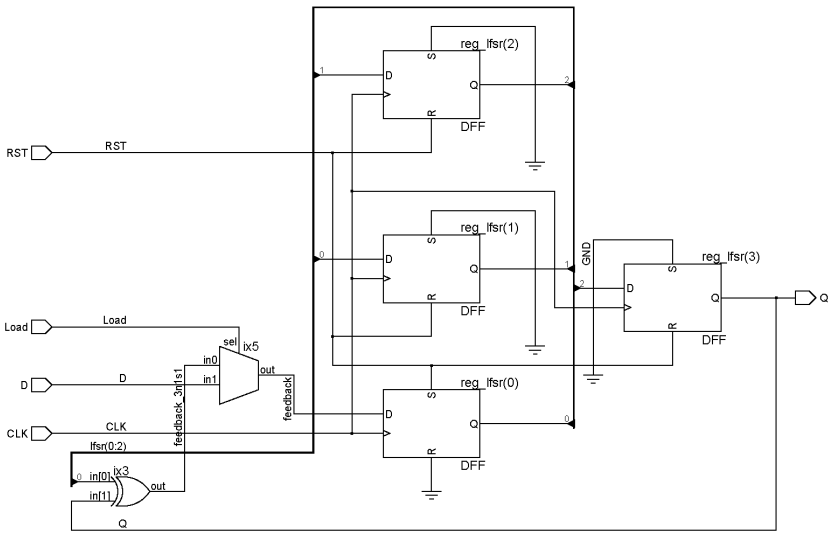
3. Составление параметризированного описания конфигурируемого сдвигового регистра с произвольно задаваемыми значениями α_i и β_j.

Применение первого подхода позволяет достичь минимальных аппаратурных затрат при синтезе описываемой LFSR-структуры, однако изменение разрядности LFSR либо множества коэффициентов α_i приведет к изменению исходного VHDL-описания и повторному циклу проектирования устройства.

Рассмотрим пример непараметризированного описания генератора М-последовательности и результат его синтеза для $n = 4$ и $\varphi(x) = 1 \oplus x \oplus x^4$ (рис. 2).

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity LFSR4 is
   port ( CLK, RST, Load, D : in   std_logic;
                              Q : out std_logic );
end LFSR4;

architecture Behavioral of LFSR4 is
signal feedback   : std_logic;
signal lfsr       : std_logic_vector( 0 to 3 );
constant init     : std_logic_vector( 0 to 3 ) := "1000";
begin
PSHIFT: process( RST, CLK, feedback, lfsr )
begin
        if ( RST = '1' ) then
                lfsr <= init;
        elsif rising_edge( CLK ) then
                lfsr <= feedback & lfsr( 0 to 2);
        end if;
end process;

PFB: process( Load, D, lfsr )
begin
        if ( Load = '1' ) then
                feedback  <= D;
        else
                feedback <= lfsr( 0 ) xor lfsr( 3 );
        end if;
end process;

Q <= lfsr(3);

end behavioral;
```

а)



б)

Рис. 2. Генератор псевдослучайной последовательности:
а) описание ГПП для $n = 4$; *б)* результат его RTL-синтеза

Из рис. 2 видно, что параметр $(n-1)$ в явном виде присутствует как при объявлении сигналов, так и при описании подсхем LFSR, а именно в процессе PSHIFT, описывающем синхронный сдвиговый регистр; в процессе PFB, описывающем двухвходовый элемент XOR и мультиплексор, необходимые для формирования значения сигнала линейной обратной связи, и в последнем параллельном операторе, описывающем передачу одного бита вырабатываемой последовательности на выходной порт устройства.

Определение начального состояния ГПП возможно двумя способами: асинхронно при установке единичного значения сигнала на входном порту *RST*, при этом сдвиговый регистр принимает фиксированное значение "1000"; синхронно при удержании единичного значения сигнала на входном порту *Load*. В первом случае изменение инициализирующего значения регистра возможно только при составлении его проектного описания, во втором случае – во время функционирования устройства по назначению.

Основной задачей при трансформации представленного описания в параметризированное описание ГПП для целочисленного параметра *n* будет являться описание оператора, формирующего значение сигнала обратной связи в процессе PFB для произвольно задаваемых коэффициентов α_i. Для этого введем generic-параметр *ALPHA* безразмерного типа std_logic_vector, значение которого будет определять бинарную маску множества коэффициентов $\{\alpha_1, \alpha_2, ..., \alpha_n\}$ (рис. 3).

Из рис. 3 видно, что параметр $ALPHA := "1011"$ задает порождающий полином $\varphi(x) = 1 \oplus x^3 \oplus x^4$. Размерность таких объектов, как lfsr и init, можно определить посредством оператора **range**, а значение номера старшего разряда lfsr – атрибута оператора **high**. Вычисление значения сигнала в линии обратной связи можно произвести посредством оператора **for** в процессе PFB.

Представленный пример параметризированного описания является синтезируемым, и при значении $ALPHA := "1001"$ результатом RTL-синтеза является схема, изображенная на рис. 2. В общем случае для возможности использования различных ГПП в HDL-описаниях проектировщику достаточно определить компоненту LFSRn с указанием одного параметра ALPHA.

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity LFSRn is
   generic (
         ALPHA : std_logic_vector := "1011" );
   port ( CLK     : in  std_logic;
         RST     : in  std_logic;
         Load    : in  std_logic;
         D       : in  std_logic;
         Q       : out std_logic );
end LFSRn;

architecture behavioral of LFSRn is
signal feedback : std_logic;
signal lfsr     : std_logic_vector( ALPHA'range );
begin

PSHIFT: process( CLK, RST, lfsr )
variable init : std_logic_vector( ALPHA'range );
begin
         init    := ( others => '0' );
         init( 0 ) := '1';
         if ( RST = '1' ) then
                  lfsr <= init;
         elsif rising_edge( CLK ) then
                  lfsr <= feedback & lfsr( 0 to ALPHA'high-1 );
         end if;
end process;
```

```
PFB: process( Load, D, lfsr )
variable fb : std_logic;
begin
         fb := '0';
         if ( Load = '1' ) then
                  feedback  <= D;
         else
                  for i in ALPHA'range loop
                           if ( ALPHA( i ) = '1' ) then
                                    fb := fb xor lfsr( i );
                           end if;
                  end loop;
                  feedback <= fb;
         end if;
end process;

Q <= lfsr( ALPHA'high );

end behavioral;
```

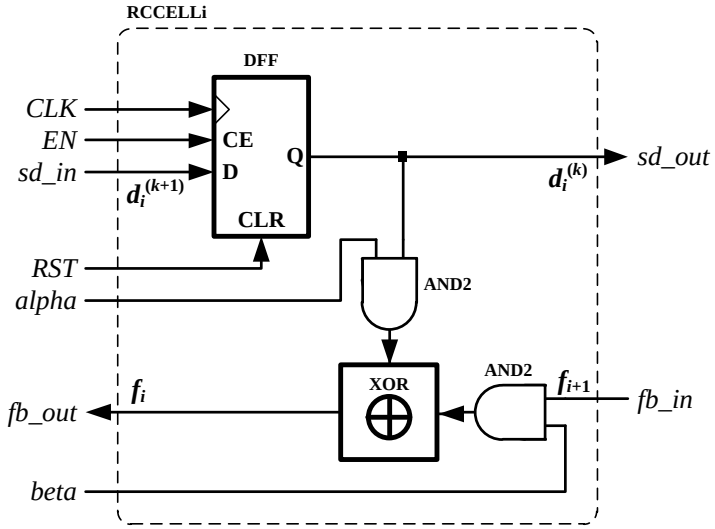
Рис. 3. Параметризированное описание ГПП

Структура конфигурируемого сдвигового регистра должна иметь возможность определять его разрядность в пределах *n* и значения коэффициентов α_i непосредственно в процессе функционирования. Для этого согласно выражению (10) модифицируем разряды LFSR, представленные *D*-триггерами, следующим образом. Для каждого разряда LFSR общими сигналами будут: *CLK* – сигнал синхронизации, *RST* – сигнал асинхронной инициализации, *EN* – асинхронный сигнал разрешения. Каждый разряд LFSR должен иметь входной порт данных *sd_in* и выходной порт данных *sd_out* для возможности обеспечения микрооперации сдвига. Для каждого разряда LFSR введем два дополнительных входных порта: *alpha* – для передачи значения коэффициента α_i и *beta* – для передачи значения сигнала, определяющего старший разряд сдвигового регистра. Для этого введем в структуру каждого элемента дополнительную аппаратуру, которая будет отвечать за конфигурацию цепи обратной связи, что потребует наличия двух дополнительных портов: входного порта сигнала обратной связи *fb_in* и выходного *fb_out*.

С учетом описанных модификаций структурная схема одного разряда конфигурируемого сдвигового регистра может выглядеть следующим образом (рис. 4).

Конфигурация разряда RCCELLi осуществляется посредством задания значений сигналов *alpha* и *beta*.

Например, в случае $alpha = '0'$ и $beta = '0'$ разряд играет роль элемента памяти и может быть использован для конфигурации линейного сдвигового регистра без обратной связи. При условии $alpha = '0'$ и $beta = '1'$ значение, хранимое на триггере, не участвует в формировании обратной связи, а значение сигнала на входе *fb_in* транслируется на выходной порт *fb_out*. Условие $alpha = '1'$ и $beta = '0'$ может быть использовано для конфигурации старшего разряда LFSR, значение которого непосредственно передается по линии обратной связи. Четвертое условие $alpha = '1'$ и $beta = '1'$ конфигурирует разряд LFSR как разряд, значение которого участвует в формировании сигнала обратной связи.




```
...
PDFF: process( CLK, RST, EN, sd_in )
begin
        if ( RST = '1' ) then
                dff <= '0';
        elsif rising_edge( CLK ) then
                if ( EN = '1' ) then
                        dff <= sd_in;
                end if;
        end if;
end process;

sd_out <= dff;

and0  <= dff    when ( alpha = '1' ) else '0';
and1  <= fb_in when ( beta  = '1' ) else '0';

fb_out <= and0  xor and1;
...
```

б)

Рис. 4. Конфигурируемый LFSR:
а) структура одного разряда; *б)* VHDL-описание

На основе имеющейся компоненты RCCELLi можно сформировать структуру *n*-разрядного конфигурируемого сдвигового регистра (рис. 5).

Младший разряд RCCELL0 дополнен двухвходовым мультиплексором, двумя элементами XOR и одним элементом AND для возможности изменения значения сигнала обратной связи f_1. Дополнительные схемы мультиплексора и двух демультиплексоров необходимы для управления процессом последовательной загрузки инициализирующих данных в триггеры элементов RCCELLi либо в память конфигурации в зависимости от значения сигнала на входе *ADR*.

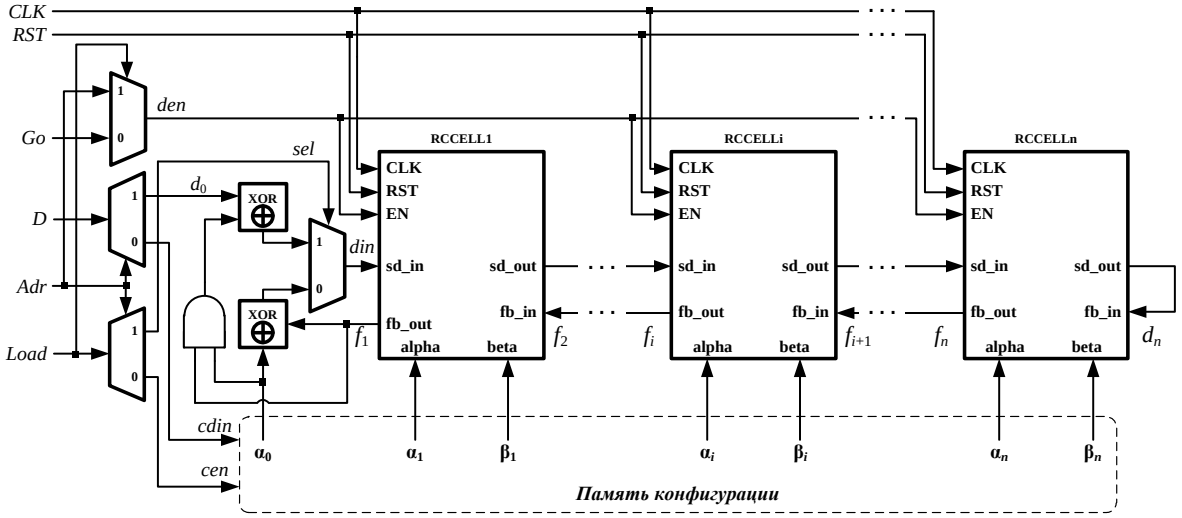


Рис. 5. Структура *n*-разрядного конфигурируемого сдвигового регистра

Так, последовательная загрузка данных из порта *D* в триггеры LFSR ($din = D$) осуществляется при выполнении следующих условий: $Adr = '1'$, $Load = '1'$ и $\alpha_0 = '0'$. Полный перечень возможных конфигураций представленной схемы приведен в табл. 1.

Таблица 1

Возможные варианты конфигураций

Adr	*Load*	*Go*	α_0	*den*	*cen*	*sel*	*din*	*cdin*	Пояснение
X	0	0	X	0	0	0	X	X	режим регистра хранения
X	0	1	0	1	0	0	f_1	X	режим ГПП сдвигового регистра
X	0	1	1	1	0	0	$\overline{f_1}$	X	режим ГПП счетчика Джонсона
0	1	X	X	0	1	0	X	*D*	режим инициализации памяти конфигурации
1	1	X	0	1	0	1	d_0	0	режим инициализации LFSR
1	1	X	1	1	0	1	$d_0 \oplus f_1$	0	режим одноканального сигнатурного анализатора

Память конфигурации, хранящая значения коэффициентов $\{\alpha_0, ..., \alpha_n\}$ и $\{\beta_1, ..., \beta_n\}$, может представлять собой набор из двух сдвиговых регистров, значения которых, как и значения LFSR, могут загружаться из общего входного порта *D*. Предположим, что значения $\{\alpha_0, ..., \alpha_n\}$ хранятся в $(n+1)$-разрядном сдвиговом регистре reg_alpha, а $\{\beta_1, ..., \beta_n\}$ – в n-разрядном сдвиговом регистре reg_beta. Покажем, что для определения значения reg_beta достаточно последовательного определения значений разрядов регистра reg_alpha. Во время процедуры инициализации ($RST = '1'$) оба регистра принимают нулевые значения. После инициализации значения коэффициентов порождающего полинома $\varphi(x)$ ($\deg(\varphi(x)) = k$) последовательно записываются в регистр reg_alpha, начиная со старшего значимого $\alpha_k = '1'$. После k тактов сигнала синхронизации в регистр записывается значение младшего коэффициента $\alpha_0 = '0'$ для возможности конфигурации структуры ГПП. Таким образом, по прошествии $k+1$ тактов синхронизации содержимое регистра reg_alpha равно $\{0, \alpha_1, \alpha_2, ..., \alpha_{k-1}, 1, 0, ..., 0\}$, что эквивалентно порождающему полиному вида $\varphi(x) = 1 \oplus \alpha_1 x \oplus \alpha_2 x^2 \oplus ... \oplus \alpha_{k-1} x^{k-1} \oplus x^k$. Для корректной конфигурации структуры LFSR значение регистра reg_beta должно принимать следующее значение: $\{\beta_1, ..., \beta_{k-1}, \beta_k, \beta_{k+1}, ..., \beta_n\} = \{1, ..., 1, 0, 0, ..., 0\}$. Видно, что единичные значения $k-1$ младших разрядов регистра reg_beta могут устанавливаться параллельно с приемом в регистр reg_alpha очередного значения коэффициента α_i без учета $\alpha_k = '1'$, которое может служить индикатором начала процесса заполнения регистра reg_beta.

Исходя из сказанного выше, память конфигурации можно спроектировать в виде схемы, изображенной на рис. 6.

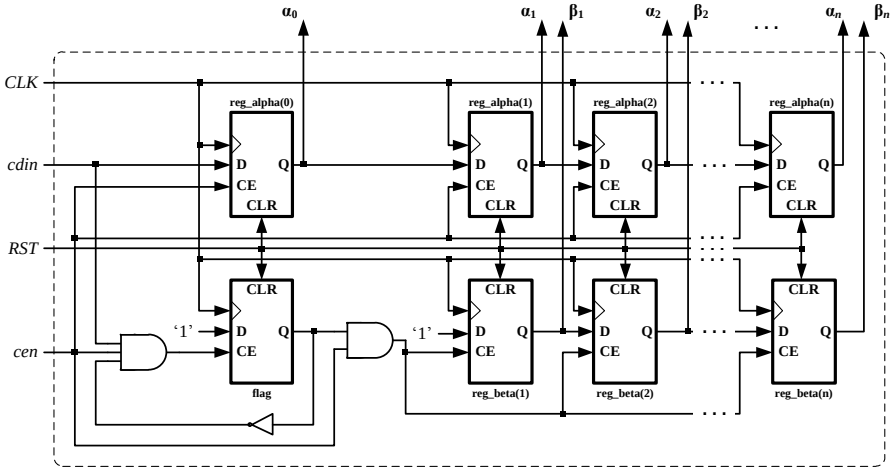


Рис. 6. Структура памяти конфигурации

В табл. 2 приведены некоторые значения регистров reg_alpha и reg_beta и соответствующие конфигурации предложенной схемы LFSR для $n=8$.

Таблица 2

Варианты конфигурирования LFSR для $n=8$

Регистр	Номер разряда регистра									Пояснения	
	0	1	2	3	4	5	6	7	8		
reg_alpha	0	0	0	0	0	0	0	0	1	8-разрядный циклический сдвиговый регистр	$d_1^{(k+1)}=d_8^{(k)}$
reg_beta	-	1	1	1	1	1	1	1	0		
reg_alpha	1	0	0	0	1	0	0	0	0	4-разрядный счетчик Джонсона	$d_1^{(k+1)}=\overline{d_4^{(k)}}$
reg_beta	-	1	1	1	0	0	0	0	0		
reg_alpha	0	1	0	0	0	1	1	0	1	8-разрядный ГПП, $\varphi(x)=1\oplus x\oplus x^5\oplus x^6\oplus x^8$	$d_1^{(k+1)}=d_1^{(k)}\oplus d_5^{(k)}\oplus d_6^{(k)}\oplus d_8^{(k)}$
reg_beta	-	1	1	1	1	1	1	1	0		
reg_alpha	1	1	0	0	0	0	1	0	0	6-разрядный ОСА, $\varphi(x)=1\oplus x\oplus x^6$	$d_1^{(k+1)}=d_0^{(k)}\oplus d_1^{(k)}\oplus d_6^{(k)}$
reg_beta	-	1	1	1	1	1	0	0	0		

Для определения конфигурации представленного LFSR необходимо выполнить следующую последовательность действий.

1. Осуществить инициализацию LFSR посредством подачи на вход *RST* сигнала с единичным логическим значением.

2. Задать начальное состояние регистра данных путем установки следующих значений на входных портах: $Adr='1'$, $Load='1'$, для каждого фронта сигнала синхронизации на входе *CLK* формировать бит инициализации на входном порту *D*. Процесс инициализации может занимать от 1 до n периодов сигнала синхронизации.

3. Задать значение памяти конфигурации путем установки следующих значений на входных портах: $Adr='0'$, $Load='1'$, для каждого фронта сигнала синхронизации на входе *CLK* формировать бит конфигурации на входном порту *D*. Процесс конфигурации потребует $n+1$ периодов сигнала синхронизации.

4. В зависимости от заданного значения конфигурации (см. табл. 1) обеспечить функционирование LSFR путем установки соответствующих сигналов на входных портах *Go*, *ADR* и *Load*.

Представленную структуру конфигурируемого LFSR можно описать, используя смешанный стиль языка VHDL, который подразумевает использование структурного и поведенческого подмножества языка в одном проекте.

3. Анализ результатов аппаратной реализации

Функциональное моделирование составленного VHDL-описания конфигурируемого LFSR производилось при помощи программного средства ISim Simulator, входящего в состав пакета проектирования цифровых устройств Xilinx ISE [16]. Процесс технологического синтеза для различных параметров *n* осуществлялся для кристаллов FPGA Xilinx Spartan-3E XC3S500E [17]. С точки зрения RTL-представления схемная реализация конфигурируемого LFSR разрядности *n* требует наличия следующих компонентов: $3n+2$ триггеров *D*-типа, $n+2$ двухвходовых элементов XOR, $2n+4$ двухвходовых элементов AND, двух мультиплексоров с конфигурацией 2×1 и двух демультиплексоров с конфигурацией 1×2.

Интегральная схема XC3S500E имеет в своем составе 1164 CLB-блока, каждый из которых содержит четыре Slice-блока: два SliceM- и два SliceL-блока. В свою очередь, каждый из перечисленных Slice-блоков состоит из двух генераторов функций LUT, способных реализовывать произвольную переключательную функцию от четырех переменных, и из двух элементов памяти, которые могут быть настроены в качестве синхронных триггеров D-типа. Таким образом, для выбранного кристалла FPGA имеется 4656 Slice-блоков, содержащих 9312 триггеров (DFF) и 9312 LUT-блоков.

Оценка аппаратурных затрат на реализацию конфигурируемого LFSR разрядности *n* изображена на рис. 7.

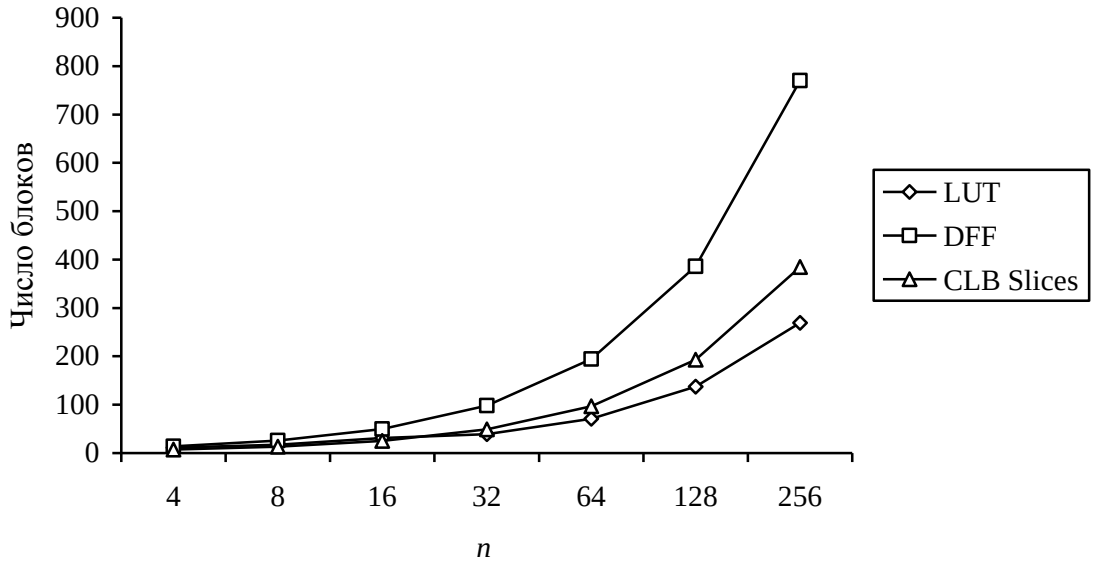


Рис. 7. График зависимости числа технологических блоков от параметра *n*

Так, при значении $n = 256$ для реализации конфигурируемого LFSR потребуется 385 Slice-блоков либо 269 LUT-блоков и 770 триггеров, что в совокупности составляет 8,27 % от общих ресурсов кристалла XC3S500E. В случае реализации 256-разрядного ГПП, представленного параметризированным описанием (см. рис. 4), аппаратурные затраты составят 256 триггеров и 2 LUT-блока либо 147 Slice-блоков, что в 2,6 раза меньше по сравнению с конфигурируемым LFSR. Однако в отличие от стандартной схемной реализации предложенный конфигурируемый LFSR способен изменять свое функционирование путем задания нового значения памяти конфигурации непосредственно в рабочем режиме.

Представленная методика проектирования конфигурируемого генератора может быть применена для синтеза LFSR с внутренними сумматорами по модулю два и схем многоканальных сигнатурных анализаторов.

Заключение

В статье предложена схемная реализация цифрового конфигурируемого генератора псевдослучайных последовательностей, позволяющая реализовывать различные режимы функционирования регистра хранения, сдвигового регистра, счетчика Джонсона, генератора М-последовательности, одноканального сигнатурного анализатора. Задание конкретного режима осуществляется путем последовательной инициализации памяти конфигурации представленной схемы. Конфигурируемый генератор может быть применен в приложениях, требующих использования различных псевдослучайных последовательностей, например при реализации средств встроенного самотестирования цифровых устройств.

Список литературы

1. VLSI Test Principles and Architecture / L.-T. Wang [et al.]. – San Francisco : Morgan Kaufman, 2006. – 777 p.

2. Ярмолик, В.Н. Контроль и диагностика цифровых узлов ЭВМ / В.Н. Ярмолик. – Минск : Наука и техника, 1988. – 240 с.

3. Gentle, J.E. Random Number Generation and Monte Carlo Methods / J.E. Gentle. – N. Y. : Springer-Verlag, 2003. – 281 p.

4. Stream Ciphers and Number Theory / T.W. Cusick [et al.]. – Amsterdam : Elsevier, 2004. – 413 p.

5. Moon, T.K. Error Correction Coding: Mathematical Methods and Algorithms / T.K. Moon. – New Jersey : John Wiley & Sons, 2005. – 756 p.

6. Wesolowski, K. Introduction to Digital Communication Systems / K. Wesolowski. – Chichester : John Wiley & Sons, 2009. – 561 p.

7. Shparlinski, I.E. Finite Fields: Theory and Computation / I.E. Shparlinski. – Dordrecht : Kluwer Academic Publishers, 1999. – 525 p.

8. Уэйкерли, Дж. Проектирование цифровых устройств: в 2 т. / Дж. Уэйкерли. – М. : Постмаркет, 2002. – Т. 2. – 528 с.

9. Konemann, B. Built-In Logic Block Observation Techniques / B. Konemann, J. Mucha, G. Zwiehoff // International Test Conference (ITC'79) : Proc. on IEEE Int. Conf. – New Jersey, USA, 1979. – P. 37–42.

10. Bolling, R. Reconfigurable Linear Feedback Register Design, Analysis and Applications / R. Bolling, S.A. Al-Arian // Circuits and Systems (ISCAS'94) : Proc. on IEEE Int. Symp. – London, England, UK, 1994. – Vol. 4. – P. 87–90.

11. A Reconfigurable Linear Feedback Shift Register (LFSR) for the Bluetooth System / P. Kitos [et al.] // Electronics, Circuits and Systems (ICECS'01) : Proc. On IEEE Int. Conf. – Malta, 2001. – P. 991–994.

12. Alaus, L. A Reconfigurable Linear FeedBack Shift Register Operator for Software Defined Radio Terminal / L. Alaus, D. Noguet, J. Palicot // Wireless Pervasive Computing (ISWPC'2008) : Proc. of Int. Symp. – Santorini, Greece, 2008. – P. 319–323.

13. Zhiyuan, W. A Kind of Reconfigurable Linear Feedback Register Design / W. Zhiyuan, H. Jianhua, G. Ziming // Information Technology and Applications (IFITA'2009) : Proc. of Int. Forum. – Chengdu, China, 2009. – P. 657–660.

14. Мурашко, И.А. Автоматизированное проектирование генераторов псевдослучайных последовательностей с использованием аппарата клеточных автоматов / И.А. Мурашко, Д.Е. Храбров // Информационные технологии и системы 2012 (ИТС 2012) : материалы Междунар. науч. конф., Минск, Беларусь, 24 октября 2012 г. – Минск : БГУИР, 2012. – С. 188–189.

15. Chu, P.P. RTL Hardware Design Using VHDL / P.P. Chu. – New Jersey : John Wiley & Sons, 2006. – 669 p.

16. ISE Design Suite: Logic Edition [Electronic resource]. – Xilinx Inc., 2012. – Mode of access : http://www.xilinx.com/products/design-tools/ise-design-suite/logic-edition.htm. – Date of access : 28.12.2012.

17. Spartan-3E FPGA Family Data Sheet [Electronic resource]. – Xilinx Inc., 2006. – Mode of access : http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf. – Date of access : 28.12.2012.

Поступила 18.01.13

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

A.A. Ivaniuk

DESIGNING CONFIGURABLE SHIFT REGISTER WITH A LINEAR FEEDBACK

A method for designing a configurable shift register is considered, which allows setting its capacity in various operation modes. The suggested shift register with a linear feedback can be used as a cyclic shift register, a generator of M-sequences, Johnson's counter and a single-channel signature analyzer. An assessment of the relevant hardware implementation expenses is given.

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.912

Л.В. Степура

АВТОМАТИЧЕСКОЕ РЕФЕРИРОВАНИЕ ТЕКСТОВОЙ ИНФОРМАЦИИ НА ОСНОВЕ МОДЕЛИРОВАНИЯ СИТУАТИВНЫХ СВЯЗЕЙ МЕЖДУ ПОНЯТИЯМИ ПРЕДМЕТНОЙ ОБЛАСТИ

Рассматривается модель процессов реферирования текстовой информации на основе формализации информационных языков средствами специальной порождающей грамматики, а также понятий информативности слов и ситуативных связей между ними. В рамках реализации модели предлагается метод синтеза связных рефератов текстовых документов путем выявления информативных предложений, построения их контекста и генерации кортежей синтаксических деревьев.

Введение

Стремительный рост объемов данных, в том числе текстовой информации, ведет к спросу на системы интеллектуального анализа текста и к бурному развитию такого инструментария. Значимое место в списке систем обработки текстовой информации занимают программы автоматического реферирования и аннотирования, позволяющие многократно упрощать и ускорять процедуры обработки больших массивов текстов.

Реферат играет важную роль в системе сведений о текстовом документе: он дополняет его библиографическое описание и содержит ряд характеристик, которые дают первичное представление о содержании публикации. Известны три основных подхода к автоматическому реферированию текстовых документов: создание рефератов путем выделения и последующей «склейки» информативных фрагментов текста с использованием статистических оценок их информативности; синтез рефератов на основе лингвистической обработки документов; гибридные методы, в которых используются как статистические характеристики текста, так и результаты лингвистического анализа.

В статье предлагается метод автоматического реферирования текстовой информации на основе моделирования главных структурных и функциональных компонентов информационной системы. Рассматривается модель представления знаний о предметной области в виде ситуативной сети, т. е. графа, вершинами которого являются информативные слова, а ребрами – ситуативные связи между ними. Моделируются процессы синтаксического анализа текстовых документов, разбиения их на предложения и синтеза связных рефератов. Для целей реферирования используется база знаний, включающая обобщенное представление выходного текста в виде упорядоченного множества синтаксических шаблонов предложений, а также словари информативных словоформ и устойчивых словосочетаний.

В существующих системах поиска и аналитической обработки текстовых документов используются главным образом технологии, ориентированные на исследование структуры и статистических характеристик самих документов без привлечения дополнительной информации [1, 2]. В данной статье эти задачи решаются с использованием тематических корпусов текстов и сформированных на их основе лингвистических словарей [3, 4].

1. Информационные языки системы реферирования

При реферировании текстовой информации используются три информационных языка – входной, внутренний и выходной. Для их определения рассмотрим формальную порождающую грамматику $G = \langle V, N, I, R \rangle$, где V – непустое множество терминальных элементов (слов), $N = \{I, '\}$ – множество нетерминальных, I – начальный символ, а R – схема грамматики, т. е.

множество правил вывода вида $\alpha \to \beta$ (α и β – различные непустые цепочки в словаре $V \cup N$). Схема R грамматики G формируется по следующим правилам:

– для любого слова $a \in V$ существуют правила вывода $I \to a'$ и $a' \to a$;

– все остальные правила вывода имеют вид $a' \to a'b'$ или $a' \to b'a'$, где $a, b \in V$.

Для удобства в состав нетерминальных символов введен символ «′» (штрих). В связи с этим грамматику G будем называть *штрихграмматикой*.

1.1. Входной язык

Пусть $V_{вх}$ – словарь некоторого естественного языка, который будем называть *входным словарем*, а его элементы – *словами* входного языка. По аналогии со схемой R штрихграмматики G построим совокупность правил вывода $R_{вх}$. Тогда язык $L(G_{вх})$, порождаемый штрихграмматикой $G_{вх} = \langle V_{вх}, N, I, R_{вх} \rangle$, будем называть *входным языком*.

1.2. Внутренний язык

Обозначим через W некоторое непустое подмножество лексем входного словаря $V_{вх}$ (под лексемой в лингвистике понимают «слово в совокупности всех его словоизменительных форм»). Зафиксируем также некоторое непустое подмножество Si элементов словаря $V_{вх}$ (назовем их *семантическими признаками*). Рассмотрим множество $V_{вн}$ цепочек вида ap языка $L(G_{вх})$, где $a \in W$, $p \in Si$. Множество $V_{вн}$ будем называть *внутренним словарем*, а его элементы – *понятиями*.

Пусть имеется штрихграмматика $G_{вн} = \langle V_{вн}, N, I, R_{вн} \rangle$ и язык $L(G_{вн})$, порождаемый этой грамматикой. Язык $L(G_{вн})$ будем называть *внутренним языком* системы, а словарь $V_{вн}$ – *внутренним словарем*. Схема $R_{вн}$ грамматики $G_{вн}$ аналогична схеме $R_{вх}$ грамматики $G_{вх}$.

1.3. Выходной язык

Пусть $V_{вых}$ – некоторое непустое множество терминальных элементов (назовем его *выходным словарем*). Тогда наряду с входным $L(G_{вх})$ и внутренним $L(G_{вн})$ языками информационной системы будем рассматривать *выходной язык* $L(G_{вых})$ как язык, порождаемый штрихграмматикой $G_{вых} = \langle V_{вых}, N, I, R_{вых} \rangle$. Схема $R_{вых}$ этой грамматики формируется по аналогии со схемой $R_{вх}$ грамматики $G_{вх}$.

В конкретной реализации системы реферирования выходной язык может совпадать с входным. Возможны также случаи, когда входных и/или выходных языков несколько.

При рассмотрении положений, касающихся всех трех рассмотренных языков, индексы «вх», «вн» и «вых» будем опускать.

2. Ситуативные связи между понятиями предметной области

С целью интеллектуализации системы реферирования будем использовать модель знаний о предметной области в виде ситуативной сети, т. е. графа, вершинами которого являются информативные слова и словосочетания предметной области, а ребрами – ситуативные связи между ними.

2.1. Корпусы текстов

В ко́рпусной лингвистике под корпусом текстов понимают совокупность текстов, накопленных и размеченных по определенным принципам в зависимости от назначения. В случае отсутствия разметки эти совокупности иногда называют корпусами текстов первого порядка. Будем различать тематические и полные корпусы текстов.

Любое непустое подмножество T входного языка $L(G_{вх})$ будем называть *текстом*, если на этом подмножестве определена редукция $\prec^r = \prec \setminus \prec^2$ линейного порядка $\prec$ (транзитивного и антисимметричного бинарного отношения на множестве T, которое связанно на T, т. е. для любых $a, b \in T$ или $a \prec b$, или $b \prec a$, или $a = b$). Цепочки текста T назовем *предложениями*.

Пусть имеется некоторое непустое множество текстов (совокупность текстов по конкретной тематике). Сформируем текст Ct, объединив все множества предложений каждого из этих текстов, и назовем его тематическим корпусом текстов. Поскольку в информационной системе

представлено, как правило, несколько таких корпусов, будем обозначать их *Ctj* (*j* – номер корпуса). Объединение $Cf_i = \bigcup_{j=1}^{n_i} Ct_j$ всех тематических корпусов назовем полным корпусом текстов (*i* – номер полного корпуса текстов).

2.2. Ситуативное отношение

Рассмотрим тематические корпусы текстов Ct_j и полные корпусы $Cf_i = \bigcup_{j=1}^{n_i} Ct_j$ ($i = \overline{1, m}$; $j = \overline{1, n_i}$). Полный корпус текстов Cf_i соответствует *i*-му входному языку (например, английскому), а тематический корпус Ct_j – *j*-й предметной области для *i*-го языка (например, предметной области Mathematics, представленной текстами на английском языке).

Обозначим через W_i множество всех слов полного корпуса текстов Cf_i. Тогда отношение толерантности Θ_i (рефлексивное и симметричное бинарное отношение) на множестве W_i назовем *ситуативным отношением* в полном корпусе текстов Cf_i, если любая упорядоченная пара слов (*a*, *b*) из множества W_i является элементом отношения Θ_i тогда и только тогда, когда слова *a* и *b* из этой пары содержатся хотя бы в одном предложении корпуса Cf_i.

Обозначим через W_j множество всех слов тематического корпуса текстов Ct_j. Рассмотрим сужение Θ_j отношения Θ_i на множество W_j, т. е. $\Theta_j = \Theta_i \cap (W_j \times W_j)$. Отношение Θ_j назовем *ситуативным отношением* в тематическом корпусе текстов Ct_j.

2.3. Информативность слов и ситуативных связей между словами

Информативность $I^a_{Ct_j}$ слова *a* из тематического корпуса текстов Ct_j – это вероятность того, что слово *a* имеется в данном корпусе при условии, что оно содержится в полном корпусе текстов. При достаточно больших объемах тематического и полного корпусов текстов формула для вычисления информативности слова имеет вид [3]

$$I^a_{Ct_j} = n^a_{Ct_j} / n^a_{Cf_i}, \qquad (1)$$

где $n^a_{Ct_j}$, $n^a_{Cf_i}$ – абсолютные частоты встречаемости слова *a* (с учетом синонимии и словоизменения) в тематическом Ct_j и полном Cf_i корпусах текстов.

Понятие информативности ситуативной связи между словами определим по аналогии с понятием информативности слова.

Пусть имеются слова *a, b* входного языка. Рассмотрим следующую совокупность событий (в теоретико-вероятностном смысле):

$S^{ab}_{Ct_j}$ – извлечение случайным образом слов *a* и *b* из одного и того же предложения тематического корпуса текстов Ct_j;

$S^{ab}_{Cf_i}$ – извлечение случайным образом слов *a* и *b* из одного и того же предложения полного корпуса текстов Cf_i;

H_{Ct_j} – появление тематического корпуса текстов Ct_j.

Обозначим через $P(S^{ab}_{Ct_j} / S^{ab}_{Cf_i})$ вероятность того, что слова *a* и *b* извлечены из одного и того же предложения множества C_{ij} при условии, что они уже извлечены из одного и того же предложения полного корпуса текстов Cf_i. Эта условная вероятность вычисляется следующим образом:

$$P(S^{ab}_{Ct_j} / S^{ab}_{Cf_i}) = \frac{P(S^{ab}_{Ct_j} \cdot S^{ab}_{Cf_i})}{P(S^{ab}_{Cf_i})} = \frac{P(S^{ab}_{Ct_j}) \cdot P(S^{ab}_{Cf_i} / S^{ab}_{Ct_j})}{P(S^{ab}_{Cf_i})}.$$

Вероятность $P(S^{ab}_{Ct_j} / S^{ab}_{Cf_i})$ будем называть *информативностью ситуативной связи между словами a и b* в тематическом корпусе текстов Ct_j (или предметной области, определяемой корпусом Ct_j).

По аналогии с формулой (1) информативность $I^{ab}_{Ct_j}$ можно представить в виде

$$I^{ab}_{Ct_j} = n^{ab}_{Ct_j} / n^{ab}_{Cf_i}, \tag{2}$$

где $n^{ab}_{Ct_j}$, $n^{ab}_{Cf_i}$ – абсолютные частоты совместной встречаемости слов a и b (с учетом синонимии и словоизменения) в одном и том же предложении тематического Ct_j и полного Cf_i корпусов текстов.

Информативность I_π предложения (словосочетания) $\pi \in L(G_{\text{вх}})$ определяется по формуле [5]

$$I_\pi = \frac{I_a + I_b + \ldots}{\sqrt{I_a{}^2 + I_b{}^2 + \ldots}}, \tag{3}$$

где $I_a, I_b, \ldots$ – показатели информативности слов $a, b, \ldots$ предложения или словосочетания π.

2.4. Информативность ситуативных связей между предложениями и фрагментами текста

Пусть π и ρ – произвольные предложения или словосочетания некоторого текста T, а I^{ab}_T – информативность ситуативной связи между его словами. Тогда информативность $I^{\pi\rho}_T$ ситуативной связи между этими предложениями будем вычислять по аналогии с вычислением информативности предложений по формуле

$$I^{\pi\rho}_T = \frac{\sum\limits_{a\in\pi,\, b\in\rho} I^{ab}_T}{\sqrt{\sum\limits_{a\in\pi,\, b\in\rho} (I^{ab}_T)^2}}. \tag{4}$$

Обозначим через Sub_1 и Sub_2 фрагменты, или субтексты, текста T. Пусть по-прежнему $I^{\pi\rho}_T$ – информативность ситуативной связи между любыми предложениями π и ρ текста T. Тогда для вычисления информативности ситуативной связи между фрагментами Sub_1 и Sub_2 построим аналог предыдущей формулы:

$$I^{Sub_1, Sub_2}_T = \frac{\sum\limits_{\pi\in Sub_1,\, \rho\in Sub_2} I^{\pi\rho}_T}{\sqrt{\sum\limits_{\pi\in Sub_1,\, \rho\in Sub_2} (I^{\pi\rho}_T)^2}}. \tag{5}$$

2.5. Определение ситуативной сети предметной области

Пусть S_j – граф ситуативного отношения Θ_j в корпусе Ct_j. Пометим каждую вершину a графа S_j значением информативности $I^{a}_{Ct_j}$ этого слова (с учетом синонимии и словоизменения), а каждое ребро (a, b) – значением информативности $I^{ab}_{Ct_j}$ ситуативной связи слов a и b (также учитывая синонимию и словоизменения). Обозначим полученный граф через Net_j.

Граф Net_j назовем *ситуативной сетью предметной области*, определяемой тематическим корпусом текстов Ct_j.

3. Синтаксический анализ реферируемого текста

Предлагается метод синтаксического анализа и разбиения текста на предложения, основанный на моделировании процесса распознавания синтагм (пар синтаксически связанных слов) средствами рассмотренной выше штрихграмматики, которая обеспечивает универсальность метода для различных проективных естественных языков. Конечная цель синтаксического анализа текста при автоматическом реферировании – построение для него кортежа синтаксических деревьев и разбиение таким образом его на предложения.

3.1. Основное свойство маргинальных синтагм

Содержательно под маргинальной будем понимать синтагму, определяющий член которой не имеет в реферируемом тексте синтаксически зависимых членов, т. е. не является определяемым ни для каких других слов текста. Определим формально понятие маргинальной синтагмы.

Пусть $\alpha a \beta b \gamma$ (или $\alpha b \beta a \gamma$) – произвольная цепочка языка $L(G_{\text{вх}})$, где $\alpha, \beta, \gamma \in V^*$ (V^*– множество всех цепочек в словаре $V_{\text{вх}}$ грамматики $G_{\text{вх}}$), ab (или ba) – синтагма этой цепочки с определяемым членом a и определяющим b.

Синтагму ab (или ba) назовем *маргинальной синтагмой* цепочки $\alpha a \beta b \gamma$ (или $\alpha b \beta a \gamma$), если для любого вхождения слова c ($c \neq b$) в $\alpha a \beta b \gamma$ (или в $\alpha b \beta a \gamma$) цепочки bc и cb не являются синтагмами цепочки $\alpha a \beta b \gamma$ (или $\alpha b \beta a \gamma$). Слово b синтагм ab и ba будем называть *маргинальным словом* синтагм αb и ba.

Пусть ab – синтагма некоторой цепочки языка $L(G)$. Тогда будем говорить, что синтаксическая связь направлена от слова a к слову b, если $(a, b) \in \Omega_\pi$. Если же $(b, a) \in \Omega_\pi$, то у такой связи противоположное направление. Для краткости направление синтаксической связи между словами будем обозначать стрелкой с началом над определяемым членом синтагмы и концом над определяющим (например, $\alpha \overrightarrow{a \beta b} \gamma$, $\alpha \overleftarrow{a \beta b} \gamma$).

Следующее утверждение является теоретической основой для построения алгоритма синтаксического анализа текста при автоматическом реферировании.

У т в е р ж д е н и е 1. *Если* $\rho \in L(G_{\text{вх}})$, *а* ab (*или* ba) – *маргинальная синтагма цепочки* ρ, *причем в схеме* $R_{\text{вх}}$ *грамматики* $G_{\text{вх}}$ *имеется правило вывода* $a' \to a'b'$ (*или* $a' \to b'a'$), *то цепочка* σ, *полученная из* ρ *удалением определяющего члена* b *синтагмы* ab (*или* ba), *является цепочкой языка* $L(G_{\text{вх}})$.

Д о к а з а т е л ь с т в о . Поскольку ab – синтагма цепочки ρ, то для цепочки ρ существует вывод $W = (I, \alpha, \beta, \ldots, \gamma, \mu a' \nu, \mu a'b' \nu, \ldots, \mu ab \nu, \ldots, \rho)$ в грамматике G, где $\alpha, \beta, \gamma, \mu, \nu \in V^*$. Так как ab – маргинальная синтагма цепочки ρ, то в силу определения маргинальной синтагмы для любого слова c цепочки ρ цепочка bc не является синтагмой, т. е. при выводе цепочки ρ не используются правила типа $b' \to b'c'$, а цепочка $\mu a'b' \nu$ в выводе W получена из цепочки $\mu a' \nu$ применением правила вывода $a' \to a'b'$. Если цепочку $\mu a'b' \nu$ исключить из вывода W, то получим вывод цепочки σ из начального символа I. Аналогично рассматривается случай, когда синтагмой цепочки ρ является цепочка ba. ■

3.2. Поиск маргинальных синтагм в реферируемом тексте

Согласно утверждению 1 процесс синтаксического анализа текста и разбиения его на предложения может быть реализован следующим образом:

– в исходном тексте ищутся маргинальные синтагмы;

– строятся синтаксические деревья найденных синтагм;

– исключаются определяющие члены найденных синтагм. В результате их исключения в полученном тексте могут появиться новые маргинальные синтагмы;

– далее процесс повторяется до тех пор, когда поиск новых маргинальных синтагм окажется безрезультатным.

Докажем ряд утверждений, которые будут способствовать нахождению в тексте маргинальных синтагм.

Пусть по-прежнему $\pi = a_1 a_2 \ldots a_n$. Исследуем условия существования маргинальных синтагм цепочки π в синтагматических структурах следующих четырех типов:

1) $\overleftarrow{a_1 a_2}$, $\overrightarrow{a_{n-1} a_n}$;

2) $\overrightarrow{a_i \overleftarrow{a_{i+1} a_{i+2}}}$, $\overleftarrow{a_i \overleftarrow{a_{i+1} a_{i+2}}}$ $(i = \overline{1, n-2})$; $\overrightarrow{a_i \ldots \overrightarrow{a_j a_{j+1}} a_{j+2}}$, $\overleftarrow{a_i \ldots \overrightarrow{a_j a_{j+1}} a_{j+2}}$ $(i = \overline{1, j-1},\ j = \overline{2, n-2})$;

3) $\overrightarrow{\overrightarrow{a_i a_{i+1}}\, a_{i+2}}$, $\overleftarrow{\overrightarrow{a_i a_{i+1}}\, a_{i+2}}$ $(i = \overline{1, n-2})$; $\overrightarrow{\overrightarrow{a_i a_{i+1}} a_{i+2} \ldots a_j}$, $\overleftarrow{\overrightarrow{a_i a_{i+1}} a_{i+2} \ldots a_j}$ $(i = \overline{1, j-3},\ j = \overline{4, n})$;

4) $\overrightarrow{\overrightarrow{a_i a_{i+1}} \overleftarrow{a_{i+2} a_{i+3}}}$, $\overleftarrow{\overrightarrow{a_i a_{i+1}} \overleftarrow{a_{i+2} a_{i+3}}}$ $(i = \overline{1, n-3})$; $\overrightarrow{a_i \ldots \overrightarrow{a_j a_{j+1}} \overleftarrow{a_{j+2} a_{j+3}} \ldots a_k}$, $\overleftarrow{a_i \ldots \overrightarrow{a_j a_{j+1}} \overleftarrow{a_{j+2} a_{j+3}} \ldots a_k}$ $(i = \overline{1, j-1},\ j = \overline{2, k-4},\ k = \overline{6, n})$.

Утверждение 2. *Синтагмы* $\overleftarrow{a_1 a_2}$, $\overrightarrow{a_{n-1} a_n}$ *являются маргинальными синтагмами цепочки* π.

Доказательство. Пусть от противного цепочка $a_{n-1} a_n$ не является маргинальной синтагмой, т. е. в цепочке π имеется слово c, которое служит определяемым членом синтагмы ca_n. Тогда в схеме R грамматики G должны существовать правила, обеспечивающие вывод цепочки $c' \ldots a'_{n-1} a'_n$ из цепочки $a'_{n-1} a'_n$, что противоречит определению штрихграмматики G. Аналогично доказывается, что синтагма $\overleftarrow{a_1 a_2}$ является маргинальной. ■

По аналогии с доказательством утверждения 2 доказываются утверждения 3–5.

Утверждение 3. *Синтагматические структуры* $\overrightarrow{a_i \overleftarrow{a_{i+1} a_{i+2}}}$, $\overleftarrow{a_i \overleftarrow{a_{i+1} a_{i+2}}}$ *цепочки* π *содержат маргинальную синтагму* $\overleftarrow{a_{i+1} a_{i+2}}$.

Синтагматические структуры $\overrightarrow{a_i \ldots \overrightarrow{a_j a_{j+1}} a_{j+2}}$, $\overleftarrow{a_i \ldots \overrightarrow{a_j a_{j+1}} a_{j+2}}$ *цепочки* π *содержат маргинальную синтагму* $\overrightarrow{a_j a_{j+1}}$.

Утверждение 4. *Синтагматические структуры* $\overrightarrow{\overrightarrow{a_i a_{i+1}}\, a_{i+2}}$, $\overleftarrow{\overrightarrow{a_i a_{i+1}}\, a_{i+2}}$ *цепочки* π *содержат маргинальную синтагму* $\overrightarrow{a_i a_{i+1}}$.

Синтагматические структуры $\overrightarrow{\overrightarrow{a_i a_{i+1}} a_{i+2} \ldots a_j}$, $\overleftarrow{\overrightarrow{a_i a_{i+1}} a_{i+2} \ldots a_j}$ *цепочки* π *содержат маргинальную синтагму* $\overrightarrow{a_i a_{i+1}}$.

Утверждение 5. *Синтагматические структуры* $\overrightarrow{\overrightarrow{a_i a_{i+1}} \overleftarrow{a_{i+2} a_{i+3}}}$, $\overleftarrow{\overrightarrow{a_i a_{i+1}} \overleftarrow{a_{i+2} a_{i+3}}}$ *цепочки* π *содержат маргинальные синтагмы* $\overrightarrow{a_i a_{i+1}}$, $\overleftarrow{a_{i+2} a_{i+3}}$.

Синтагматические структуры $\overrightarrow{a_i \ldots \overrightarrow{a_j a_{j+1}} \overleftarrow{a_{j+2} a_{j+3}} \ldots a_k}$, $\overleftarrow{a_i \ldots \overrightarrow{a_j a_{j+1}} \overleftarrow{a_{j+2} a_{j+3}} \ldots a_k}$ *цепочки* π *содержат маргинальные синтагмы* $\overrightarrow{a_j a_{j+1}}$, $\overleftarrow{a_{j+2} a_{j+3}}$.

Утверждения 3–5 позволяют найти маргинальные синтагмы в анализируемом тексте. Процесс их поиска реализуется следующим образом. В цепочке $\pi = a_1 a_2 \ldots a_n$ ищутся маргинальные синтагмы типа $\overleftarrow{a_1 a_2}$ и $\overrightarrow{a_{n-1} a_n}$ и исключаются из π их определяющие члены. Процесс поиска таких синтагм и исключения определяющих членов повторяется до тех пор, пока синтагмы указанного типа будут присутствовать в цепочке π. Далее аналогичная процедура повторяется для синтагматических структур типа $\overrightarrow{a_i \overleftarrow{a_{i+1} a_{i+2}}}$, $\overleftarrow{a_i \overleftarrow{a_{i+1} a_{i+2}}}$ и $\overrightarrow{\overrightarrow{a_i a_{i+1}}\, a_{i+2}}$, $\overleftarrow{\overrightarrow{a_i a_{i+1}}\, a_{i+2}}$, затем для структур $\overrightarrow{a_i \ldots \overrightarrow{a_j a_{j+1}} a_{j+2}}$, $\overleftarrow{a_i \ldots \overrightarrow{a_j a_{j+1}} a_{j+2}}$ и т. д.

Пример 1. Рассмотрим процесс синтаксического анализа текста и разбиения его на предложения на примере цепочки «Данный проект реализуется поэтапно первый этап запланирован на этот год окончание работ в следующем году».

На первом шаге анализа в исходной цепочке выявляются неразделенные синтагмы:

Данный проект $\overleftarrow{\text{реализуется поэтапно}}$ *первый этап* $\overleftarrow{\text{запланирован на этот}}$ *год*

$\overleftarrow{\text{окончание работ в следующем}}$ *году*

На втором шаге из полученной цепочки исключаются определяющие члены маргинальных синтагм:

проект реализуется этап запланирован год окончание работ в году

Процесс последовательного выявления маргинальных синтагм и исключения из них определяющих членов продолжается аналогичным образом на третьем, четвертом, пятом и шестом шагах синтаксического анализа:

проект реализуется этап запланирован год окончание в году

проект реализуется этап запланирован год окончание в году

проект реализуется этап запланирован год окончание в

проект реализуется этап запланирован год окончание

В результате получим три синтаксических дерева анализируемого текста (рис. 1).

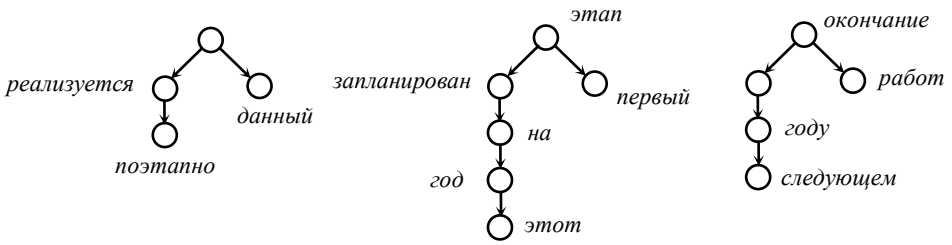


Рис. 1. Синтаксические деревья текста

Полученным синтаксическим деревьям соответствует разбиение исходного текста на три предложения: данный проект реализуется поэтапно, первый этап запланирован на этот год, окончание работ в следующем году.

4. Выявление контекста информативных предложений

Понятие контекста предложения тесным образом связано с понятием сверхфразового единства, т. е. кортежа предложений единой тематической направленности. При построении в тексте сверхфразовых единств вначале будем использовать ситуативные связи между словами текста, а затем между его предложениями. Ситуативные связи между предложениями текста представим в виде графа ситуативных связей. Введем предварительно понятие графа информативности текста.

4.1. Граф информативности текста

Пусть имеется текст (т. е. кортеж предложений) $T = \langle \pi_1, \pi_2, \ldots \rangle$. Вычислим информативность всех предложений текста T и исключим из T неинформативные предложения, т. е. все предложения π, информативность I_π которых меньше некоторого I_0. В результате получим кортеж предложений $T_{\text{инф}} = \langle \pi_{i_1}, \pi_{i_2}, \ldots \rangle$, который будем называть *маршрутом информативности* текста T. Соединив последовательно вершины графа текста G_T (т. е. графа редукции линейного порядка на множестве всех предложений текста T), соответствующие информативным предложениям, получим орграф $G_{\text{инф.}}$, который будем называть *графом информативности* текста T (рис. 2). Вершины и дуги графа текста, не вошедшие в состав графа информативности $G_{\text{инф}}$, изображены пунктирными линиями.

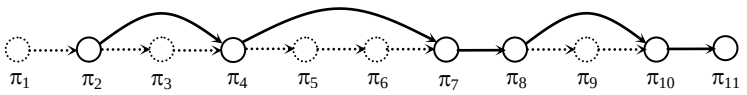


Рис. 2. Пример графа информативности текста

4.2. Граф ситуативных связей между предложениями текста

Пусть T^+ – множество всех информативных, а T^- – всех неинформативных предложений текста T ($T = T^+ \cup T^-$). Определим на паре множеств T^+, T^- симметричное отношение Ξ, такое, что для любых предложений $\pi \in T^+$ и $\rho \in T^-$ $(\pi, \rho) \in \Xi$ тогда и только тогда, когда информативность $I_{\pi\rho}$ ситуативной связи между предложениями π и ρ не меньше некоторого значения. Граф отношения Ξ назовем *графом ситуативных связей* между предложениями текста T (рис. 3). Информативные предложения изображены на рис. 3 сплошными линиями, а неинформативные – пунктирными. Пунктирными стрелками представлены дуги графа текста, а скобками объединены возможные кандидаты в сверхфразовые единства.

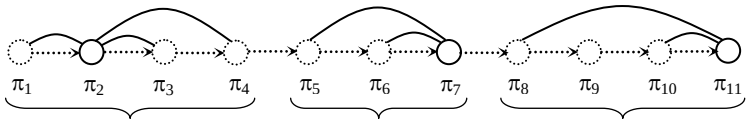


Рис. 3. Пример графа ситуативных связей между предложениями текста

Процесс разбиения текста на сверхфразовые единства осуществляется в два этапа. На первом этапе в тексте выявляются информативные предложения, т. е. строится маршрут информативности $T_{\text{инф.}}$. На втором этапе выявляются ситуативные связи между предложениями текста, на основе которых формируются сверхфразовые единства.

5. Синтез связного реферата

Для целей синтеза связного реферата будем использовать специальную базу знаний, включающую обобщенное представление выходного текста в виде упорядоченного множества синтаксических шаблонов предложений, а также словари информативных словоформ и устойчивых словосочетаний. Задачу синтеза будем решать в два этапа: на первом этапе сформируем кортеж синтаксических деревьев, используя синтаксический шаблон предметной области, а на втором синтезируем предложения текста.

5.1. Синтаксические шаблоны

Пусть имеется текст $T = \langle \pi_1, \pi_2, \ldots, \pi_m \rangle$ в виде кортежа предложений $\pi_1, \pi_2, \ldots, \pi_m$. Обозначим через D_π синтаксическое дерево любого предложения π из текста T. Ордерево Dr_π, полученное из синтаксического дерева D_π заменой всех его поддеревьев, которые являются синтаксическими деревьями прагматически полных синтагматических структур (ПП-структур), слотами («пустыми» вершинами), будем называть *синтаксическим шаблоном предложения* π (рис. 4). Под ПП-структурой понимается информативная в некотором тематическом разделе предметной области (т. е. хотя бы в одном тематическом корпусе текстов) синтагматическая структура, выражаемая устойчивым словосочетанием (например, «информационные технологии», «входной язык», «радиоаппаратура»).

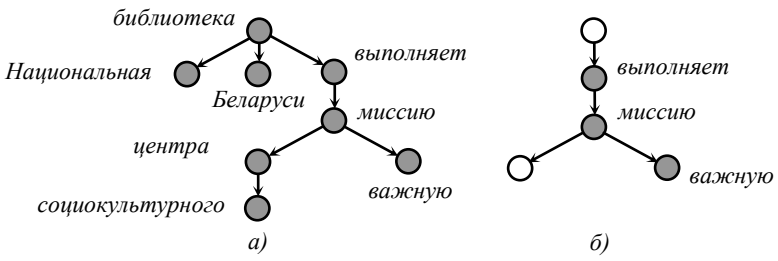


Рис. 4. Анализ предложения «Национальная библиотека Беларуси выполняет важную миссию социокультурного центра»: *а)* синтаксическое дерево; *б)* синтаксический шаблон

При синтезе синтаксического дерева предложения слоты заменяются синтаксическими деревьями синтагматических структур из графа ситуативных связей между предложениями текста.

Обозначим через D_i ($i = \overline{1, m}$) синтаксический шаблон предложения π_i. Тогда кортеж $Sh = \langle D_1, D_2, \ldots, D_m \rangle$ синтаксических шаблонов всех предложений текста $T = \langle \pi_1, \pi_2, \ldots, \pi_m \rangle$ назовем *синтаксическим шаблоном* этого текста.

5.2. Отношение дискурсивной сочетаемости

Синтаксический шаблон текста строится на основе отношения дискурсивной сочетаемости Δ, которое определим следующим образом.

Пусть имеется множество $\{Sh_i \mid i = \overline{1, n}\}$ синтаксических шаблонов некоторых текстов. Рассмотрим объединение множеств синтаксических шаблонов $Sh = \bigcup_{i=1}^{n} Sh_i$. Определим на множестве Sh антирефлексивное бинарное отношение Δ, такое, что для любых синтаксических шаблонов $D_r, D_s \in Sh$ некоторых предложений π_r и π_s соотношение $(D_r, D_s) \in \Delta$ справедливо тогда и только тогда, когда существует синтаксический шаблон текста Sh_j ($1 \le j \le n$), элементами которого являются синтаксические шаблоны D_r и D_s предложений π_r и π_s соответственно; в синтаксическом шаблоне текста Sh_j синтаксический шаблон предложения π_r непосредственно предшествует синтаксическому шаблону предложения π_s. Отношение Δ назовем *отношением дискурсивной сочетаемости* синтаксических шаблонов предложений.

Определим на множестве всех пар отношения Δ (т. е. на множестве Δ) отношение строгого порядка $\prec$ (антирефлексивное и транзитивное бинарное отношение) следующим образом: будем считать, что для любых пар синтаксических шаблонов предложений (D_i, D_j) и (D_k, D_l) отношения Δ соотношение $(D_i, D_j) \prec (D_k, D_l)$ справедливо тогда и только тогда, когда $D_j = D_k$, т. е. D_j и D_k – один и тот же синтаксический шаблон.

Множество Sh с определенным на нем отношением дискурсивной сочетаемости Δ и строгим порядком $\prec$, заданным на множестве Δ, назовем *синтаксическим шаблоном предметной области*.

Используя строгий порядок $\prec$, синтаксический шаблон реферата можно построить в два этапа: сначала в виде ориентированного маршрута в графе, вершинами которого являются упорядоченные пары синтаксических шаблонов предложений, а дуги соответствуют отношению $\prec$, затем в виде орцепи, где вершины (синтаксические шаблоны предложений) соединены дугами, определяющими порядок использования этих шаблонов при синтезе текста. Выбор каждого очередного элемента множества Δ реализуется путем сравнения синтаксического шаблона и синтаксического дерева реферата.

5.3. Синтез кортежа синтаксических деревьев

Процедура построения кортежа синтаксических деревьев синтезируемого текста реализуется в два этапа.

На первом этапе ищется минимальный (в смысле строгого порядка $\prec$) элемент множества Δ, т. е. пара синтаксических шаблонов предложений (D_1, D_2). Шаблон D_1 должен удовлетворять следующему условию: в графе ситуативных связей между предложениями текста должны существовать ПП-структуры для заполнения слотов шаблона D_1. Далее в множестве Δ ищутся пары синтаксических шаблонов предложений (D_2, D_3), (D_3, D_4), … . Шаблоны предложений $D_2, D_3, \ldots$ также должны удовлетворять упомянутому выше условию. После заполнения слотов всех найденных шаблонов (кроме поименованных) ПП-структурами из графа ситуативных связей между предложениями текста и ситуативной сети получим требуемый кортеж синтаксических деревьев с незаполненными поименованными слотами.

На втором этапе заполняются поименованные слоты сформированного кортежа синтаксических деревьев.

5.4. Упорядоченное синтаксическое дерево

Определим предварительно понятия «расстояние между словами предложения», «отношение семантической близости» и «упорядочивающие отображения».

Расстоянием $R(a_i, a_j)$ между словами a_i и a_j цепочки $a_1a_2\ldots a_i\ldots a_j\ldots a_n$ назовем модуль разности j и i, т. е. $R(a_i, a_j) = |j - i|$.

Пусть a – произвольное слово некоторого предложения ($a \in V$, V – словарь), а L – множество синтаксически корректных синтагматических структур из полного корпуса текстов. (Факт синтаксической корректности устанавливает эксперт-лингвист.) Определим на множестве $\Omega \cap (\{a\} \times V)$ бинарное отношение $\geq_a$, являющееся объединением эквивалентности $=_a$ и строгого порядка $>_a$, следующим образом. Будем считать, что для любых слов $b, c \in V$, таких, что $(a, b) \in \Omega$ и $(a, c) \in \Omega$, выполняется соотношение $(a, b) >_a (a, c)$, если в множестве L существует синтагматическая структура из слов a, b и c, такая, что $R(a, b) > R(a, c)$, и нет структуры из этих же слов, где выполняется неравенство противоположного знака. Считаем также, что $(a, b) =_a (a, c)$, если существует синтаксически корректная синтагматическая структура, где $R(a, b) = R(a, c)$, или найдутся две таких структуры, в которых соответственно $R(a, b) < R(a, c)$ и $R(a, b) > R(a, c)$. Отношение $\geq_a$ назовем *отношением семантической близости*. Если $(a, b) >_a (a, c)$, то будем говорить, что слова a и b *семантически связаны сильнее,* чем слова a и c. Если же $(a, b) =_a (a, c)$, то скажем, что слова b и c *семантически равнозначны относительно слова* a.

Для всех троек слов a, b, c типа рассмотренных выше построим совокупность отображений $\Phi_a : \Omega \cap (\{a\} \times V) \to \{1, 2, \ldots\}$, таких, что $\Phi_a(a, b) > \Phi_a(a, c)$, если $(a, b) >_a (a, c)$, а если $(a, b) =_a (a, c)$, то $\Phi_a(a, b) = \Phi_a(a, c)$. Такие отображения Φ_a назовем *упорядочивающими.*

На практике в качестве совокупности L используется полный корпус текстов, а образы всех дуг, исходящих из вершины a синтаксического дерева предложения, при упорядочивающем отображении Φ_a можно рассматривать как числовые метки на этих дугах.

Синтаксическое дерево любого предложения назовем *упорядоченным*, если все его дуги помечены натуральными числами, являющимися образами этих дуг при отображениях Φ_a.

Процесс построения упорядоченного синтаксического дерева реализуется следующим образом.

Ищется произвольная висячая вершина синтаксического дерева b_1, являющаяся конечной вершиной орцепи максимальной длины, и смежная ей вершина a.

Ищутся все дуги $(a, b_1), (a, b_2), \ldots$, исходящие из вершины a.

Выявляются натуральные числа, которые являются образами всех найденных дуг при отображениях Φ_a, и помечаются ими эти дуги.

Условно исключаются из синтаксического дерева все дуги, исходящие из вершины a, и их конечные вершины. Если в синтаксическом дереве после такого исключения имеются дуги, то процесс начинается сначала, иначе алгоритм заканчивает работу.

Пример 2. Рассмотрим процедуру синтеза предложения, упорядоченное синтаксическое дерево которого изображено на рис. 5.

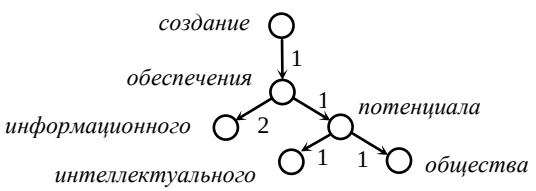


Рис. 5. Пример упорядоченного синтаксического дерева

Процедура синтеза осуществляется поэтапно (рис. 6):
1) строится синтагма «создание обеспечения»;
2) строится синтагма «информационного обеспечения» и «обеспечения потенциала»;
3) строится синтагма «потенциала общества»;
4) синтезируется синтагма «интеллектуального потенциала».

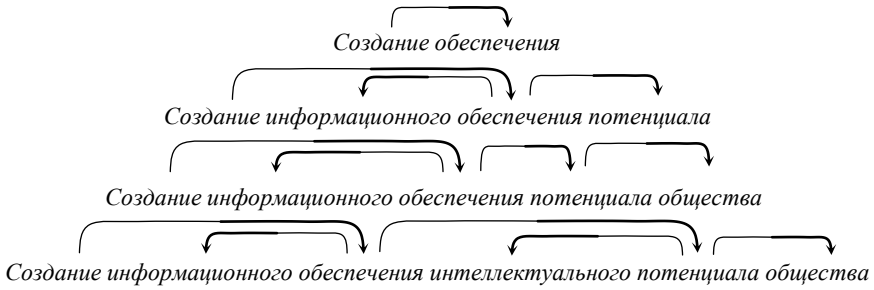


Рис. 6. Этапы синтеза предложения

В результате получаем предложение «Создание информационного обеспечения интеллектуального потенциала общества».

Заключение

Предложенная модель представления знаний о предметной области в виде ситуативной сети отличается универсальностью, т. е. независимостью от конкретного естественного языка. Адаптация системы реферирования к входному языку реализуется путем наполнения соответствующей базы знаний без изменения программного обеспечения.

Разработанный в статье метод автоматического реферирования текстовой информации обеспечивает качественное формирование рефератов с применением синтаксического анализа и разбиением текста на предложения, с выявлением контекста информативных предложений и синтеза связного реферата.

Список литературы

1. Тактаев, С. Поиск информации в компьютерных сетях: новые подходы / С. Тактаев // [Электронный ресурс]. – Режим доступа : http:// www.searchengines.ru/articles/004603.html. – Дата доступа : 30.01.2013.

2. Тарасов, С.Д. Современные методы автоматического реферирования / С.Д. Тарасов // Научно-технические ведомости СПбГПУ. – СПб. : СПбГПУ, 2010. – № 6. – С. 59–74.

3. Кравцов, А.А. Индексирование и реферирование текста на основе ситуативно-синтагматической сети / А.А. Кравцов, С.Ф. Липницкий, Л.В. Степура // Искусственный интеллект. Интеллектуальные системы (ИИ-2009) : материалы X Междунар. науч.-техн. конф. – Таганрог : Изд-во ТТИ ЮФУ, 2009. – С. 277–279.

4. Липницкий, С.Ф. Модели знаний о предметной области для решения задач поиска и обработки текстовой информации / С.Ф. Липницкий // Информатика. – 2007. – № 2. – С. 25–34.

5. Липницкий, С.Ф. Алгоритмы создания гипертекста на основе ситуативно-синтагматической сети / С.Ф Липницкий, Л.В. Степура // Весці НАН Беларусі. Сер. фіз.-тэхн. навук. – 2010. – № 3. – С. 90–95.

Поступила 12.02.13

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

L.V. Stepura

AUTOMATIC ABSTRACTING OF TEXTUAL INFORMATION BASED ON SITUATIONAL LINKS MODELING OF APPLICATION DOMAIN CONCEPTS

A model of abstracting textual information, based on the formalization of information languages by a special generative grammar and the concepts of word self-descriptiveness and contextual links between words, is considered. A method of abstracts synthesis for textual documents by identifying informative sentences, constructing their context and generating chains of syntax trees is suggested.

УДК 681.515

А.Г. Стрижнев, А.Н. Русакович

АВТОМАТИЗИРОВАННЫЙ СИНТЕЗ ЦИФРОВЫХ РЕГУЛЯТОРОВ НА ОСНОВЕ ДИСКРЕТНЫХ ПЕРЕДАТОЧНЫХ ФУНКЦИЙ ОБЪЕКТОВ УПРАВЛЕНИЯ

Рассматриваются методы дискретизации передаточных функций объектов управления, используемые в пакете MATLAB, в том числе экстраполяторы нулевого и первого порядков, билинейная аппроксимация (преобразование Тастина) и преобразование Тастина с коррекцией по частоте среза. В среде MATLAB разрабатывается программа, которая позволяет автоматизировать процесс определения дискретных передаточных функций различных объектов управления по их непрерывным моделям, а также рассчитываются цифровые регуляторы. Для объектов управления второго и третьего порядков программно определяются дискретные передаточные функции и рассчитываются цифровые регуляторы. Путем математического моделирования осуществляется проверка работы указанных объектов управления, а также систем автоматического управления с различными цифровыми регуляторами.

Введение

Проектирование систем автоматического управления (САУ) обычно начинают с изучения объектов управления (ОУ) с целью получения их математических моделей, связывающих регулируемые выходные переменные с возможными управляющими сигналами и возмущениями. Наличие математических моделей ОУ позволяет осуществить расчет цифровых регуляторов (ЦР), которые придают системе требуемые динамические свойства и широко используются в различной технике. Большинство ОУ являются аналоговыми и описываются непрерывными математическими моделями, для некоторых из ОУ непосредственно осуществлен расчет ЦР [1]. Следует отметить, что определение передаточных функций ЦР для систем с ОУ третьего и выше порядков является трудоемкой задачей и для ее разрешения требуются новые подходы. Вместе с тем достаточно просто можно осуществить расчет ЦР по дискретным передаточным функциям (ДПФ) ОУ, которые определены для многих, но не для всех ОУ [1]. Однако расчет ДПФ ОУ третьего и выше порядков также является сложной задачей. В связи с этим возникла необходимость автоматизировать процесс разработки ЦР, базирующийся на ДПФ ОУ, которая определяется с использованием пакета MATLAB.

1. Методы дискретизации передаточных функций объектов управления

Для определения ДПФ по непрерывной модели ОУ используют различные методы дискретизации [2], многие из которых поддерживаются пакетом MATLAB:

– экстраполятор нулевого порядка;

– экстраполятор первого порядка;

– билинейную аппроксимацию (преобразование Тастина);

– преобразование Тастина с коррекцией по частоте среза.

Экстраполятор нулевого порядка фиксирует значение входного сигнала $u(t)$ в начале интервала квантования h и поддерживает на выходе это значение (сигнал $u[k]$) до окончания интервала квантования. Затем выходной сигнал изменяется скачком до величины входного сигнала на следующем шаге квантования:

$$u(t) = u[k], \quad kh \le t < (k+1)h. \tag{1}$$

Экстраполятор нулевого порядка имеет импульсную переходную функцию прямоугольного вида.

Экстраполятор первого порядка восстанавливает в виде кусочно-линейной аппроксимации изначально оцифрованный сигнал. Выходной сигнал на каждом такте дискретизации изменяется в соответствии с крутизной входного сигнала на предыдущем интервале дискретизации:

$$u(t) = u[k] + \frac{t - kh}{h}(u[k+1] - u[k]), \quad kh \le t < (k+1)h. \tag{2}$$

Экстраполятор первого порядка имеет импульсную переходную функцию треугольного вида. По сравнению с экстраполятором нулевого порядка экстраполятор первого порядка в общем случае имеет меньший шум квантования и, следовательно, более точно восстанавливает сигнал [2].

Билинейная аппроксимация представляет собой функцию, аппроксимирующую натуральный логарифм, который является точным отображением *z*-плоскости на *s*-плоскость, *Z*- и *L*-изображения связаны между собой соотношением

$$z = e^{sh} \approx \frac{1 + sh/2}{1 - sh/2}.$$

Следовательно,

$$s = \frac{1}{h}\ln(z). \tag{3}$$

При разложении выражения (3) в ряд Тейлора получим [3]

$$s = \frac{2}{h}\left[\frac{z-1}{z+1} + \frac{1}{3}\left(\frac{z-1}{z+1}\right)^3 + \frac{1}{5}\left(\frac{z-1}{z+1}\right)^5 + \frac{1}{7}\left(\frac{z-1}{z+1}\right)^7 + \ldots\right] \approx \frac{2}{h}\frac{z-1}{z+1}. \tag{4}$$

Билинейная аппроксимация использует выражение (4) для замены непрерывной передаточной функции (НПФ) $G(s)$ на ее дискретный аналог $GH(z)$:

$$GH(z) = G(s)\Big|_{s=\frac{2}{h}\frac{z-1}{z+1}} = G\left(\frac{2}{h}\frac{z-1}{z+1}\right). \tag{5}$$

Преобразование Тастина с коррекцией по частоте среза – модификация билинейной аппроксимации [2]. Здесь для получения ДПФ $GH'(z)$ используется следующее выражение:

$$GH'(z) = G(s)\Big|_{s=\frac{\omega}{tg(\omega h/2)}\frac{z-1}{z+1}} = G\left(\frac{\omega}{tg(\omega h/2)}\frac{z-1}{z+1}\right), \tag{6}$$

где ω – частота среза непрерывного ОУ.

Используя различные методы дискретизации, можно получить ДПФ для ОУ различных порядков, но сделать это легко только для ОУ второго и третьего порядков. Значительно проще получить ДПФ для ОУ третьего и выше порядков с использованием пакета MATLAB и специально разработанной программы. Следует заметить, что независимо от выбранного метода дискретизации в пакете MATLAB ДПФ отображаются в виде ступенчатой характеристики.

2. Автоматизированный расчет ДПФ объектов управления и цифровых регуляторов

Для автоматизированного расчета ДПФ ОУ и ЦР была разработана специальная программа для пакета MATLAB, алгоритм работы которой показан на рис. 1. Используя выражения (1), (2), (5) и (6), программа позволяет получить ДПФ ОУ, рассчитать ЦР и осуществить математическое моделирование работы САУ и ее элементов. В ходе выполнения программы возможно построение частотных и временных характеристик непрерывных и дискретных моделей ОУ и САУ с ЦР. Следует отметить, что данная программа и алгоритм ее работы являются универсальными и применимы для ОУ любого порядка, некоторые из них будут рассмотрены в дальнейшем.

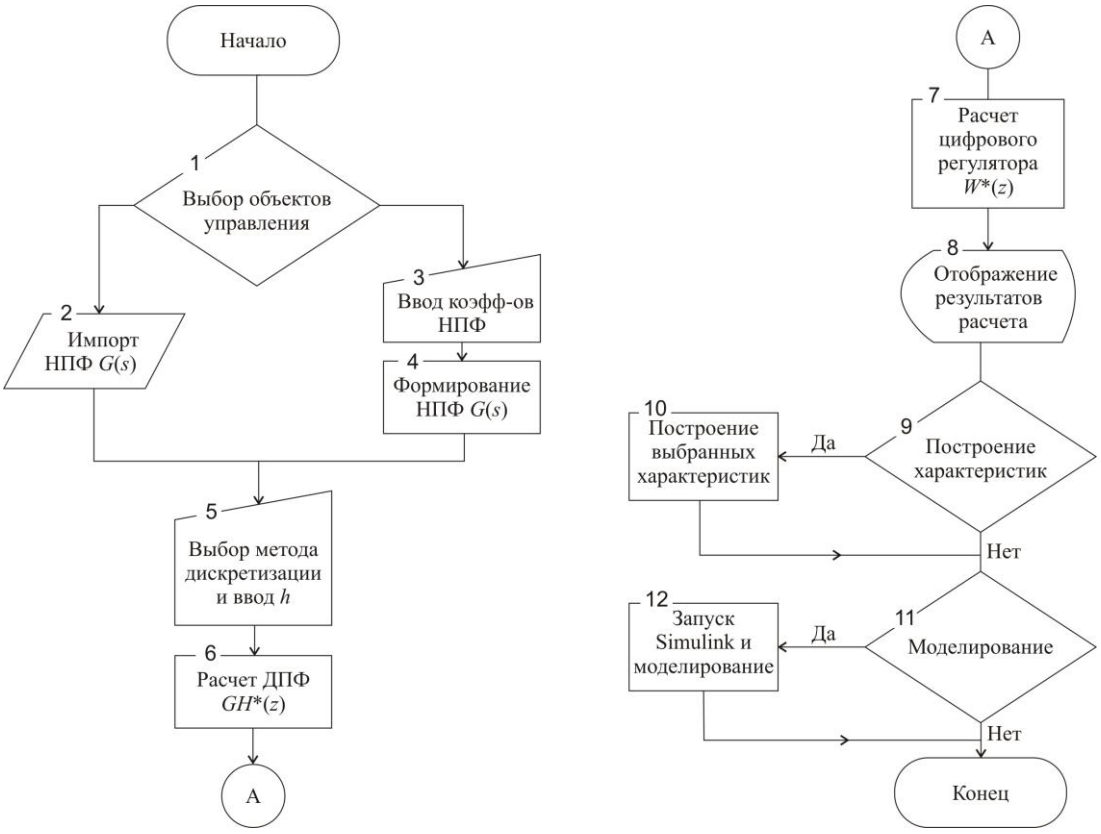


Рис 1. Алгоритм работы программы

Для запуска программы необходимо с помощью блока 1 выбрать НПФ исследуемого ОУ в виде [4]

$$G(s) = \frac{x_m s^m + x_{m-1} s^{m-1} + ... + x_0}{u_n s^n + u_{n-1} s^{n-1} + ... + u_0}, \quad (7)$$

где x_m, u_n – постоянные коэффициенты.

Выбор $G(s)$ возможен из рабочего пространства (workspace) MATLAB с помощью блока 2 или ручного ввода. В случае ручного ввода необходимо в блок 3 ввести постоянные коэффициенты (x_m, u_n), после чего блок 4 формирует НПФ ОУ требуемого вида (7), которая в дальнейшем и будет использоваться.

Для получения ДПФ ОУ необходимо выбрать метод дискретизации и ввести величину шага квантования *h* в блок 5 (рекомендации по выбору *h* приведены ниже).

Блок 6 осуществляет расчет ДПФ исследуемого ОУ с помощью функции пакета MATLAB

$$c2d(w, h, 'method'), \quad (8)$$

где w – ОУ; h – шаг квантования; $method$ – метод дискретизации.

Выбор параметра $method$ производится согласно табл. 1 [5].

Таблица 1

Методы дискретизации, поддерживаемые рабочим пространством MATLAB

Метод дискретизации	Параметр *'method'*
Экстраполятор нулевого порядка	*'zoh'*
Экстраполятор первого порядка	*'foh'*
Билинейная аппроксимация	*'tustin'*
Преобразование Тастина с коррекцией по частоте среза	*'prewarp'*

Согласно рекомендациям [1] для расчета ЦР нужно использовать ДПФ ОУ, полученную с помощью экстраполятора нулевого порядка (*'zoh'*) в виде

$$GH(z)=\frac{c_1 z^{-1}+c_2 z^{-2}+...+c_n z^{-n}}{1+d_1 z^{-1}+d_2 z^{-2}+...+d_n z^{-n}}, \tag{9}$$

где c_n, d_n – постоянные коэффициенты.

Если использовать другие методы дискретизации, то получается приближенная ДПФ ОУ

$$GH(z)=\frac{c'_0+c'_1 z^{-1}+c'_2 z^{-2}+...+c'_n z^{-n}}{1+d'_1 z^{-1}+d'_2 z^{-2}+...+d'_n z^{-n}}, \tag{10}$$

где c'_n, d'_n – постоянные коэффициенты.

Передаточная функция (10) отличается от точной передаточной функции (9) не только приближенными значениями коэффициентов $c'_1,...,c'_n$; $d'_1,...,d'_n$, но и $c'_0 \neq 0$. С уменьшением шага квантования h значения коэффициентов $c'_1,...,c'_n$; $d'_1,...,d'_n$ приближаются к точным [1].

Независимо от выбранного метода дискретизации блок 8 рассчитывает ДПФ ЦР [1]:

$$W(z)=K_0\frac{1+b_1 z^{-1}+b_2 z^{-2}+...+b_n z^{-n}}{1-k_1 z^{-1}-k_2 z^{-2}-...-k_n z^{-n}}, \tag{11}$$

где $b_1=d_1$; $b_2=d_2$; $...b_n=d_n$; $k_1=c_1K_0$; $k_2=c_2K_0$; $...k_n=c_nK_0$; $K_0=1/(c_1+c_2+...+c_n)$.

Блок 9 отображает результаты расчета $GH(z)$ и $W(z)$.

С помощью блоков 10 и 11 имеется возможность построения различных характеристик (переходных, частотных и др.) НПФ $G(s)$, ДПФ $GH(z)$ и САУ с ЦР $W(z)$.

Кроме того, с помощью блоков 12 и 13 имеется возможность в пакете расширения MATLAB Simulink провести цифровое моделирование работы НПФ $G(s)$, ДПФ $GH(z)$ и САУ с ЦР $W(z)$.

3. Проверка работы программы

В качестве примера рассмотрим работу программы для ОУ второго и третьего порядков, имеющих передаточные функции

$$\begin{gathered} G_1(s)=\frac{\alpha}{s(s+a)},\ \alpha=3684{,}1c^{-2},\ a=8{,}477c^{-1}; \\ G_2(s)=\frac{\alpha}{s(s^2+bs+a)},\ \alpha=59319{,}5c^{-3},\ a=5434c^{-2},\ b=67{,}3c^{-1}. \end{gathered} \tag{12}$$

Объекты содержат цифровые входы и выходы с периодом дискретизации $T_o=0{,}001$ с, который намного меньше постоянных времени ($T=1/a=0{,}118$ с, $T=\sqrt{1/a}=0{,}122$ с) их аналоговых моделей (12). Объект $G_1(s)$ обладает нелинейностью типа насыщение $u_н=\pm 255$ делений. Аналоговый и цифровой выходы объекта связаны коэффициентом преобразования измерителя $K_п=100$ дел/град. Частота среза объекта $\omega=11{,}1$ рад/с. Объект $G_2(s)$ обладает нелинейностью типа насыщение $u_н=\pm 5400$ делений. Аналоговый и цифровой выходы объекта связаны коэффициентом преобразования измерителя $K_п=182$ дел/град. Частота среза объекта $\omega=60{,}4$ рад/с.

Используя табл. 2 и 3, для объектов (12) аналитически определим ДПФ ОУ и передаточные функции оптимальных ЦР для САУ при ступенчатых входных воздействиях [1]:

$$\begin{gathered} HG_1(z)=\frac{0{,}1791z^{-1}+0{,}1741z^{-2}}{1-1{,}9187z^{-1}+0{,}9187z^{-2}},\ h=0{,}01\text{ с}, \\ HG_2(z)=\frac{0{,}0250z^{-1}+0{,}0740z^{-2}+0{,}0149z^{-3}}{1-1{,}6687z^{-1}+1{,}0331z^{-2}-0{,}3644z^{-3}},\ h=0{,}015\text{ с}; \end{gathered} \tag{13}$$

$$W_1(z) = 2{,}8310\frac{1-0{,}9189z^{-1}}{1+0{,}4929z^{-1}},\ h = 0{,}01\,\text{с},$$
$$W_2(z) = 8{,}7780\frac{1-0{,}6687z^{-1}+0{,}3644z^{-2}}{1+0{,}7809z^{-1}+0{,}1312z^{-2}},\ h = 0{,}015\,\text{с}. \qquad (14)$$

Для расчета параметров выражений (13) и (14) предварительно был выбран шаг квантования h. При выборе шага квантования h требуется учитывать ряд противоречивых требований и следовать рекомендациям [6]. Однако практически установлено, что эффект квантования по времени мало отражается на динамике цифровой САУ, если выбирать шаг квантования h из соотношения $T_{95}/45 < h < T_{95}/15$, где T_{95} – время достижения выходным сигналом системы уровня 95 % от установившегося значения при подаче на вход ступенчатого сигнала. На практике обычно выполняется условие $T_{95} \approx 3T$, где T – постоянная времени ОУ.

Таблица 2

ДПФ ОУ с фиксатором нулевого порядка $HG(z)$

Передаточная функция ОУ $G(s)$	ДПФ ОУ с фиксатором нулевого порядка $HG(z)$
$\frac{\alpha}{s(s+a)}$	$HG_1(z) = \frac{c_1 z^{-1} + c_2 z^{-2}}{1 + d_1 z^{-1} + d_2 z^{-2}}$, где $c_1 = \frac{\alpha}{b^2}(bh - 1 + B)$; $c_2 = \frac{\alpha}{b^2}(bh - 1 - bhB)$; $d_1 = -1 - d_2$; $d_2 = B$; $B = e^{-bh}$
$\frac{\alpha}{s(s^2+bs+a)}$ при $4a - b^2 > 0$	$HG_2(z) = \frac{c_1 z^{-1} + c_2 z^{-2} + c_3 z^{-3}}{1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}}$, где $c_1 = \frac{\alpha}{a^2}\left[ah - b + b\sqrt{B}\left(\cos kh + \frac{b^2-2a}{2b\kappa}\sin kh\right)\right]$; $c_2 = \frac{\alpha}{a^2}\left[b(1-B) - 2ah\sqrt{B}\cos kh - 2b\sqrt{B}\frac{b^2-2a}{2b\kappa}\sin kh\right]$; $c_3 = \frac{\alpha}{a^2}\left[(ah+b)B - b\sqrt{B}\left(\cos kh - \frac{b^2-2a}{2b\kappa}\sin kh\right)\right]$; $d_1 = -\left(1 + 2\sqrt{B}\cos kh\right)$; $d_2 = B + 2\sqrt{B}\cos kh$; $d_3 = -B$; $B = e^{-bh}$; $k = \sqrt{a - b^2/4}$

Таблица 3

Передаточные функции оптимальных ЦР для САУ при ступенчатых входных воздействиях

Передаточная функция ОУ $G(s)$	Передаточная функция оптимального ЦР $W(z)$
$\frac{\alpha}{s(s+a)}$	$W_1(z) = K_0\frac{1+b_1 z^{-1}}{1+a_1 z^{-1}}$, где $K_0 = \frac{a}{\alpha h(1-B)}$; $b_1 = -B$; $a_1 = \frac{1-B(1+bh)}{bh(1-B)}$; $B = e^{-bh}$
$\frac{\alpha}{s(s^2+bs+a)}$ при $4a - b^2 > 0$	$W_2(z) = K_0\frac{1+b_1 z^{-1}+b_2 z^{-2}}{1+a_1 z^{-1}+a_2 z^{-2}}$, где $K_0 = \frac{a}{\alpha h(1-2\sqrt{B}\cos\kappa h + B)}$; $b_1 = -2\sqrt{B}\cos\kappa h$; $b_2 = B$; $k = \sqrt{a - b^2/4}$; $B = e^{-bh}$; $a_1 = 1 - \left\{1 - \frac{b}{ah}\left[1 - \sqrt{B}\left(\cos\kappa h + \frac{b^2-2a}{2b\kappa}\sin\kappa h\right)\right]\right\}\frac{1}{1-2\sqrt{B}\cos\kappa h + B}$; $a_2 = \left[B + \frac{b}{ah}\sqrt{B}\left(\sqrt{B} - \cos\kappa h + \frac{b^2-2a}{2b\kappa}\sin\kappa h\right)\right]\frac{1}{1-2\sqrt{B}\cos\kappa h + B}$

Используя специально разработанную программу и различные методы дискретизации, были определены ДПФ ОУ $\left(HG_{1.i}(z),\ HG_{2.i}(z)\right)$ и ЦР $\left(W_{1.i}(z),\ W_{2.i}(z)\right)$ (табл. 4).

Таблица 4

Результаты расчетов ДПФ ОУ и ЦР

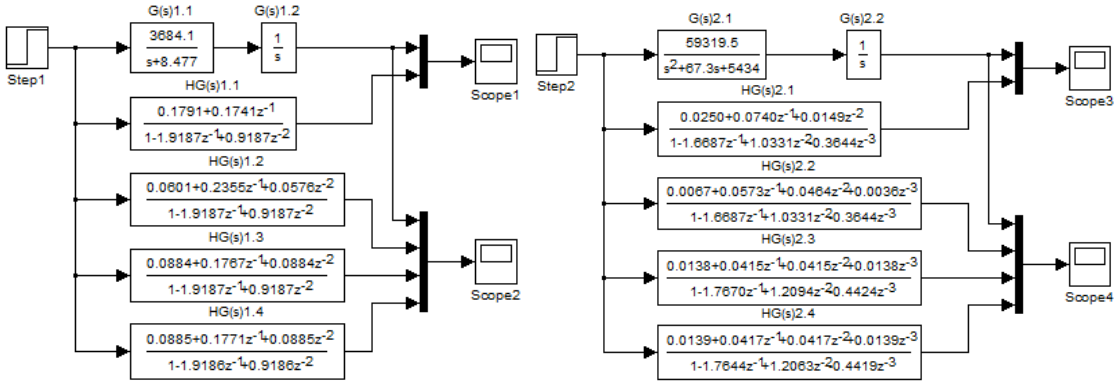
Параметры дискретизации *'method'*	ДПФ $HG_1(z)$	ДПФ $HG_2(z)$
'zoh'	$HG_{1,1}(z)=\frac{0,1791z^{-1}+0,1741z^{-2}}{1-1,9187z^{-1}+0,9187z^{-2}}$	$HG_{2,1}(z)=\frac{0,0250z^{-1}+0,0740z^{-2}+0,0149z^{-3}}{1-1,6687z^{-1}+1,0331z^{-2}-0,3644z^{-3}}$
'foh'	$HG_{1,2}(z)=\frac{0,0601+0,2355z^{-1}+0,0576z^{-2}}{1-1,9187z^{-1}+0,9187z^{-2}}$	$HG_{2,2}(z)=\frac{0,0067+0,0573z^{-1}+0,0464z^{-2}+0,0036z^{-3}}{1-1,6687z^{-1}+1,0331z^{-2}-0,3644z^{-3}}$
'tustin'	$HG_{1,3}(z)=\frac{0,0884+0,1767z^{-1}+0,0884z^{-2}}{1-1,9187z^{-1}+0,9187z^{-2}}$	$HG_{2,3}(z)=\frac{0,0138+0,0415z^{-1}+0,0415z^{-2}+0,0138z^{-3}}{1-1,7670z^{-1}+1,2094z^{-2}-0,4424z^{-3}}$
'prewarp'	$HG_{1,4}(z)=\frac{0,0885+0,1771z^{-1}+0,0885z^{-2}}{1-1,9186z^{-1}+0,9186z^{-2}}$	$HG_{2,4}(z)=\frac{0,0139+0,0417z^{-1}+0,0417z^{-2}+0,0139z^{-3}}{1-1,7644z^{-1}+1,2063z^{-2}-0,4419z^{-3}}$
Параметры дискретизации *'method'*	ДПФ $W_1(z)$	ДПФ $W_2(z)$
'zoh'	$W_{1,1}(z)=2,8313\frac{1-1,9187z^{-1}+0,9187z^{-2}}{1-0,5071z^{-1}-0,4929z^{-2}}$,	$W_{2,1}(z)=8,7796\frac{1-1,6687z^{-1}+1,0331z^{-2}-0,3644z^{-3}}{1-0,2195z^{-1}-0,6497z^{-2}-0,1308z^{-3}}$
'foh'	$W_{1,2}(z)=3,4118\frac{1-1,9187z^{-1}+0,9187z^{-2}}{1-0,8035z^{-1}-0,1965z^{-2}}$	$W_{2,2}(z)=9,3197\frac{1-1,6687z^{-1}+1,0331z^{-2}-0,3644z^{-3}}{1-0,5340z^{-1}-0,4324z^{-2}-0,0336z^{-3}}$
'tustin'	$W_{1,3}(z)=3,7722\frac{1-1,9187z^{-1}+0,9187z^{-2}}{1-0,6665z^{-1}-0,3335z^{-2}}$	$W_{2,3}(z)=10,3306\frac{1-1,7670z^{-1}+1,2094z^{-2}-0,4424z^{-3}}{1-0,4287z^{-1}-0,4207z^{-2}-0,1426z^{-3}}$
'prewarp'	$W_{1,4}(z)=3,7651\frac{1-1,9186z^{-1}+0,9186z^{-2}}{1-0,6668z^{-1}-0,3332z^{-2}}$	$W_{2,4}(z)=10,2775\frac{1-1,7644z^{-1}+1,2063z^{-2}-0,4419z^{-3}}{1-0,4286z^{-1}-0,4286z^{-2}-0,1429z^{-3}}$

Из табл. 4 следует, что ДПФ ОУ (12), полученные с помощью метода *'zoh'* и рассчитанные аналитически (13), полностью совпадают. ДПФ ОУ, полученные с помощью методов (*'foh','tustin','prewarp'*), отличаются от (13) приближенными значениями коэффициентов. Следует заметить, что передаточные функции ЦР, приведенных в табл. 4, отличаются от (14) дополнительным шагом квантования, который при малых значениях h незначительно влияет на длительность переходного процесса САУ.

Для проверки качества работы непрерывных и дискретных моделей рассматриваемых ОУ, а также САУ, содержащей ОУ из (12) и различные ЦР из (14) и табл. 4, было осуществлено математическое моделирование.

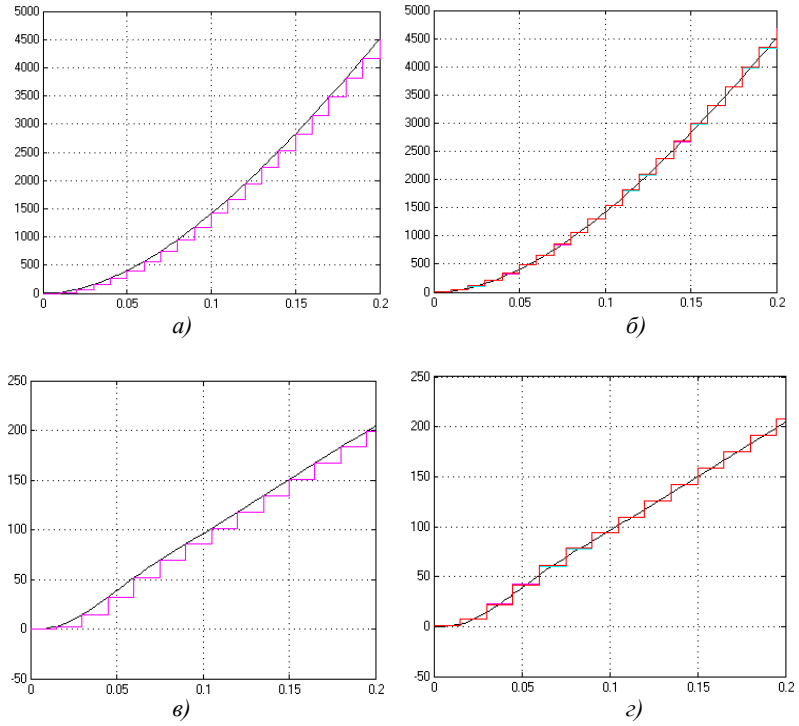
4. Математическое моделирование

С помощью блоков 12 и 13 в среде Simulink пакета прикладных программ MATLAB проведено моделирование работы непрерывных (12) и дискретных (из табл. 4) моделей рассматриваемых ОУ. Схемы моделирования показаны на рис. 2, а переходные характеристики при подаче ступенчатого входного сигнала амплитудой 100 делений – на рис. 3. Амплитуда входного сигнала выбрана исходя из обеспечения работы ОУ в зоне линейного регулирования.



а) *б)*

Рис. 2. Схемы моделирования работы непрерывных и дискретных моделей:
а) ОУ $G_1(s)$, $HG_{1,1}(z)$, $HG_{1,2}(z)$, $HG_{1,3}(z)$, $HG_{1,4}(z)$;
б) ОУ $G_2(s)$, $HG_{2,1}(z)$, $HG_{2,2}(z)$, $HG_{2,3}(z)$, $HG_{2,4}(z)$

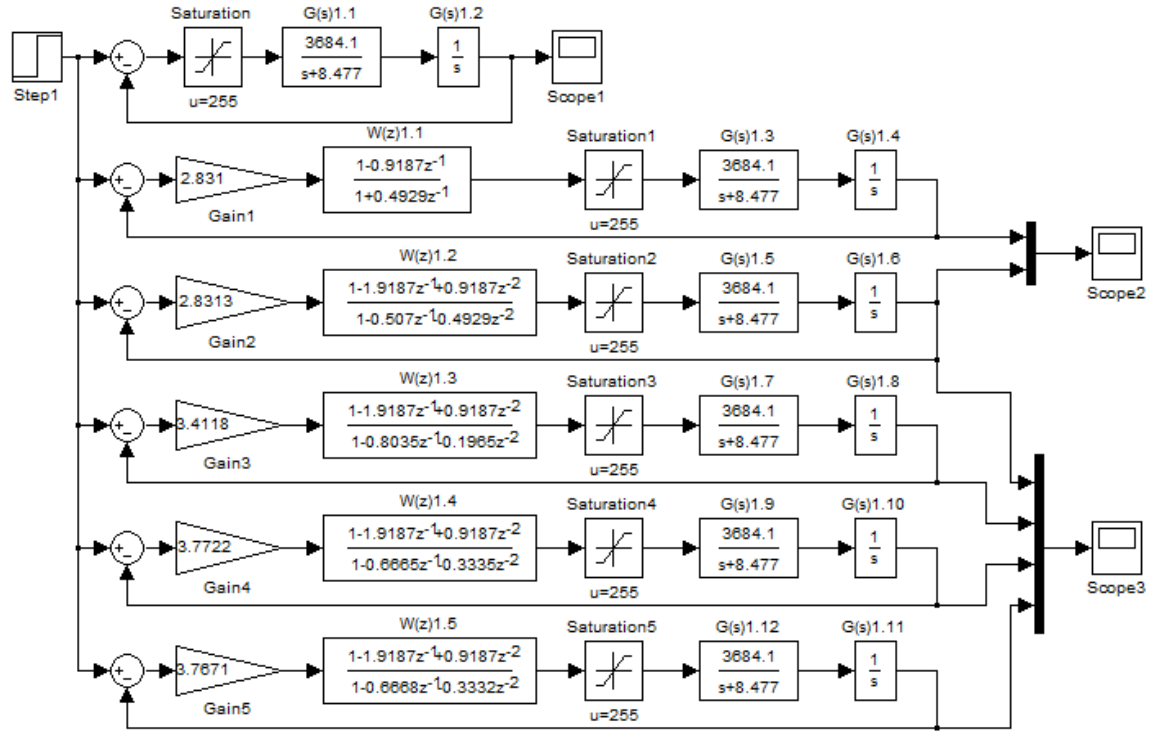


а) *б)* *в)* *г)*

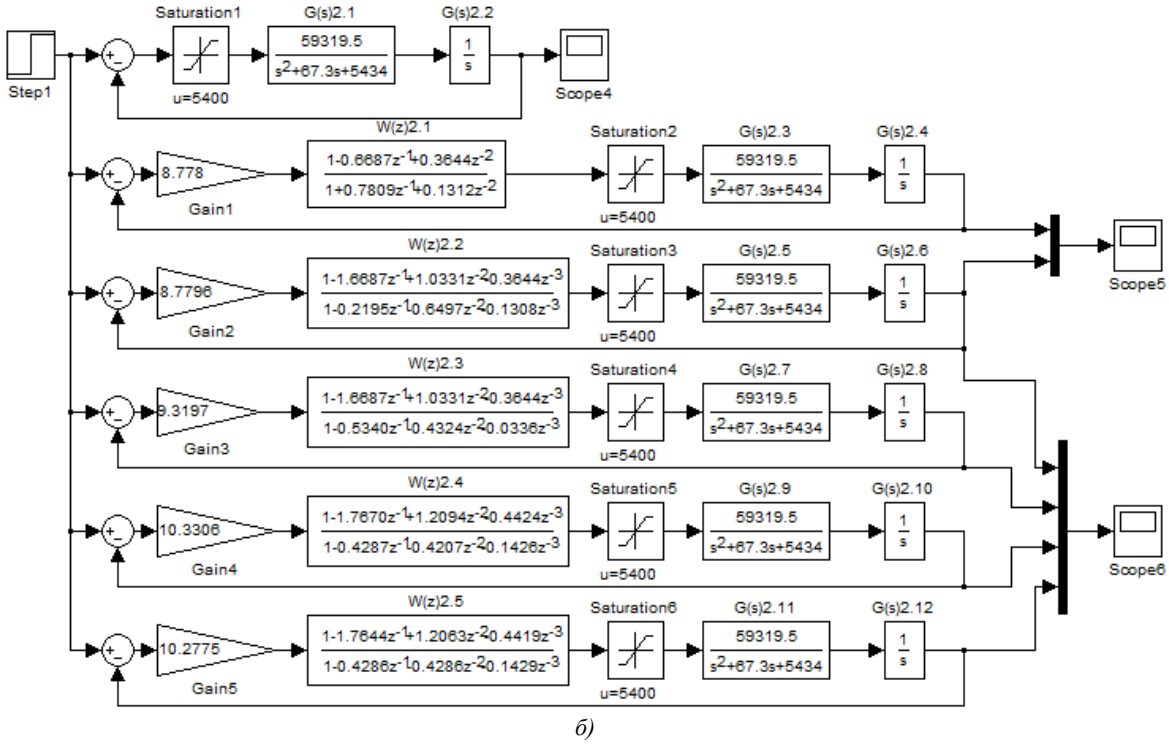
Рис. 3. Переходные характеристики непрерывных и дискретных моделей ОУ:
а) $G_1(s)$, $HG_{1,1}(z)$; *б)* $G_1(s)$, $HG_{1,2}(z)$, $HG_{1,3}(z)$, $HG_{1,4}(z)$;
в) $G_2(s)$, $HG_{2,1}(z)$; *г)* $G_2(s)$, $HG_{2,2}(z)$, $HG_{2,3}(z)$, $HG_{2,4}(z)$

На рис. 3 видно, что переходные характеристики ДПФ из табл. 4 имеют практически одинаковую степень соответствия (запаздывание не более h) с переходными характеристиками НПФ ОУ из (12).

С помощью блоков 12 и 13 в среде Simulink пакета прикладных программ MATLAB проведено математическое моделирование работы САУ, содержащей различные ОУ из (12) и ЦР из (14) и табл. 4. Схемы моделирования приведены на рис. 4, а переходные характеристики при подаче ступенчатого входного сигнала амплитудой 50 делений – на рис. 5. Амплитуда входного сигнала выбрана исходя из обеспечения работы САУ в зоне линейного регулирования.



а)



б)

Рис. 4. Схемы моделирования работы САУ:
а) с ОУ $G_1(s)$ и ЦР: $W_1(z)$, $W_{1,1}(z)$, $W_{1,2}(z)$, $W_{1,3}(z)$, $W_{1,4}(z)$;
б) ОУ $G_2(s)$ и ЦР: $W_2(z)$, $W_{2,1}(z)$, $W_{2,2}(z)$, $W_{2,3}(z)$, $W_{2,4}(z)$

Из рис. 5 и табл. 5 видно, что при работе САУ без регулятора наблюдается длительный (0,6825 с) колебательный переходной процесс. Регуляторы $W_1(z)$ и $W_{1,1}(z)$, $W_2(z)$ и $W_{2,1}(z)$ обеспечивают лучшие и одинаковые показатели качества работы САУ с минимальной длительностью переходного процесса (0,0175 с; 0,0343 с). Это указывает на то, что традиционный метод расчета ЦР по НПФ ОУ и программный метод расчета ЦР по ДПФ ОУ, определенной с помощью метода *'zoh'*, эквивалентны и могут использоваться в инженерной практике. Когда требования к быстродействию САУ и виду переходного процесса невысокие, могут использоваться регуляторы $W_{1,2}(z)$, $W_{1,3}(z)$, $W_{1,4}(z)$, $W_{2,2}(z)$, $W_{2,3}(z)$ и $W_{2,4}(z)$, рассчитанные по ДПФ ОУ, которые получены с помощью методов *'foh','tustin','prewarp'*. Данные ЦР обеспечивают длительность переходного процесса для ОУ $G_1(s)$ и $G_2(s)$ не более 0,054 и 0,102 с соответственно.

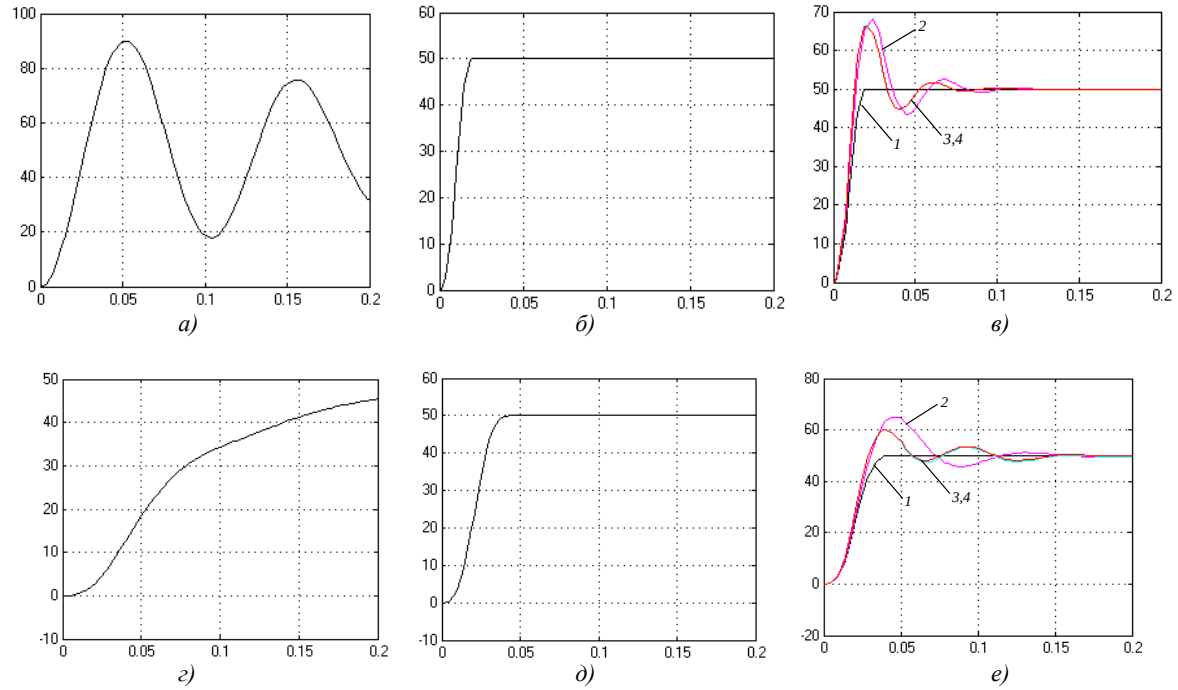


Рис. 5. Переходные характеристики САУ:
а) с ОУ $G_1(s)$ без ЦР; *б)* с ОУ $G_1(s)$ и ЦР $W_1(z)$, $W_{1,1}(z)$; *в)* с ОУ $G_1(s)$ и ЦР $W_{1,1}(z)$, $W_{1,2}(z)$, $W_{1,3}(z)$, $W_{1,4}(z)$;
г) с ОУ $G_2(s)$ без ЦР; *д)* с ОУ $G_2(s)$ и ЦР $W_2(z)$, $W_{2,1}(z)$; *е)* с ОУ $G_2(s)$ и ЦР $W_{2,1}(z)$, $W_{2,2}(z)$, $W_{2,3}(z)$, $W_{2,4}(z)$.
Нумерация на рис. 5, *в* и *е* приведена в соответствии с порядковым номером ЦР

Таблица 5

Показатели качества переходных процессов САУ

Переходной процесс	САУ	Входной сигнал, дел./град	Параметры переходного процесса	
			вид	длительность 95 %, с
Рис. 5, *а*	с ОУ $G_1(s)$ без ЦР;	50 / 0,5	колебательный	0,6825
Рис. 5, *б*	с ОУ $G_1(s)$ и ЦР $W_1(z)$, $W_{1,1}(z)$;	50 / 0,5	апериодический	0,0175
Рис. 5, *в* (2)	с ОУ $G_1(s)$ и ЦР $W_{1,2}(z)$;	50 / 0,5	колебательный	0,0485
Рис. 5, *в* (3,4)	с ОУ $G_1(s)$ и ЦР $W_{1,3}(z)$, $W_{1,4}(z)$;	50 / 0,5	колебательный	0,0540
Рис. 5, *г*	с ОУ $G_2(s)$, без ЦР;	50 / 0,28	апериодический	0,2495
Рис. 5, *д*	с ОУ $G_2(s)$ и ЦР $W_2(z)$, $W_{2,1}(z)$;	50 / 0,28	апериодический	0,0343
Рис. 5, *е* (2)	с ОУ $G_2(s)$ и ЦР $W_{2,2}(z)$;	50 / 0,28	колебательный	0,1020
Рис. 5, *е* (3,4)	с ОУ $G_2(s)$ и ЦР $W_{2,3}(z)$, $W_{2,4}(z)$	50 / 0,28	колебательный	0,1010

Заключение

Проведенные исследования показывают, что с помощью пакета MATLAB и специально разработанного алгоритма могут быть получены ДПФ ОУ различных порядков и на их основе синтезированы ЦР. Показатели качества работы САУ с ЦР, рассчитанными с помощью метода *'zoh'* , не уступают показателям качества САУ с ЦР, рассчитанными традиционным аналитическим методом (по непрерывным моделям ОУ). Вместе с тем разработанный алгоритм является универсальным и позволяет достаточно просто осуществить синтез ЦР для систем с ОУ любого (выше третьего) порядка. Использование разработанного программного метода значительно упрощает процесс расчета ЦР, уменьшает трудоемкость и сокращает время синтеза регуляторов при проектировании цифровых САУ.

Список литературы

1. Гостев, В.И. Системы автоматического управления с цифровыми регуляторами : справочник / В.И. Гостев, В.К. Стеклов. – Київ : Радіоаматор, 1998. – 704 с.

2. Franklin, G.F. Digital Control of Dynamic Systems (3rd Edition) / G.F. Franklin, J.D. Powell, M.L. Workman. – Addison – Wesley, 1997. – 850 p.

3. Бермант, А.Ф. Курс математического анализа. Часть 1 / А.Ф. Бермант – М. : Гос. изд-во физ.-мат. лит., 1959. – 467 с.

4. Методы классической и современной теории автоматического управления : учебник. В 5 т. / Н.Д. Егупов [и др.]; под общ. ред. Н.Д. Егупова. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2000. – Т.1 : Математические модели, динамические характеристики и анализ систем автоматического управления. – 656 с.

5. Tewari, A. Modern control design with MATLAB and Simulink / A. Tewari. – Wiley, 2002. – 503 p.

6. Гостев, В.И. Синтез цифровых регуляторов систем автоматического управления / В.И. Гостев, Д.А. Худолий, А.А. Баранов. – Киев : Техника, 2000. – 575 с.

Поступила 10.01.2013

НПООО «ОКБ Техносоюзпроект»,
Минск, пр. Независимости, 115
e-mail: aliaksei.rusakovich@gmail.com.

A.G. Stryzhniou, A.N. Rusakovich

COMPUTER-AIDED SYNTESIS OF DIGITAL CONTROLLERS BASED ON THE DISCRETE TRANSFER FUNCTION OF THE CONTROL OBJECTS

The paper presents discretization methods of control objects transfer functions, which are used in MATLAB, including zero- and first-order extrapolators, bilinear Tustin approximation and Tustin approximation with frequency prewarping. The MATLAB program which automates the process of determining discrete transfer functions of various control objects from their continuous models and calculates the digital controllers is developed. Discrete transfer functions and digital controllers for control objects of the second and third order are obtained programmatically. The digital modeling is applied to verify the operability of the control objects and the automatic control systems with different digital controllers.

УДК 629.33.021:004.94

В.С. Кончак[2], А.А. Назаренко[1], С.В. Хитриков[2], Д.А. Бузановский[1], С.П. Лазакович[2], Ю.И. Николаев[3]

ВЕРИФИКАЦИЯ КОМПЬЮТЕРНЫХ МОДЕЛЕЙ ЭЛЕМЕНТОВ РЫЧАЖНОЙ ДЛИННОХОДОВОЙ ПОДВЕСКИ ПО РЕЗУЛЬТАТАМ СТЕНДОВЫХ ИСПЫТАНИЙ

Рассматриваются методы, позволяющие построить компьютерную модель длинноходовой подвески, соответствующую реальной. Показывается, что основным источником информации для получения параметров аналитической модели являются стендовые испытания. Проводится сравнительный анализ результатов стендовых и виртуальных испытаний, выполненных с использованием верифицированной модели.

Введение

Необходимость снижения стоимости и сроков разработки объектов новой техники потребовала широкого использования компьютерного моделирования узлов и систем машиностроительных конструкций как средства оценки их динамических характеристик. При этом в качестве модели чаще всего используется математическое описание кинематики и динамики исследуемых конструкций. Максимальный эффект от применения моделирования может быть достигнут, если проектирование ведется с использованием гаммы типовых унифицированных сборочных единиц, применение которых позволяет создать конструкцию, обладающую необходимыми потребительскими качествами. Наличие отработанных узлов и агрегатов обеспечивает возможность на их основе создать библиотеку моделей типовых элементов, используя которые, уже на начальном этапе проведения конструкторских работ можно разрабатывать модели для проверки функционирования различных систем проектируемого изделия.

При разработке математической модели применяют аналитические либо вероятностные подходы. Для механических конструкций, работающих под воздействием переменных нагрузок, динамика колебаний отображает зависимость перемещения их элементов от приложенной силы и описывается уравнениями Лагранжа второго рода совместно с алгебраическими уравнениями связи, а в простейших случаях – вторым законом Ньютона, задающим равновесие приложенных сил.

Всякая модель механической конструкции содержит ряд параметров, которые отражают ее физические и механические свойства и могут быть получены расчетным путем либо экспериментально. Вычисленные таким образом параметры импортируются в модель, после чего проводятся сравнительные испытания, на основании которых выполняется ее верификация. В соответствии с определением, данным в [1], верификация – это «разновидность анализа, имеющая целью установление соответствия двух описаний одного и того же объекта. Различают верификацию структурную и функциональную. При структурной верификации устанавливается соответствие структур, отображаемых двумя описаниями. При функциональной (параметрической) верификации проверяется соответствие процессов функционирования и, в частности, выходных параметров, отображаемых сравниваемыми описаниями. Функциональная верификация выполняется путем анализа переходных процессов с учетом перекрестных помех и задержек сигнала». Следовательно, верификация – это процедура получения оператора исследуемой системы, который преобразует входные сигналы в выходную реакцию.

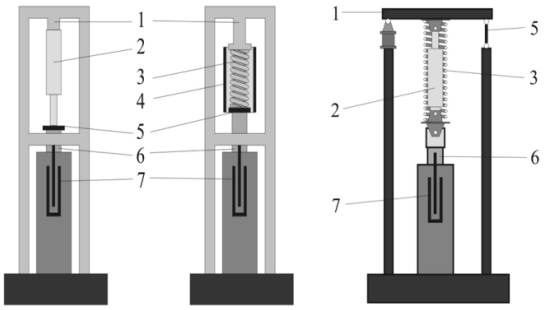
Основным источником информации для получения таких характеристик динамических систем, как собственная частота колебаний, коэффициент затухания, амплитудные спектры входа, выхода и передаточной функции, фазовый спектр, коэффициенты жесткости и сопротивления перемещению, являются стендовые либо натурные испытания реальной конструкции.

1. Методы организации стендовых и виртуальных испытаний

Постановка эксперимента на стенде осуществляется в соответствии со схемой испытаний, которая разрабатывается в строгом соответствии с выбранной математической моделью и должна содержать способ крепления изделия на стенде, схему установки датчиков, координаты и направление приложенных сил и перемещений элементов конструкции. Исследования должны проводиться в соответствии с действующими стандартами, а также программой и методикой испытаний.

Рассмотрим основные принципы получения экспериментальной информации при решении задачи верификации на примере моделирования колебаний длинноходовой рычажной подвески, у которой в качестве упругого элемента используются пружина, обеспечивающая возможность перемещения ступицы колеса в пределах ±200 мм; амортизатор, допускающий те же перемещения, и стойка подвески в сборе.

Техническое оснащение стендовых испытаний выполнено на основании схемы испытаний (рис. 1). В качестве нагружающего устройства использован испытательный комплекс фирмы SHENK, состоящий из гидравлической станции, гидроцилиндров различной мощности, крепежной плиты, системы гидрорегуляторов для управления потоком жидкости, аналоговой системы контроля за ходом эксперимента и видеонаблюдения за ним.



а) *б)* *в)*

Рис. 1. Схемы испытательных стендов: *а)* амортизатора; *б)* пружины; *в)* стойки подвески в сборе (*1* – неподвижная балочка; *2* – амортизатор; *3* – пружина; *4* – защитный кожух; *5* – датчик силы; *6* – гидроцилиндр; *7* – датчик перемещения)

Процесс моделирования сигналов управления экспериментом и измерения экспериментальной информации выполнен цифровым многоканальным управляющим комплексом, разработанным и изготовленным в Республиканском компьютерном центре машиностроительного профиля Объединенного института машиностроения НАН Беларуси. Измерение входных и выходных функций в процессе эксперимента осуществлялось с помощью датчиков силы 5 и перемещений 7 (рис. 1). Оцифровка информации с датчиков и передача данных в вычислительный комплекс выполнены с помощью многоканальных синхронных аналого-цифровых преобразователей.

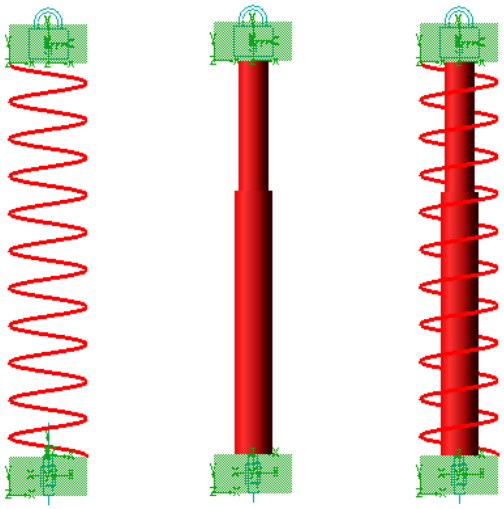
Фотографии стендов, использованных для проведения экспериментов, показаны на рис. 2.

а) *б)* *в)*

Рис. 2. Испытательные стенды: *а)* амортизатора; *б)* пружины; *в)* стойки подвески в сборе

Испытания проводились в режиме вынужденных колебаний. Поэтому исследуемый объект (амортизатор, пружина, стойка подвески) устанавливался в замкнутый контур оснастки таким образом, чтобы его верхняя опора упиралась в станину стенда, а нижняя была прикреплена к штоку гидроцилиндра 6, который в процессе испытаний выполнял гармонические колебания с заданной частотой и амплитудой. Процессы управления цилиндром и измерения экспериментальной информации синхронизированы, что позволяет с высокой точностью определить фазовые характеристики.

При проведении виртуальных испытаний динамических моделей соблюдаются все условия их организации, принятые для реальных объектов. Для их организации в пакете ADAMS была разработана динамическая модель стенда. На рис. 3 изображены модели стендов, использованных при испытании моделей пружины, амортизатора и стойки подвески в сборе. С учетом принципа максимального правдоподобия виртуальные стенды построены в соответствии со схемами эксперимента (см. рис. 1). Верхний элемент исследуемого объекта закреплен неподвижно, а для нижнего задается закон движения, имитирующий работу гидроцилиндра. В ходе испытания ведется измерение деформации, полученной под воздействием приложенной вынуждающей силы, скорости перемещения и силы сопротивления движению исследуемого объекта.



а) *б)* *в)*

Рис. 3. Модели виртуальных стендов для испытания: *а)* пружины; *б)* амортизатора; *в)* стойки подвески в сборе

При проведении виртуальных испытаний закон движения формировался сплайнами [2] с использованием импортированных массивов процессов нагружения, полученных по результатам стендовых испытаний.

2. Методы обработки информации

С учетом принятой схемы эксперимента, когда у исследуемого объекта отсутствует подрессоренная масса, условия динамического равновесия преобразуются в уравнение

$$r\dot{y} + cy = f \sin \omega t, \tag{1}$$

решением которого будет функция перемещения

$$y = A\sin(\omega t + \alpha), \tag{2}$$

где r – коэффициент сопротивления перемещению; c – жесткость механической конструкции; f – амплитуда вынуждающей силы; A – амплитуда вынужденных колебаний. Как было показано в работе [3], параметры c и r по результатам обработки полученных в процессе эксперимента данных вычисляются как

$$c = \frac{f\cos\alpha}{A},\ r = \frac{f\sin\alpha}{A\omega}. \qquad (3)$$

Из соотношений (3) видно, что для вычисления параметров уравнения (1) в процессе эксперимента должны быть получены:

– амплитуда $f(\omega_k)$ входного силового воздействия;

– амплитуда $A(\omega_k)$ выходной реакции (перемещения);

– фазовое смещение α_k, характеризующее время реакции исследуемой системы на приложенное воздействие.

Таким образом, в процессе эксперимента достаточно измерить входную $x(i\Delta t)$ и выходную $y(i\Delta t)$ последовательности. Использование при проведении эксперимента гармонической функции в качестве вынуждающего силового воздействия продиктовано условием, что решением уравнения (1) является функция (2), а так как при вынужденных колебаниях на установившемся режиме функция перемещения (2) практически повторяет форму колебаний силового воздействия, то оно должно быть гармонической функцией. Кроме того, наличие у объекта моделирования нелинейных свойств приводит к нарушению принципа суперпозиции. Использование в таком случае в качестве входа негармонических функций вызывает в спектре выхода неконтролируемые погрешности.

Следует все же заметить, что уравнение (1) пригодно для описания только таких объектов, у которых колебания, вызванные гармонической вынуждающей силой, не порождают в реакции системы других гармонических колебаний. Чтобы убедиться в том, что система линейна, достаточно вычислить спектр выходного сигнала. Если в спектре присутствует только частота вынужденных колебаний, то объект линеен. В противном случае объект вызывает искажения входного воздействия и требует построения другой модели.

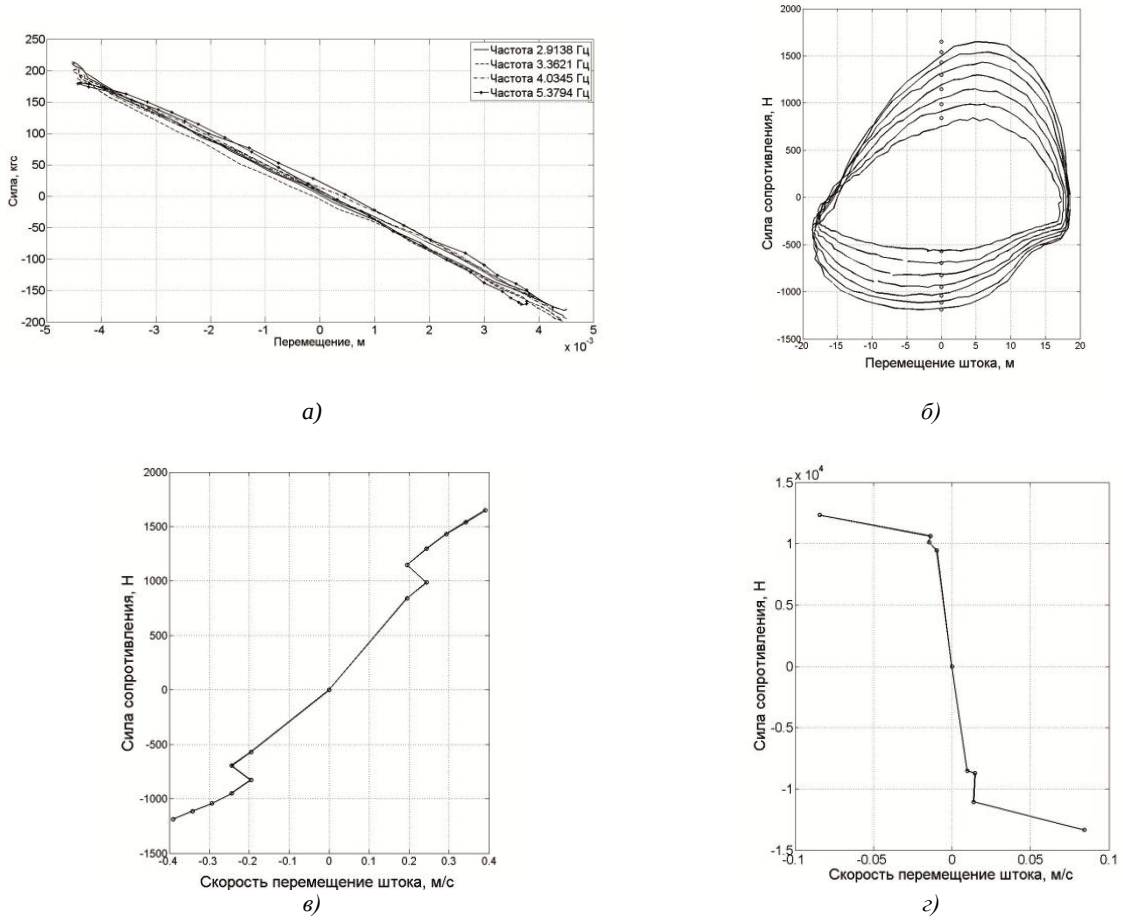
Для определения характера взаимной зависимости реакции исследуемого объекта на входное воздействие в системах подрессоривания автомобильной техники используют рабочие диаграммы, отображающие функциональную связь силы сопротивления движению от величины перемещения их элементов. На рис. 4, *а* показана рабочая диаграмма пружины для перемещений поршня стенда в пределах ±0,045 м на частотах колебаний 2,91; 3,36; 4,0; 5,38 Гц. Видно, что упругая сила пружины линейно зависит от перемещений и практически не зависит от частоты колебаний. Тангенс угла наклона прямой, отображающей функциональную зависимость выхода от входа, характеризует величину жесткости пружины, которая может быть вычислена как $c = \mathrm{tg}\,\gamma$. На рис. 4, *б* изображена рабочая диаграмма амортизатора для перемещения поршня в пределах ±0,02 м на частотах от 0,24 до 1,7 Гц. Из приведенной диаграммы можно сделать вывод, что амортизатор в рассмотренной полосе частот обладает некоторой нелинейностью. Для исследования характера нелинейности [4] была построена характеристика, график которой изображен на рис. 4, *в*. График зависимости силы сопротивления перемещению от скорости колебаний штока амортизатора был построен на максимальных скоростях перемещений. На рис. 4, *г* приведена характеристика стойки подвески в сборе. Как и следовало ожидать, эта характеристика нелинейная и из-за высокой жесткости пружины повернута по отношению к характеристике амортизатора более чем на 45 °.

Следует заметить, что характеристику амортизатора можно построить и для каждой частоты, используя весь диапазон изменения скорости перемещения штока амортизатора. Так как функция перемещения задана уравнением

$$y(i\Delta t) = A\sin 2\pi k\Delta f i\Delta t,$$

ее производная (функция скорости) задается уравнением

$$\dot{y}(i\Delta t) = A2\pi k\Delta f\cos 2\pi k\Delta f i\Delta t.$$



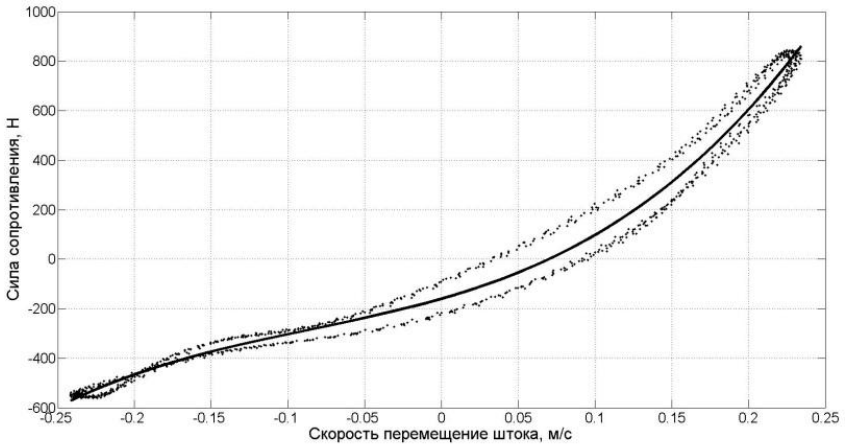
а) *б)* *в)* *г)*

Рис. 4. Функциональные зависимости: *а)* зависимость упругой силы пружины от величины перемещения; *б)* рабочая диаграмма амортизатора; *в)* характеристика амортизатора; *г)* характеристика стойки подвески в сборе

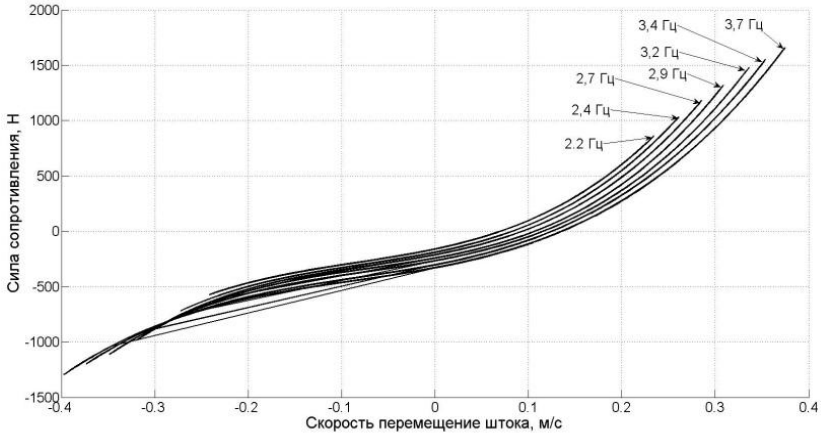
На рис. 5, *а* сплошной линией изображена характеристика амортизатора, построенная с помощью полиномиальной аппроксимации по методу наименьших квадратов с использованием точек рассеяния, отображающих функциональную зависимость силы сопротивления движению от скорости перемещения штока амортизатора. На рис. 5, *б* приведены характеристики амортизатора, построенные указанным выше способом, для частот 2,2–3,7 Гц, а на рис. 5, *в* – для стойки подвески в сборе.

Из соотношения (3) видно, что параметры, полученные в результате подстановки решения (2) в уравнение (1), являются функциями частоты. Эти параметры присутствуют и в импульсной переходной характеристике, которая является интегральным оператором для линейных систем с памятью. В отличие от соотношений (3) в данном операторе они присутствуют как константы. Так как исследуемая система обладает незначительной нелинейностью, которая возникла из-за нелинейного поведения амортизатора, то далее ансамбли реализаций входных и выходных последовательностей перед обработкой подверглись фильтрации, в результате чего объект и его модель были линеаризованы. Учитывая, что импульсная переходная характеристика исследуемого объекта является источником информации для получения таких параметров, как собственная частота и коэффициент затухания колебаний, которые, в свою очередь, отображают жесткость и коэффициент сопротивления колебаниям, по коэффициентам преобразования Фурье $S_x(j\omega_k)$ и $S_y(j\omega_k)$ входной и выходной последовательностей были вычислены коэффициенты импульсной переходной характеристики $S_h(j\omega_k)$ в соответствии с формулой

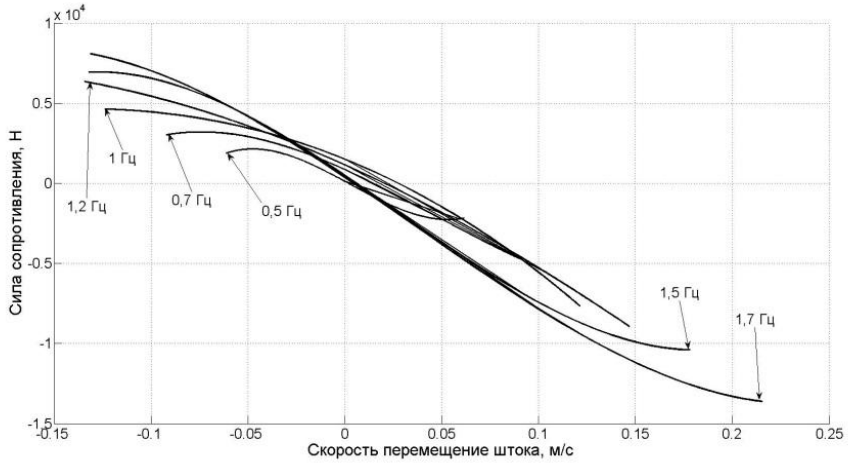
$$S_h(j\omega_k) = \frac{S_y(j\omega_k)S^*(j\omega_k)}{|S_x(j\omega_k)|}.$$



а)



б)



в)

Рис. 5. Характеристики нелинейных объектов, построенные полиномом третьей степени: *а*) амортизатора для частоты 0,97656 Гц; *б)* амортизатора для частот 2,2–3,7 Гц; *в)* стойки подвески для частот 0,5–1,7 Гц

Импульсная переходная характеристика была получена в результате обратного преобразования Фурье от коэффициентов $S_h(j\omega_k)$:

$$\overrightarrow{h(i\Delta t)} = F_{ik}^{-1}\overrightarrow{S_h(j\omega_k)} = A_0 e^{-\beta i\Delta t}\sin(\omega_0 i\Delta t + \alpha),$$

где $\beta = r/2m$ – коэффициент демпфирования; $\omega_0 = \sqrt{c/m}$ – частота собственных колебаний.

Используя алгоритм, представленный в работе [5], параметры исследуемого объекта вычисляются как собственная частота:

$$\omega_0 = \frac{2\pi}{T_0} \text{ рад/с},$$

где T_0 – период собственных колебаний, вычисляемый как среднее значение от временных интервалов T_i, измеренных между соседними локальными максимумами импульсной переходной характеристики:

$$T_0 = \frac{\sum_{i=1}^{n} T_i}{n}. \quad (4)$$

Скорость затухания колебаний характеризует декремент затухания, который определяется как среднее значение натурального логарифма от отношения величин двух соседних максимумов:

$$\lambda = \frac{\sum_{i=1}^{n} \ln\left(\frac{A_i}{A_{i+1}}\right)}{n}. \quad (5)$$

Коэффициент демпфирования вычисляется в соответствии с выражением

$$\beta = \frac{\lambda}{T_0}. \quad (6)$$

3. Сравнение результатов обработки экспериментальной информации, полученной по итогам стендовых и виртуальных испытаний

В процессе проведения сравнительных испытаний были вычислены:

– импульсная переходная характеристика пружины;

– импульсная переходная характеристика амортизатора;

– импульсная переходная характеристика стойки подвески в сборе при статическом нагружении 2,15 т;

– импульсная переходная характеристика стойки подвески в сборе при статическом нагружении 4,5 т;

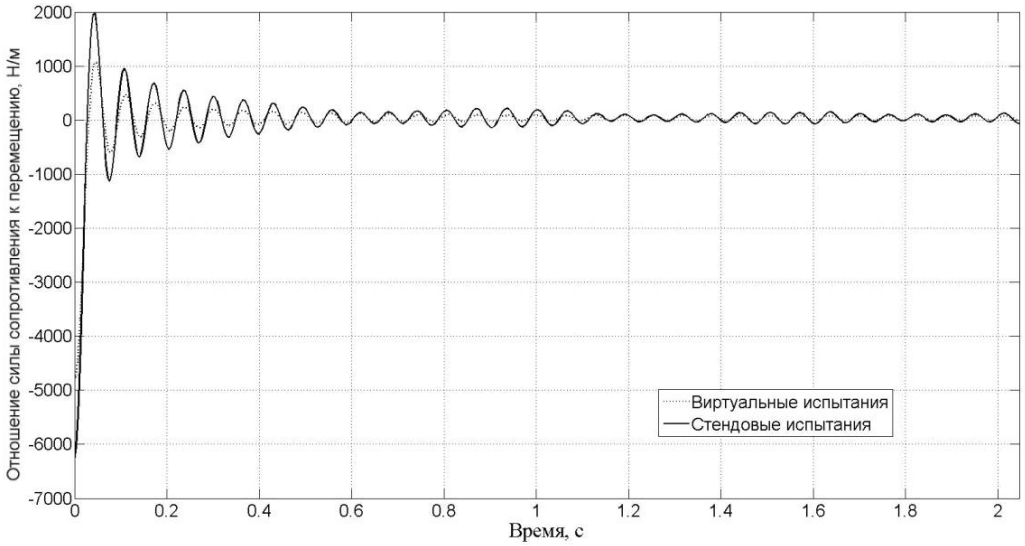
– аппроксимированные полиномами четвертой степени характеристики реального амортизатора и его виртуальной модели.

Результаты выполненных расчетов оператора исследуемых объектов приведены на графиках:

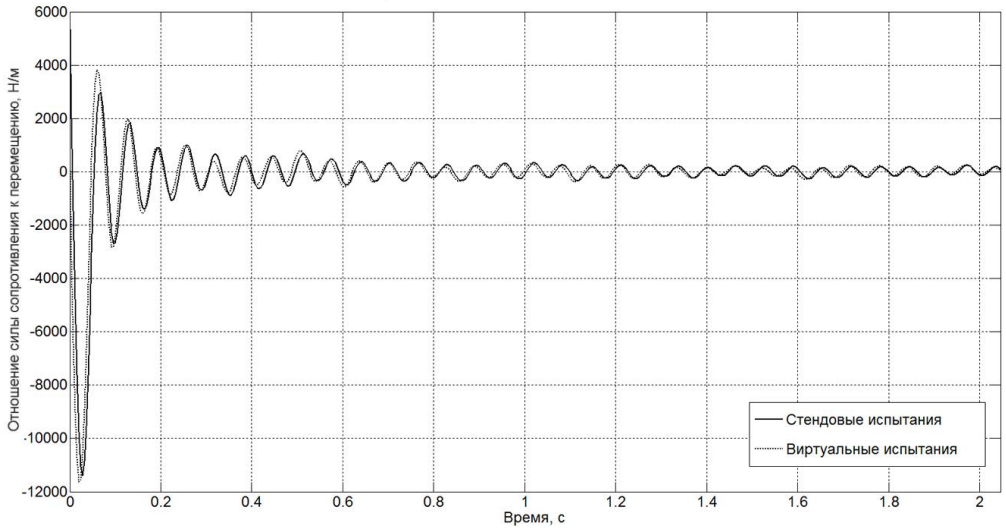
– импульсной переходной характеристики пружины (рис. 6, *а*);

– импульсной переходной характеристики амортизатора (рис. 6, *б*);

– импульсной переходной характеристики стойки подвески для статики 4,5 т (рис. 7, *а*) и для статики 2,15 т (рис. 7, *б*).



а)



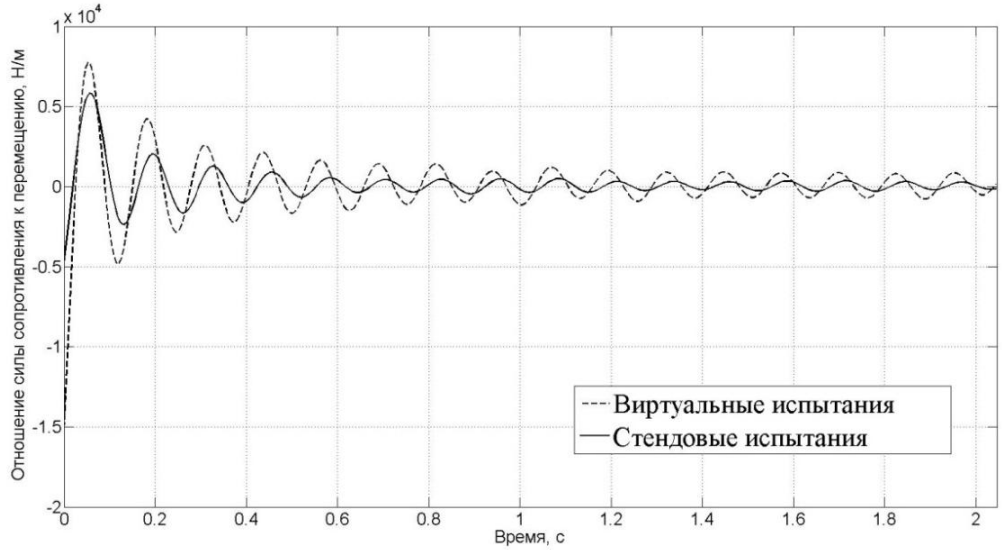
б)

Рис. 6. Импульсные переходные характеристики элементов подвески: *а)* пружины; *б)* амортизатора

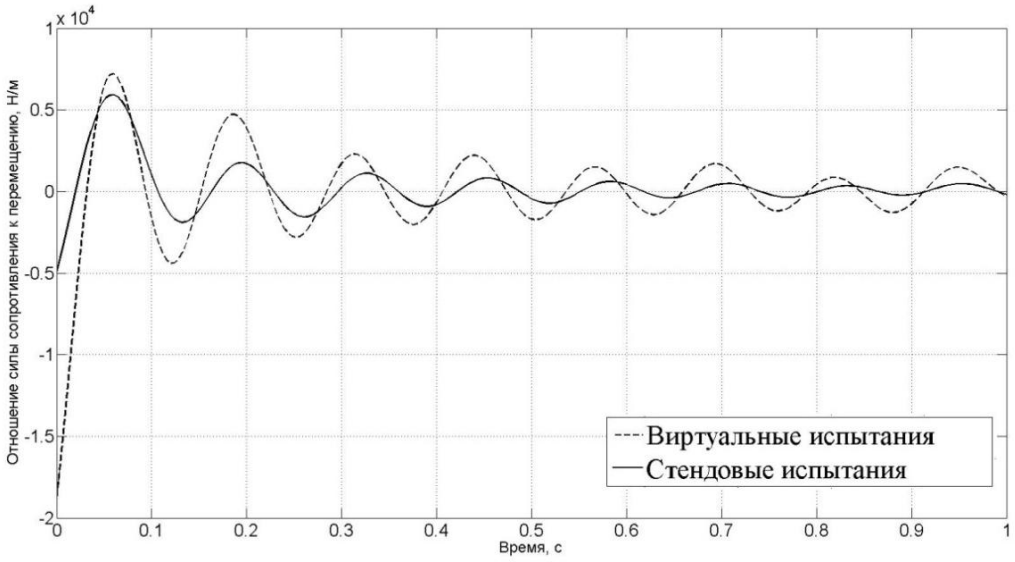
Из рис. 6 и 7 видно, что импульсные переходные характеристики исследуемых элементов подвески, полученные по результатам стендовых и виртуальных испытаний, хорошо совпадают как по частотам, так и по характеру затухания колебаний.

Сравнительный анализ результатов стендовых и виртуальных испытаний

Предмет исследования	T_0, с		Погрешность, %	ω_0, рад/с		Погрешность, %	λ		Погрешность, %	β, $с^{-1}$		Погрешность, %
	стенд.	вирт.		стенд.	вирт.		стенд.	вирт.		стенд.	вирт.	
Пружина	0,058	0,0587	1,192	108,33	107,1	1,135	0,0976	0,0913	6,455	1,683	1,555	7,605
Амортизатор	0,0643	0,0587	8,709	97,666	107,1	8,808	0,0949	0,0917	3,372	1,476	1,563	5,566
Подвеска в сборе (статика 4,5 т)	0,128	0,127	0,781	49,1	49,5	0,808	0,45	0,42	6,667	3,516	3,307	5,944
Подвеска в сборе (статика 2,15 т)	0,129	0,127	1,551	48,8	49,5	1,414	0,38	0,3	7,895	2,946	2,756	6,449



а)



б)

Рис. 7. Импульсная переходная характеристика стойки подвески при статическом нагружении: *а)* 4,5 т; *б)* 2,15 т

Результаты сравнительного анализа сведены в таблицу, где приведены как сами значения параметров, так и расчет погрешностей их вычисления по результатам виртуальных испытаний. Максимальная погрешность оценки не превосходит 9 %. Исследуемая модель объекта нелинейная. Ее характеристика, представленная на рис. 8 сплошной линией, была вычислена по результатам стендовых испытаний и импортирована в ADAMS для создания модели.

Полученная по результатам виртуальных испытаний характеристика модели показана на рис. 8 пунктирной линией. Несовпадение этих характеристик связано с погрешностями их построения. Оценка указанных характеристик была вычислена в виде полинома четвертой степени по методу наименьших квадратов, а полученная разность в их представлении не превосходит среднеквадратического отклонения поля рассеяния точек, по которым строится полином.

Кроме того, в соответствии с технологией изготовления амортизаторов точность усилия, при котором открываются перепускные дроссели на скорости перемещения 0,52 м/с, гарантируется в пределах ±15 % от расчетной величины. Поэтому точность расчета характеристики, не превышающую 9 %, можно считать удовлетворительной.

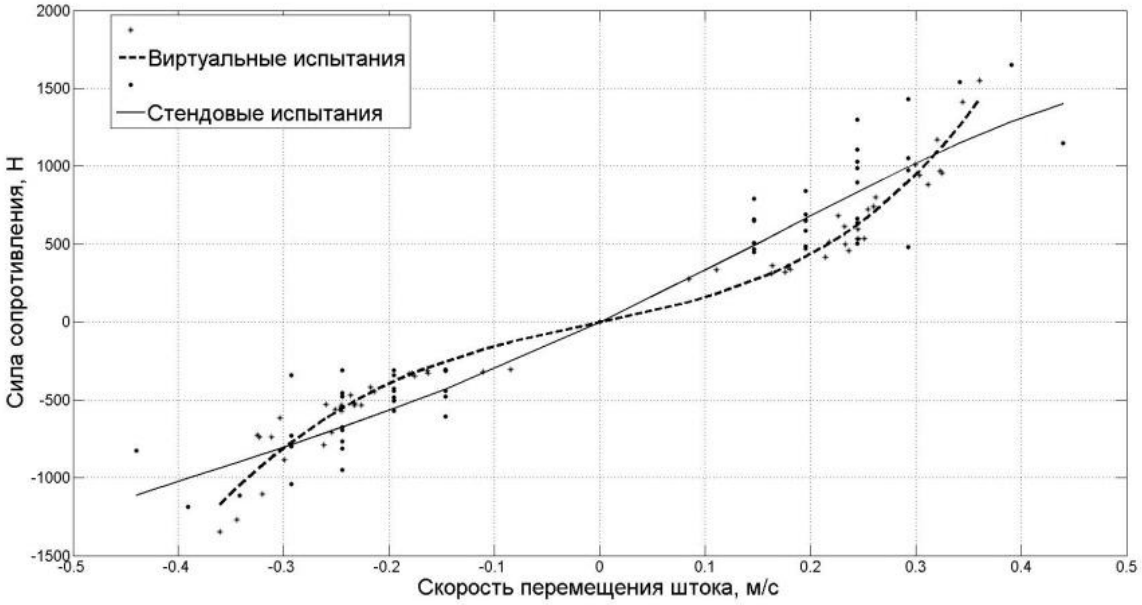


Рис. 8. Сравнение характеристик амортизатора, полученных по результатам стендовых и виртуальных испытаний

Заключение

В работе приведены результаты обработки экспериментальной информации, полученной при выполнении стендовых испытаний элементов длинноходовой подвески автомобиля МЗКТ 6001.

В результате обработки получены параметры пружины, нелинейная характеристика амортизатора и стойки подвески, которые были переданы в пакет ADAMS для создания компьютерных моделей перечисленных элементов системы подрессоривания шасси указанного автомобиля. Рассмотрены схемы испытаний, по которым в пакете ADAMS были разработаны модели виртуальных стендов. Проведены виртуальные испытания разработанных моделей элементов подвески и дан сравнительный анализ из результатов. Показано, что величины погрешности не превосходят допусков на регулировку гидросистемы амортизатора, определяемых стандартами.

Разработанная и верифицированная модель стойки подвески может быть использована при построении модели подрессоривания шасси автомобиля для прогнозирования плавности его хода.

Список литературы

1. Системы автоматизированного проектирования. Кн. 9 / Д.М. Жук [и др.] – М. : Высшая школа, 1986. – 160 с.

2. Поляков, К.А. Создание виртуальных моделей в пакете прикладных программ ADAMS / К.А. Поляков. – Самара : Самарский университет, 2003. – 88 с.

3. Методы определения динамических характеристик упругих элементов подвески по экспериментальным данным / В.С. Кончак [и др.] // Весці НАН Беларусі. Сер. фіз.-тэхн. навук. – 2008. – № 2. – С. 20–25.

4. Дербаремдикер, А.Д. Амортизаторы транспортных машин / А.Д. Дербаремдикер. – М. : Машиностроение, 1985. – 197 с.

5. Петько, В.И. Экспериментальные исследования динамических свойств элементов машиностроительных конструкций / В.И. Петько, В.С. Кончак, А.Н. Колесникович // Материалы Междунар. науч. конф. «Механика машин на пороге III тысячелетия», 23, 24 ноября 2000 г. – Минск, 2000. – С. 397–404.

Поступила 11.12.12

[1]*Объединенный институт проблем информатики НАН Беларуси,*
Минск, ул. Сурганова, 6
e-mail: lipn@newman.bas-net.by

[2]*Объединенный институт машиностроения НАН Беларуси,*
Минск, ул. Академическая, 12
e-mail: bats@ncpmm.bas-net.by

[3]*ОАО «Минский завод колесных тягачей»,*
Минск, пр. Партизанский, 150
e-mail: ugk@mzkt.by

V.S. Konchak, A.A. Nazarenko, S.V. Hitrikov, D.A. Buzanovsky, S.P. Lazakovich, J.I. Nikolaev

COMPUTER MODELS VERIFICATION OF LEVER LONG-RUNNING SUSPENSION ELEMENTS BASED ON THE BENCH TESTS RESULTS

The computer modeling methods of lever long-running suspension are considered. It is shown that the main source of information for the analytical model is bench tests. The results of bench tests and virtual tests performed by the verified model are given. A comparative analysis of the tests results is performed.

ЗАЩИТА ИНФОРМАЦИИ

УДК 004.421.6: 519.237

Ю.С. Харин, В.Ю. Палуха

ИНФОРМАТИВНЫЕ ПРИЗНАКИ ДЛЯ СТАТИСТИЧЕСКОГО РАСПОЗНАВАНИЯ КРИПТОГРАФИЧЕСКИХ ГЕНЕРАТОРОВ

Разрабатывается общий подход к построению информативных признаков для решения задачи статистического распознавания криптографических генераторов. Описываются признаки, построенные в соответствии с этим подходом. Приводятся результаты компьютерных экспериментов применения построенных информативных признаков для статистического распознавания генераторов.

Введение

Современная криптография невозможна без случайных и псевдослучайных последовательностей $x_1, x_2, ... \in V = \{0,1\}$ [1]. Случайные последовательности генерируются при помощи физических генераторов, а псевдослучайные – при помощи программных генераторов. Практическую значимость имеют генераторы последовательностей, близких по своим свойствам к равномерно распределенной случайной последовательности (РРСП). РРСП – это случайная последовательность x_1, x_2, …, x_t, x_{t+1}, …, определенная на вероятностном пространстве $(\Omega, \mathcal{F}, P)$ и удовлетворяющая двум требованиям [1]:

C_1: Для любого $n \in \mathbb{N}$ и произвольных значений индексов $1 \le t_1 < ... < t_n$ случайные величины $x_{t_1}, ..., x_{t_n}$ независимы в совокупности.

C_2: Для любого номера $t \in \mathbb{N}$ случайная величина x_t является бернуллиевой и имеет равномерное распределение вероятностей $P\{x_t = i\} = \frac{1}{2}, \quad i \in V = \{0,1\}$.

Гипотезу о том, что выходная последовательность генератора $\{x_t\}$ является равномерно распределенной, будем обозначать H_* = {{x_t} есть РРСП}, а альтернативу – $H = \overline{H_*}$.

При проведении испытаний средств криптографической защиты информации [1, 2] с целью оценки их надежности возникают следующие задачи: определить, какой тип генератора случайной или псевдослучайной последовательности использовался; обнаружить, не ухудшились ли криптографические свойства генератора с течением времени. Математической сущностью этих практических задач является задача статистического распознавания генераторов случайных и псевдослучайных последовательностей, т. е. задача отнесения (классификации) наблюдаемой выходной последовательности генератора $x_1, x_2, ..., x_T \in V = \{0,1\}$ некоторой конечной длительности T к одному из L $(2 \le L < +\infty)$ классов Ω_1, …, Ω_L.

Генераторы могут быть разделены на следующие классы $\{\Omega_i\}$: физические и программные генераторы; программные генераторы различных типов; программные генераторы одного типа, но с различными параметрами; программные генераторы с одинаковыми параметрами, но с различной инициализирующей информацией. Будем придерживаться классификации криптографических генераторов [2], представленной на рис. 1.

Приведем описания прореживающего и самосжимающего генераторов, которые в статье будут использоваться в компьютерных экспериментах.

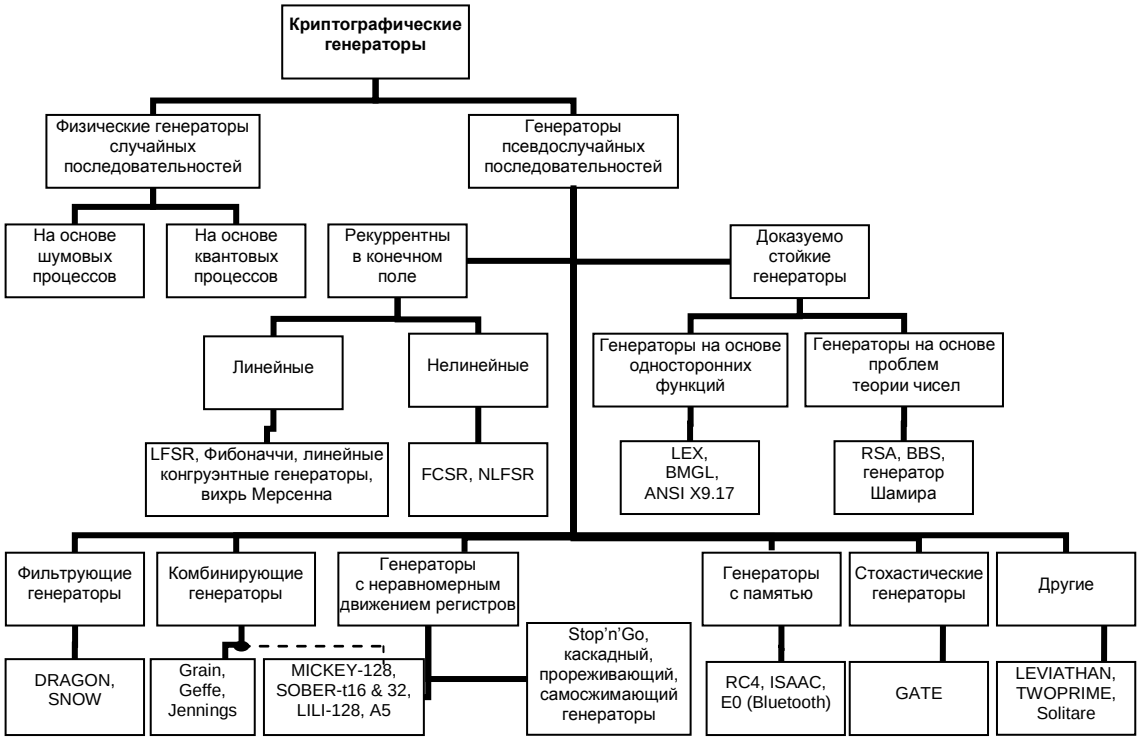


Рис. 1. Схема классификации криптографических генераторов

Пусть LFSR-генератор (регистр сдвига с линейной обратной связью) с порождающим многочленом степени L_1 генерирует «элементарную» двоичную последовательность $\{a_t\}$, а LFSR-генератор с порождающим многочленом степени L_2 генерирует двоичную «управляющую» последовательность $\{s_t\}$. С помощью этих двух последовательностей $\{a_t\}$, $\{s_t\}$ строится выходная последовательность $\{x_t\}$, которая включает те биты a_t, для которых соответствующее значение $s_t = 1$; если $s_t = 0$, то значение a_t игнорируется (отбрасывается). Такой генератор называется прореживающим [3]. Самосжимающий генератор [4] является модификацией прореживающего. Вместо генерации от двух разных регистров управляющей последовательности и последовательности, подлежащей сжатию, они обе берутся из одного и того же регистра. Регистр с порождающим многочленом степени d сдвигается на два такта. Если первый бит в паре равен 1, то выход генератора – второй бит этой пары. Если первый бит равен 0, то оба бита игнорируются и делаются еще два такта.

Как известно, при решении задачи статистического распознавания образов [5] выделяются два этапа: наиболее трудный этап построения пространства M информативных признаков $\rho_1, \ldots, \rho_M$, несущих информацию о разделимости классов $\{\Omega_i\}$, и этап построения решающего правила (разделяющих поверхностей) в пространстве найденных информативных признаков $\rho_1, \ldots, \rho_M$. Данная статья посвящена задаче построения пространства информативных признаков для статистического распознавания криптографических генераторов.

1. Подход к построению информативных признаков генераторов

Будем предполагать, что выходная последовательность генератора $x_t \in V$ является случайной последовательностью на некотором вероятностном пространстве $(\Omega, \mathcal{F}, \mathrm{P})$. Разобьем последовательность $x_1, x_2, \ldots, x_T$ на $l = [T / n]$ фрагментов (n-грамм) длины n $X^{(1)}, X^{(2)}, \ldots, X^{(l)}$, $X^{(k)} = (x_{(k-1)n+1}, \ldots, x_{kn}) \in V^n$ (если $nl < T$, $x_{nl+1}, \ldots, x_T$ не рассматриваем). Пусть наблюдается некоторая статистика $a_k(n) = f(X^{(k)})$, $k = 1, 2, \ldots, l$, при различных длинах фрагмента $n \in [n_-, n_+]$, $1 \le n_- < n_+$; $a_*(n) = E_{H_*}\{a_k(n)\}$ – математическое ожидание этой статистики при истинной гипотезе H_*; $a(n)$ – наблюдаемое выборочное среднее значение статистики:

$a(n)=\frac{1}{l}\sum_{k=1}^{l}a_k(n)$ (рис. 2). В качестве признака предлагается использовать уклонение $a(n)$ от математического ожидания в l_p-метрике $(p\in\mathbb{N})$:

$$\rho^{(p)}=\frac{1}{n_+-n_-+1}\sqrt[p]{\sum_{n=n_-}^{n_+}|a(n)-a_*(n)|^p}. \qquad (1)$$

Чем $\rho^{(p)}$ больше, тем больше свойства генератора отличаются от РРСП.

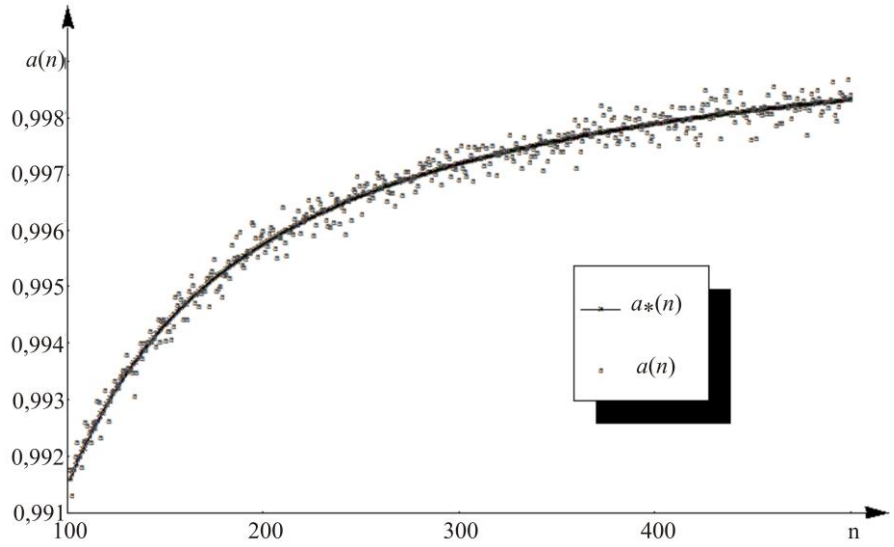


Рис. 2. Статистика $a(n)$ и ее математическое ожидание $a_*(n)$

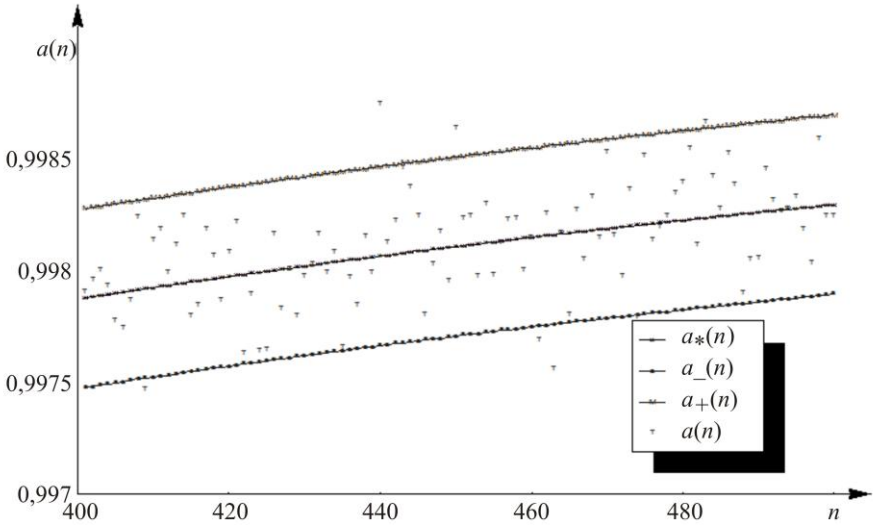


Рис. 3. Статистика $a(n)$, ее математическое ожидание $a_*(n)$ и границы допустимой окрестности

Построим некоторую допустимую окрестность $(a_-(n), a_+(n))$ ожидаемого значения $a_*(n)$: $a_-(n) < a_*(n) < a_+(n)$ – и будем учитывать выбросы за пределы этой окрестности. В качестве такой окрестности $(a_-(n), a_+(n))$ целесообразно использовать доверительный интервал заданного доверительного уровня (рис. 3). Тогда в качестве признака, учитывающего величины уклонений $a(n)$ от допустимого интервала $(a_-(n), a_+(n))$, можно взять величину

$$r = \frac{1}{n_+ - n_- + 1} \sum_{n=n_-}^{n_+} \min\{|a(n) - a_-(n)|, |a(n) - a_+(n)|\}(1 - I_{[a_-(n), a_+(n)]}(a(n))), \quad (2)$$

где $I_{[a,b]}(z) = \begin{cases} 1, z \in [a,b]; \\ 0, z \notin [a,b]. \end{cases}$

2. Нелинейные признаки на основе энтропии

Пусть $p_{i_1,\ldots,i_n} = P\{x_{t+1} = i_1, \ldots, x_{t+n} = i_n\}$ – распределение вероятностей n-граммы $(x_{t+1}, \ldots, x_{t+n}) \in V_n$, которое предполагается не зависящим от $t \in \mathbb{N}$. Многомерная (n-мерная) энтропия Шеннона для фрагмента длины n находится из выражения [1]

$$h(n) = -\sum_{i_1,\ldots,i_n} p_{i_1,\ldots,i_n} \ln p_{i_1,\ldots,i_n} . \quad (3)$$

Теорема 1. *Если справедлива гипотеза* H_*, *то*

$$h_*(n) = h(n)\big|_{H_*} = n \ln 2 = n h_*(1) . \quad (4)$$

Доказательство. При верной гипотезе имеем H_* $p_{i_1,\ldots,i_n} = 2^{-n}$, $(i_1, \ldots, i_n) \in V_n$. Поэтому согласно (3) $h_*(n) = -\sum_{i_1,\ldots,i_n} 2^{-n} \ln 2^{-n} = n \ln 2$. В силу (3) при n = 1 с учетом $p_0 = p_1 = \frac{1}{2}$ получаем $h_*(1) = -(p_0 \ln p_0 + p_1 \ln p_1) = \ln 2$, что доказывает (4). ■

Обозначим: $i = \sum_{j=1}^{n} 2^{j-1} i_j$ – представление числа $i \in \{0, 1, \ldots, 2^n - 1\}$ в двоичной системе счисления, $p_i(n) = P\{\sum_{j=1}^{n} 2^{j-1} x_j = i\} = p_{i_1,\ldots,i_n}$, $i = 0, \ldots, 2^n - 1$. Пусть наблюдается l фрагментов $X^{(1)}$, …, $X^{(l)}$. Построим статистические оценки распределения вероятностей $\{p_i(n)\}$, $i = 0, \ldots, 2^n - 1$:

$$\hat{p}_i(n) = \frac{1}{l} \sum_{k=1}^{l} \delta_{\bar{X}^{(k)}, i}, \quad \bar{X}^{(k)} = \sum_{j=1}^{n} 2^{j-1} x_j^{(k)}, \quad \delta_{\bar{X}^{(k)}, i} = \begin{cases} 1, \bar{X}^{(k)} = i; \\ 0, \bar{X}^{(k)} \neq i. \end{cases}$$

Используя подстановочный принцип, построим статистическую оценку энтропии (3):

$$\hat{h}(n) = -\sum_{i=0}^{2^n - 1} \hat{p}_i(n) \ln \hat{p}_i(n). \quad (5)$$

Теорема 2. *Если справедлива гипотеза* H_*, *то при* $l \to \infty$ *распределение вероятностей статистики* $\hat{h}(n)$ *сходится к нормальному распределению вероятностей* $\mathcal{N}(a_*(n), \sigma_*^2(n))$ *с математическим ожиданием* $a_*(n)$ *и дисперсией* $\sigma_*^2(n)$ *:*

$$a_*(n) = h_*(n), \quad \sigma_*^2(n) = \sum_{i,j=0}^{2^n - 1} (\ln p_i(n) + 1) \cdot \sigma_{ij}(n) \cdot (\ln p_j(n) + 1), \quad \sigma_{ij}(n) = \begin{cases} p_i(n) \cdot (1 - p_i(n)), i = j; \\ -p_i(n) \cdot p_j(n), i \neq j. \end{cases} \quad (6)$$

Доказательство. Свойство асимптотической нормальности и соотношение (6) вытекают из третьей теоремы непрерывности и центральной предельной теоремы [6]. ■

В качестве статистики $a(n)$ из разд. 1 можно взять оценку (5), математическое ожидание которой при истинной гипотезе H_* определяется (4). На основании статистик $\{\hat{h}(n): n_- \le n \le n_+\}$ построим информативные признаки согласно (1):

$$\rho^{(p)} = \frac{1}{n_+ - n_- + 1} \sqrt[p]{\sum_{n=n_-}^{n_+} |\hat{h}(n) - h_*(n)|^p} \,. \tag{7}$$

Для построения информативного признака согласно (2) целесообразно использовать найденное выражение дисперсии (6).

Метод оценивания многомерной энтропии согласно (5) требует вычислительных затрат порядка $O(2^n)$, для снижения которых в [7] предложено вычислять многомерную энтропию, используя расстояние Хэмминга. Опираясь на эту идею, предложим следующий подход к построению информативных признаков на основе энтропии.

Используя k-й фрагмент $X^{(k)} = (x_1^{(k)}, \ldots, x_n^{(k)}) \in V^n$ выходной последовательности и вес Хэмминга $W(x_1, ..., x_n) = \sum_{t=1}^{n} x_t$, определим семейство целочисленных статистик:

$$w_1^{(k)} = w_1(X^{(k)}) = W(X^{(k)}) = \sum_{i=1}^{n} x_i^{(k)}; \; w_2^{(k)} = w_2(X^{(k)}) = W((x_1^{(k)}x_2^{(k)}, x_1^{(k)}x_3^{(k)}, ..., x_{n-1}^{(k)}x_n^{(k)})) = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} x_i x_j; ...;$$

$$w_m^{(k)} = w_m(X^{(k)}) = W((x_1^{(k)}x_2^{(k)}...x_m^{(k)}, ..., x_{n-m+1}^{(k)}...x_{n-1}^{(k)}x_n^{(k)})) = \sum_{1 \le i_1 < i_2 < ... < i_m \le n} x_{i_1}^{(k)} x_{i_2}^{(k)} ... x_{i_m}^{(k)} \in \{0\} \cup \{C_i^m : m \le i \le n\}.$$

При $m = 1$ получаем частный случай, предложенный в [7]. Множество значений, которые могут принимать величины $w_j^{(k)}$, обозначим как $W_j = \{0\} \cup \{C_i^j : j \le i \le n\}$.

Теорема 3. *Если справедлива гипотеза H_*, то*

$$E_{H_*}\{w_j^{(k)}\} = \frac{C_n^j}{2^j}, \quad j = 1, 2, \ldots, m.$$

Доказательство. Математическое ожидание $w_j^{(k)}$ не зависит от k, поэтому для удобства записи индекс k в дальнейшем будем опускать. Исходя из определения и свойств математического ожидания, получим

$$E_{H_*}\{w_j\} = \sum_{1 \le i_1 < ... < i_j \le n} E_{H_*}\{x_{i_1} \ldots x_{i_j}\} = \sum_{1 \le i_1 < ... < i_j \le n} P\{x_{i_1} = \ldots = x_{i_j} = 1\} = \sum_{1 \le i_1 < ... < i_j \le n} 2^{-j} = \frac{C_n^j}{2^j}, \quad j = 1, \ldots, m. \; ■$$

Пусть $\tilde{p}_{i_1,...,i_m}(n) = P\{w_1 = i_1, ..., w_m = i_m\}$. Тогда величину

$$\tilde{h}^{(m)}(n) = -\sum_{i_1,\ldots,i_m} \tilde{p}_{i_1,\ldots,i_m}(n) \ln \tilde{p}_{i_1,\ldots,i_m}(n) \tag{8}$$

будем называть n-мерной псевдоэнтропией m-го типа.

Пусть наблюдается l фрагментов $X^{(1)}, \ldots, X^{(l)}$. Построим статистические оценки распределения вероятностей $p_i^{(j)}(n) = P\{w_j = i\}, \quad i \in W_j, \; j = 1, \ldots, m$:

$$\hat{p}_i^{(j)}(n) = \frac{1}{l}\sum_{k=1}^{l} \delta_{w_j^{(k)}, i}.$$

Используя подстановочный принцип, построим статистическую оценку псевдоэнтропии (8):

$$\widehat{\tilde{h}^{(m)}}(n) = -\sum_{i_1 \in W_1, \ldots, i_m \in W_m} \hat{p}_{i_1}^{(1)}(n) \ldots \hat{p}_{i_m}^{(m)}(n) \ln(\hat{p}_{i_1}^{(1)}(n) \ldots \hat{p}_{i_m}^{(m)}(n)).$$

Теорема 4. *Если справедлива гипотеза H_*, то статистическая оценка n-мерной псевдоэнтропии 1-го типа находится из выражения*

$$\tilde{h}_*(n) = \tilde{h}^{(1)}(n)\Big|_{H_*} = -\sum_{i=0}^{n} \frac{C_n^i}{2^n} \ln \frac{C_n^i}{2^n}.$$

Доказательство. Так как при верной гипотезе H_* вероятность $P\{x_t = 1\} = P\{x_t = 0\} = \frac{1}{2}$, то $p_i^{(1)}(n) = P\{w_1(X) = i\} = \frac{C_n^i}{2^n}$. Следовательно, $\tilde{h}^{(1)}(n) = -\sum_{i=0}^{n} p_i^{(1)}(n) \ln p_i^{(1)}(n) = -\sum_{i=0}^{n} \frac{C_n^i}{2^n} \ln \frac{C_n^i}{2^n}$. ■

Теорема 5. *Если справедлива гипотеза H_*, то при $l \to \infty$ распределение вероятностей статистики $\widehat{\tilde{h}^{(1)}}(n)$ сходится к нормальному распределению вероятностей $\mathcal{N}(a_*(n), \sigma_*^2(n))$:*

$$a_*(n) = \tilde{h}_*(n),\ \sigma_*^2(n) = \sum_{i,j=0}^{n} (\ln p_i(n) + 1) \cdot \sigma_{ij}(n) \cdot (\ln p_j(n) + 1),\ \sigma_{ij}(n) = \begin{cases} p_i(n) \cdot (1 - p_i(n)), i = j; \\ -p_i(n) \cdot p_j(n), i \neq j. \end{cases}$$

Доказательство. Теорема доказывается аналогично теореме 2. ■

Информативные признаки согласно (2) и теореме 4 имеют вид

$$\rho^{(p)} = \frac{1}{n_+ - n_- + 1} \sqrt[p]{\sum_{n=n_-}^{n_+} |\widehat{\tilde{h}^{(1)}}(n) - \tilde{h}_*(n)|^p}.$$

Для построения информативного признака согласно (2) предлагается использовать выражение дисперсии из теоремы 5.

3. Нелинейные признаки на основе рангов матриц

Пусть на $V_n = \{0, 1\}^n$ определены $N \le n$ линейно независимых над V_n двоичных функций от n двоичных переменных $\psi_1(x_1, \ldots, x_n) \in V$, …, $\psi_N(x_1, \ldots, x_n) \in V$. Разбиваем наблюдаемый ряд $x_1, x_2, \ldots$ на фрагменты $X^{(1)}, X^{(2)}, \ldots \in V^{nN}$ длины nN: $X^{(k)} = (x_{(k-1)nN+1}, \ldots, x_{knN})$. Используя k-й фрагмент $X^{(k)} = (x_1^{(k)}, \ldots, x_{nN}^{(k)}) \in V^{nN}$ выходной последовательности, построим $(N \times N)$-матрицу

$$A^{(k)} = (a_{ij}^{(k)}) = \begin{pmatrix} \psi_1(x_1^{(k)}, \ldots, x_n^{(k)}) & \ldots \psi_N(x_1^{(k)}, \ldots, x_n^{(k)}) \\ \psi_1(x_{n+1}^{(k)}, \ldots, x_{2n}^{(k)}) & \ldots \psi_N(x_{n+1}^{(k)}, \ldots, x_{2n}^{(k)}) \\ \ldots & \ldots \quad \ldots \\ \psi_1(x_{(N-1)n+1}^{(k)}, \ldots, x_{Nn}^{(k)}) \ldots & \psi_N(x_{(N-1)n+1}^{(k)}, \ldots, x_{Nn}^{(k)}) \end{pmatrix} \in V^{N \times N} \qquad (9)$$

или поэлементно $a_{ij}^{(k)} = \psi_j(x_{(i-1)n+1}^{(k)}, \ldots, x_{in}^{(k)}) \in V, \quad i, j \in \{1, 2, \ldots, N\}$.

Ранг матрицы (9) отражает наличие функциональной зависимости в последовательности. Если $\mathrm{rank}(A^{(k)}) = r < N$, то в матрице $A^{(k)}$ имеется $N - r$ линейно зависимых над V_n строк. В рас-

сматриваемом случае это означает, что найдена зависимость элементов последовательности от предыдущих:

$$\psi_j(x^{(k)}_{(i-1)n+1}, ..., x^{(k)}_{in}) = \bigoplus_{m=1}^{r} \alpha_m \psi_j(x^{(k)}_{(m-1)n+1}, ..., x^{(k)}_{mn}), \quad i \in \{r+1, ..., N\},\ j \in \{1, ..., N\},\ \alpha_m \in V = \{0, 1\}.$$

За счет выбора функций $\psi_1, \ldots, \psi_N$ можно добиться выявления этой зависимости. Поэтому в качестве признака для фрагмента $X^{(k)}$ будем использовать ранг матрицы (9):

$$r^{(k)} = rank(A^{(k)}) \in \{0, 1, ..., \min\{n, N\}\}.$$

Рассмотрим более подробно частный случай: $N = n$, $\psi_j(x_1, \ldots, x_n) = x_j$. Тогда $a^{(k)}_{ij} = x^{(k)}_{(i-1)n+j}$. Пусть наблюдается l фрагментов, т. е. $T = l \cdot N^2$. Определим статистику

$$v(n) = \frac{1}{nl} \sum_{k=1}^{l} r^{(k)}, \tag{10}$$

имеющую смысл среднего относительного ранга.

Теорема 6. *При верной гипотезе H_* математическое ожидание и дисперсия статистики* (*10*) *имеют вид*

$$E_{H_*}\{v(n)\} = v_*(n) = \frac{1}{n} \sum_{j=0}^{n} q_{nj}\, j; \tag{11}$$

$$D_{H_*}\{v(n)\} = \frac{1}{l}\left(\frac{1}{n^2} \sum_{j=0}^{n} q_{nj}\, j(j - \sum_{k=0}^{n} q_{nk} k) \right), \tag{12}$$

где $q_{nj} = P_{H_0}\{r^{(k)} = j\} = 2^{j(2n-j)-n^2} \prod_{i=0}^{j-1} \frac{(1-2^{i-n})^2}{1-2^{i-j}}, \quad j \in \{0, 1, ..., n\}$.

Доказательство. При верной гипотезе H_* статистика $r^{(k)}$ имеет известное распределение вероятностей [1]:

$$q_{nj} = P_{H_0}\{r^{(k)} = j\} = 2^{j(2n-j)-n^2} \prod_{i=0}^{j-1} \frac{(1-2^{i-n})^2}{1-2^{i-j}},\ j \in \{0, 1, ..., n\}.$$

Тогда $E_{H_*}\{v(n)\} = E_{H_*}\left\{ \frac{1}{nl} \sum_{k=1}^{l} r^{(k)} \right\} = \frac{1}{n}\left(\frac{1}{l} \sum_{k=1}^{l} E_{H_*}\{r^{(k)}\} \right) = \frac{1}{n} E_{H_*}\{r^{(k)}\} = \frac{1}{n} \sum_{j=0}^{n} q_{nj}\, j.$

Математическое ожидание $r^{(k)}$ не зависит от k, поэтому для удобства записи индекс k будем далее опускать при вычислении вероятностных характеристик $r^{(k)}$. Используя свойства дисперсии и учитывая независимость $r^{(k)}$, получим $D_{H_*}\{v(n)\} = D_{H_*}\left\{ \frac{1}{l} \sum_{k=1}^{l} \frac{r^{(k)}}{n} \right\} = \frac{1}{l} D_{H_*}\left\{ \frac{r}{n} \right\}$.

Из определения дисперсии и (11) имеем $D_{H_*}\{v(n)\} = \frac{1}{l}\left(E_{H_*}\left\{ \frac{r^2}{n^2} \right\} - E^2_{H_*}\left\{ \frac{r}{n} \right\} \right) =$

$= \frac{1}{l}\left(\frac{1}{n^2} \sum_{j=0}^{n} q_{nj}\, j^2 - E^2_{H_*}\{v(n)\} \right) = \frac{1}{l}\left(\frac{1}{n^2} \sum_{j=0}^{n} q_{nj}\, j(j - \sum_{k=0}^{n} q_{nk} k) \right)$. ■

На основании статистик $\{v(n): n_- \le n \le n_+\}$ построим информативные признаки согласно (1):

$$\rho^{(p)} = \frac{1}{n_+ - n_- + 1} \sqrt[p]{\sum_{n=n_-}^{n_+} | v(n) - v_*(n) |^p}. \tag{13}$$

Для построения информативного признака согласно (2) целесообразно использовать найденное выражение дисперсии (12).

4. Признаки, основанные на определителях матриц

Определитель матрицы позволяет учитывать зависимости между элементами матрицы. Для наблюдаемой двоичной последовательности $x_1, x_2, ..., x_T \in V = \{0, 1\}$ вычислим следующие статистики, основанные на детерминантах $(n, T \in \mathbb{N}, \quad T >> n^2)$:

$$\begin{aligned} &\alpha(1) = \frac{1}{T}\sum_{t=1}^{T} x_t, \quad \alpha(2) = \frac{1}{T-3}\sum_{t=1}^{T-3} D_t^{(2)}, D_t^{(2)} = \begin{vmatrix} x_t & x_{t+1} \\ x_{t+2} & x_{t+3} \end{vmatrix}, ..., \\ &\alpha(n) = \frac{1}{T-n^2+1}\sum_{t=1}^{T-n^2+1} D_t^{(n)}, D_t^{(n)} = \begin{vmatrix} x_t & x_{t+1} & \dots & x_{t+n-1} \\ x_{t+n} & x_{t+n+1} & \dots & x_{t+2n-1} \\ \dots & \dots & \dots & \dots \\ x_{t+(n-1)n} & x_{t+(n-1)n+1} & \dots & x_{t+n^2-1} \end{vmatrix}. \end{aligned} \tag{14}$$

Вычислим также выборочное среднеквадратичное отклонение для $\{D_t^{(n)}\}$:

$$s_n = \sqrt{\frac{1}{(T-n^2)(T-n^2+1)}\sum_{t=1}^{T-n^2+1} (D_t^{(n)} - \alpha(n))^2}. \tag{15}$$

Исследуем вероятностные характеристики статистики (14) для построения признаков согласно (1), (2). Обозначим: $\alpha_*(n) = E_{H_*}\{D_t^{(n)}\}, n \in \mathbb{N}$.

Теорема 7. *При верной гипотезе H_* математическое ожидание случайного детерминанта из (14) определяется следующими соотношениями:*

$$\alpha_*(1) = \frac{1}{2}, \quad \alpha_*(n) = 0, n = 2,\ 3,\ \dots, \quad t = 1,\ 2,\ \dots \tag{16}$$

Доказательство. $E_{H_*}\{D_t^{(1)}\} = E\{x_t\} = \frac{1}{2}$. Пусть теперь $i_1, \dots, i_n$ – перестановка чисел от 1 до n, $n > 1$; $N(i_1,\dots,i_n) \in \{0,1\}$ – четность перестановки, т. е. четность числа ее инверсий [8]. Воспользуемся тем фактом, что число четных перестановок равно числу нечетных перестановок, получим

$$E_{H_*}\{D_t^{(n)}\} = E_{H_*}\left\{\sum_{i_1,\dots,i_n} (-1)^{N(i_1,\dots,i_n)} x_{t-1+(i_1-1)n+1} \dots x_{t-1+(i_n-1)n+n}\right\} = \sum_{i_1,\dots,i_n} (-1)^{N(i_1,\dots,i_n)} E_{H_*}\left\{x_{t+(i_1-1)n} \dots x_{t-1+(i_n-1)n+n}\right\} =$$

$$= \sum_{i_1,\dots,i_n} (-1)^{N(i_1,\dots,i_n)} P\left\{x_{t+(i_1-1)n} = \dots = x_{t-1+(i_n-1)n+n} = 1\right\} = \sum_{i_1,\dots,i_n} (-1)^{N(i_1,\dots,i_n)} 2^{-n} = 2^{-n} \sum_{i_1,\dots,i_n} (-1)^{N(i_1,\dots,i_n)} = 0. \blacksquare$$

На основании статистик $\{\alpha(n): n_- \le n \le n_+\}$ построим информативные признаки согласно (1):

$$\rho^{(p)} = \frac{1}{n_+ - n_- + 1} \sqrt[p]{\sum_{n=n_-}^{n_+} |\alpha(n) - \alpha_*(n)|^p}.$$

Для построения информативного признака согласно (2) целесообразно использовать выборочное среднеквадратичное отклонение (15).

5. Нелинейные признаки на основе дискретного преобразования Фурье

Разбиваем наблюдаемый ряд $x_1, x_2, \ldots$ на фрагменты $X^{(1)}, X^{(2)}, \ldots \in V^{2^n}$ длины 2^n. Далее фрагмент $X^{(k)} = (x_1^{(k)}, \ldots, x_{2^n}^{(k)}) \in V^{2^n}$ подвергаем дискретному преобразованию Фурье (ДПФ) [9]:

$$X^{(k)} \leftrightarrow Y^{(k)} = \begin{pmatrix} y_1^{(k)} \\ \vdots \\ y_{2^n}^{(k)} \end{pmatrix} \in \mathbb{R}^{2^n}, k = 1,\ 2,\ \ldots. \tag{17}$$

Преобразуем $Y^{(k)}$ в вектор-столбец $U^{(k)} = (u_i^{(k)}) \in \mathbb{R}^n$ при помощи нелинейного преобразования

$$u_1^{(k)} = y_1^{(k)}, u_i^{(k)} = (y_i^{(k)})^2,\ i = 2, 3, ..., 2^n. \tag{18}$$

Пусть наблюдается l фрагментов $U^{(1)}, \ldots, U^{(l)}$. Вычислим выборочную ковариационную матрицу

$$\hat{\Sigma} = (\hat{\sigma}_{ij}) = \frac{1}{l-1} \sum_{k=1}^{l} (U_k - \bar{U})(U_k - \bar{U})', \bar{U} = (\bar{u}_i) = \frac{1}{l} \sum_{k=1}^{l} U_k, \tag{19}$$

или поэлементно

$$\hat{\sigma}_{ij} = \frac{1}{l-1} \sum_{k=1}^{l} (u_{ki} - \bar{u}_i)(u_{kj} - \bar{u}_j),\ i, j = 1,\ ..., 2^n.$$

Оценим собственные значения $0 \le \hat{\lambda}_1 \le ... \le \hat{\lambda}_{2^n}$ выборочной ковариационной матрицы (19) и соответствующие им собственные векторы:

$$\hat{v}_1 = (\hat{v}_{11}, ..., \hat{v}_{1\,2^n})', ..., \hat{v}_{2^n} = (\hat{v}_{2^n 1}, ..., \hat{v}_{2^n 2^n})'. \tag{20}$$

Используя (18) и (20), определим статистики исходя из свойств собственных значений и векторов ковариационной матрицы для наблюдаемого ряда:

$$T^{(1)}(U) = \hat{v}_{11}(u_1 - \bar{u}_1) + ... + \hat{v}_{1\,2^n}(u_{2^n} - \bar{u}_{2^n}), \tag{21}$$

где используется собственный вектор $\hat{v}_1$, соответствующий наименьшему собственному значению $\hat{\lambda}_1$. Построим по наблюдаемому ряду выборку из значений статистик (21):

$$t_k^{(1)} = T^{(1)}(U(X^{(k)})), k = 1, ..., l. \tag{22}$$

Для выборки (22) вычислим выборочное среднее и выборочную дисперсию:

$$\overline{t}^{(1)} = \frac{1}{l}\sum_{k=1}^{l} t_k^{(1)},\ \ s^{2(1)} = \frac{1}{l-1}\sum_{k=1}^{l}(t_k^{(1)} - \overline{t}^{(1)})^2. \tag{23}$$

Статистики (23) можно использовать для построения признаков согласно (1) и (2), а именно $\overline{t}^{(1)}$ – в качестве статистики $a(n)$, а $s^{2(1)}$ – для построения допустимой окрестности. При этом полагаем $a_*(n) = 0, n \in \mathbb{N}$.

6. Решающее правило

Пусть методами из разд. 1 – 5 данной статьи построено пространство из M информативных признаков $(\rho_1, \ldots, \rho_M) \in \mathbb{R}^M$, где количество признаков M и конкретный набор признаков из множества признаков, построенных ранее, определяются исходя из перечня криптографических генераторов, подлежащих распознаванию.

Опишем применение метода дискриминантного анализа [6, 10] для решения задачи распознавания в случае двух классов, когда признаки имеют нормальное распределение. Обоснованием предположения нормальности распределения вероятностей признаков является общий вид признаков (2.1), допускающий применение центральной предельной теоремы при $n_+ >> 1$. Пусть по входной последовательности построен вектор признаков ρ и имеется $m_i > M$ обучающих реализаций $\rho_1^{(i)}, \ldots, \rho_{m_i}^{(i)} \in \mathbb{R}^M$, принадлежащих классу Ω_i, $i \in \{1, 2\}$. Построим оценки математического ожидания вектора признаков и ковариационной матрицы для каждого из классов. Оценка для математического ожидания в классе Ω_i есть выборочное среднее:

$$\hat{\mu}_i = \frac{1}{m_i}\sum_{m=1}^{m_i}\rho_m^{(i)}, \tag{24}$$

а оценка для ковариационной матрицы – выборочная ковариационная матрица:

$$\hat{\Sigma}_i = \frac{1}{m_i - 1}\sum_{m=1}^{m_i}(\rho_m^{(i)} - \hat{\mu}_i)(\rho_m^{(i)} - \hat{\mu}_i)'. \tag{25}$$

Байесовское решающее правило имеет вид [10]

$$d = d_0(\rho) = \arg\min_{i\in\{1,2\}}(\ln|\hat{\Sigma}_i| + (\rho - \hat{\mu}_i)'\hat{\Sigma}_i^{-1}(\rho - \hat{\mu}_i) - 2\ln\hat{\pi}_i), \quad \rho \in \mathbb{R}^M, \tag{26}$$

где оценки $\{\hat{\mu}_i\}$ и $\{\hat{\Sigma}_i\}$ вычисляются по формулам (24) и (25) соответственно, априорные вероятности классов полагаем равными, т. е. $\hat{\pi}_i = P\{\rho \in \Omega_i\} = \frac{1}{2}$, $i \in \{1, 2\}$.

В частности, при равных ковариационных матрицах $\Sigma_1 = \Sigma_2 = \Sigma$ получаем линейное решающее правило

$$d = d_0(\rho) = \arg\min_{i\in\{1,2\}}((\hat{\mu}_i - 2\rho)'\hat{\Sigma}^{-1}\hat{\mu}_i), \quad \rho \in \mathbb{R}^M. \tag{27}$$

7. Результаты компьютерных экспериментов

Проведены две серии компьютерных экспериментов.

Серия 1. Рассмотрим задачу распознавания двух описанных выше криптографических генераторов: Ω_1 – самосжимающий генератор псевдослучайных двоичных последовательностей с порождающим примитивным многочленом порождающего регистра сдвига $f_1(x) = x^{32} + x^{16} + x^7 + x^2 + 1$; Ω_2 – самосжимающий генератор с порождающим примитивным многочленом $f_2(x) = x^{63} + x^{54} + x^{44} + x^{20} + 1$.

В качестве признака ρ используется статистика (6). Таким образом, имеем одномерное пространство признаков ($M = 1$). Для каждого из классов Ω_1 и Ω_2 сгенерированы 40 последова-

тельностей одинаковой длины $T = 10^7$, которые отличаются начальным заполнением порождающего регистра сдвига. По две последовательности от каждого класса ($m_1 = m_2 = 2$) используются для обучения, на остальных 38 оценивается вероятность ошибки распознавания.

Заданы параметры $n_- = 1$, $n_+ = 20$, $p = 1$. Получены следующие значения оценок параметров (24), (25):

$$\hat{\mu}_1 = 0{,}1050088479,\ \hat{\Sigma}_1 = (0{,}3208798039\cdot 10^{-8});$$
$$\hat{\mu}_2 = 0{,}1048425232,\ \hat{\Sigma}_2 = (0{,}3079484568\cdot 10^{-8}).$$

Признаки (7) имеют асимптотически нормальное распределение при $l \to \infty$. Это следует из теоремы о функциональном преобразовании нормально распределенных величин [10] и теоремы 2. Разделимость классов Ω_1, Ω_2 по информативному признаку (7) проиллюстрирована на рис. 4, где темно-серый цвет соответствует Ω_1, светло-серый – Ω_2, серый – пересечению классов. Для распознавания применяется решающее правило (26), которое при $M = 1$ принимает следующий вид:

$$d(\rho) = \begin{cases} 1, & \rho \in [0;\, 0{,}09683808) \cup (0{,}104925219;\, +\infty); \\ 2, & \rho \in [0{,}09683808;\quad 0{,}104925219]. \end{cases}$$

Результаты распознавания представлены в табл. 1, оценка безусловной вероятности ошибки равна 0,22.

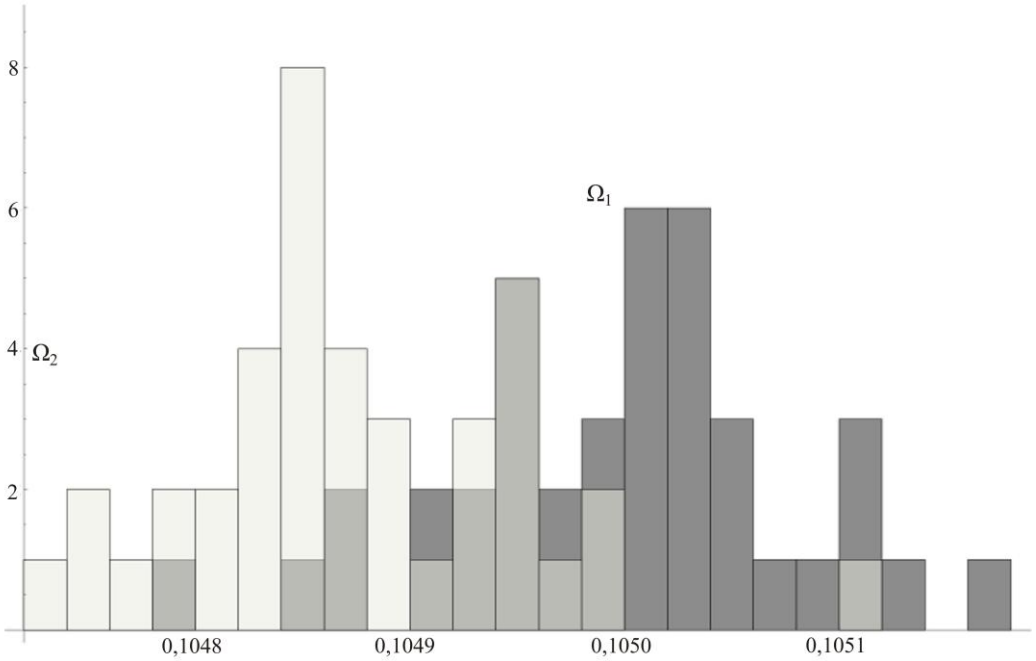


Рис. 4. Гистограммы значений признаков классов Ω_1 и Ω_2

Таблица 1

Результаты распознавания

Класс	Число решений в пользу Ω_1	Число решений в пользу Ω_2	Условная вероятность ошибки
Ω_1	32	6	0,16
Ω_2	11	27	0,29

Серия 2. Рассмотрим также задачу распознавания двух описанных ранее криптографических генераторов. Пусть Ω_1 – прореживающий генератор псевдослучайных двоичных последовательностей с порождающим примитивным многочленом 63-й степени порождающего реги-

стра сдвига и порождающим многочленом $f(x) = x^{13} + x^8 + x^5 + x^3 + 1$ управляющего регистра сдвига; Ω_2 – самосжимающий генератор псевдослучайных двоичных последовательностей с порождающим примитивным многочленом 63-й степени порождающего регистра сдвига.

В качестве признака ρ используется значение величины (13). Таким образом, имеем одномерное пространство признаков ($M = 1$). Для каждого из классов Ω_1 и Ω_2 сгенерированы 30 последовательностей по следующему принципу. В качестве характеристических многочленов порождающих регистров сдвига используются три примитивных многочлена: $g_1(x) = x^{63} + x + 1$, $g_2(x) = x^{63} + x^{54} + x^{44} + x^{20} + 1$, $g_3(x) = x^{63} + x^{60} + x^{22} + x^{18} + x^8 + x^5 + 1$. Для каждого многочлена в каждом из классов сгенерированы 10 последовательностей одинаковой длины $T = 10^7$, которые отличаются начальным заполнением порождающего регистра сдвига. Для обучения используется по одной последовательности от каждого многочлена, т. е. для каждого из классов имеем три обучающие реализации ($m_1 = m_2 = 3$). Оставшиеся 27 реализаций каждого из классов используются для оценивания вероятности ошибки распознавания.

Заданы параметры $n_- = 1$, $n_+ = 2236$, $p = 1$. Получены следующие значения оценок параметров (24), (25):

$$\hat{\mu}_1 = 0{,}001575833409,\ \hat{\Sigma}_1 = (0{,}1120998871\cdot 10^{-8});$$

$$\hat{\mu}_2 = 0{,}0001741820352,\ \hat{\Sigma}_2 = (0{,}706244773\cdot 10^{-12}).$$

Можно применить метод дискриминантного анализа, описанный в разд. 6. Однако вид гистограммы значений построенных признаков (рис. 5) указывает на то, что можно применить более простое линейное решающее правило:

$$d(\rho) = \begin{cases} 1, & \rho > \dfrac{\hat{\mu}_1 + \hat{\mu}_2}{2} = 0{,}008750077221; \\ 2, & \rho \le \dfrac{\hat{\mu}_1 + \hat{\mu}_2}{2} = 0{,}008750077221. \end{cases}$$

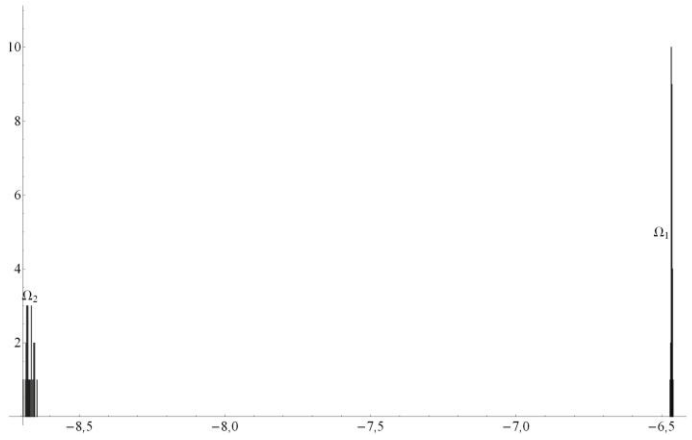


Рис. 5. Гистограмма значений признаков классов Ω_1 и Ω_2 в логарифмической шкале

Результаты распознавания представлены в табл. 2, оценка безусловной вероятности ошибки равна 0.

Таблица 2

Результаты распознавания

Класс	Число решений в пользу Ω_1	Число решений в пользу Ω_2	Условная вероятность ошибки
Ω_1	27	0	0
Ω_2	0	27	0

Заключение

Для решения актуальной задачи распознавания криптографических генераторов разработан общий подход к построению информативных признаков. Этот подход учитывает динамику изменения вероятностных характеристик генераторов при изменении длины применяемого фрагмента. Построенные согласно данному подходу признаки универсальны, так как не используют специфических свойств генераторов, и поэтому их можно применять для распознавания различных типов генераторов. Проиллюстрировано применение одного из признаков для распознавания прореживающего и самосжимающего генераторов, другого – для распознавания самосжимающих генераторов с различными параметрами.

Список литературы

1. Математические и компьютерные основы криптологии / Ю.С. Харин [и др.]. – Минск : Новое знание, 2003. – 382 с.

2. Харин, Ю.С. Проблемы математики и информатики в области защиты информации / Ю.С. Харин // Весці Нацыянальнай акадэміі навук Беларусі. Сер. фіз.-мат. навук. – 2007. – № 4. – С. 84–95.

3. Coppersmith, D. The shrinking generator / D. Coppersmith, Y. Krawchuk, Y. Mansour // Advanced in Cryptology: Proc. of Crypto 93, LNCS 773. – Berlin : Springer-Verlag, 1994. – P. 22–39.

4. Meier, W. Analysis of pseudo random sequences generated by cellular automata / W. Meier, O. Staffelbach // Advances in Cryptology / Eurocrypt '91. – Berlin : Springer-Verlag, 1992. – P. 186–199.

5. Kharin, Y. Robustness in Statistical Pattern Recognition / Y. Kharin. – Dordrecht : Kluwer Academic Publishers, 1996. – 302 p.

6. Боровков, А.А. Математическая статистика / А.А. Боровков. – Новосибирск : Наука, 1997. – 772 с.

7. Куликов, С.В. Оценка энтропии биометрических образов через переход к дискретному представлению с асимметричным распределением меры Хемминга / С.В. Куликов // Труды науч.-техн. конф. кластера пензенских предприятий, обеспечивающих безопасность информационных технологий. – Пенза, 2012. – Т. 8. – С. 18–21.

8. Милованов, М.В. Алгебра и аналитическая геометрия / М.В. Милованов, Р.И. Тышкевич, А.С. Феденко. – Минск : Вышэйшая школа, 1984. – 302 с.

9. Лукин, А.С. Введение в цифровую обработку сигналов (математические основы) / А.С. Лукин. – М. : МГУ, 2007. – 54 с.

10. Харин, Ю.С. Математическая и прикладная статистика / Ю.С. Харин, Е.Е. Жук. – Минск : БГУ, 2004. – 272 с.

Поступила 17.04.2013

Учреждение БГУ «НИИ прикладных проблем математики и информатики», Минск, пр. Независимости, 4
e-mail: kharin@bsu.by,
fpm.paluha@bsu.by

Yu.S. Kharin, V.Yu. Palukha

INFORMATIVE DESCRIPTORS FOR STATISTICAL RECOGNITION OF CRYPTOGRAPHIC GENERATORS

An approach to developing informative descriptors for solving the problem of statistical recognition of cryptographic generators is proposed. The descriptors developed by this approach are given. Theoretical results are illustrated by the computer experiments.

ПРАВИЛА ДЛЯ АВТОРОВ

1. Статьи принимаются в редакцию через электронную систему подачи по адресу **http://jinfo.bas-net.by** в формате файлов текстовых редакторов Microsoft Word 97 и Word 2000 для Windows. Основной текст статьи набирается с переносами шрифтом Times New Roman 11 пт, интервал между строками – одинарный, абзацный отступ 1 см, поля по 2,5 см со всех сторон.

2. Объем статьи не должен превышать 12 страниц (включая таблицы, иллюстрации, список литературы), количество иллюстраций – не больше пяти. Допускаются краткие сообщения до трех страниц.

3. Статья должна иметь индекс УДК (универсальная десятичная классификация).

4. Название статьи, фамилии всех авторов и аннотация должны быть переведены на английский язык. Для каждого из авторов приводится развернутое название учреждения с полным почтовым адресом, а также номер телефона и электронный адрес (e-mail) для связи с редакцией.

5. Формулы, иллюстрации, таблицы, встречающиеся в статье, должны быть пронумерованы в соответствии с порядком цитирования в тексте. Ссылки на рисунки и таблицы в тексте обязательны. Необходимо избегать повторения одних и тех же данных в таблицах, графиках и тексте статьи.

Рисунки должны быть представлены в виде файлов формата .cdr, .ai, .wmf, .psd, .jpg, .tif (.tiff) и выполнены с хорошим разрешением в масштабе, позволяющем четко различать надписи и обозначения. Подрисуночные подписи с расшифровкой всех позиций, представленных на рисунке, набираются шрифтом гарнитуры основного текста, размер символов 9 пт. Цветные иллюстрации печатаются только в том случае, когда это необходимо для понимания излагаемого материала.

6. Набор формул выполняется в формульных редакторах Microsoft Equation или Math Type и должен быть единообразным по применению шрифтов и знаков по всей статье.

Прямо (_) набираются: греческие и русские буквы; математические символы (sin, lg, ∞); символы химических элементов (C, Cl, $CHCl_3$); цифры (римские и арабские); векторы; индексы (верхние и нижние), являющиеся сокращениями слов.

Курсивом (~) набираются: латинские буквы – переменные, символы физических величин (в том числе и в индексе).

7. Сокращения в тексте статьи (за исключением единиц измерения) могут быть использованы только после упоминания полного термина. Единицы измерения физических величин следует приводить в Международной системе СИ.

8. Литература приводится автором общим списком в конце статьи. Ссылки на литературу в тексте идут по порядку и обозначаются цифрой в квадратных скобках. Ссылаться на неопубликованные работы не допускается. С примерами оформления библиографического описания в списке литературы можно ознакомиться в приложении 2 к *Инструкции по оформлению диссертации, автореферата и публикаций по теме диссертации* на сайте Высшей аттестационной комиссии Республики Беларусь **http://vak.org.by**.

9. Поступившие в редакцию статьи направляются на рецензирование специалистам. Основным критерием целесообразности публикации является новизна и информативность статьи. Если по рекомендациям рецензента статья возвращается автору на доработку, а переработанная рукопись вновь рассматривается редколлегией, датой поступления считается день получения редакцией ее окончательного варианта. Статьи не по профилю журнала возвращаются авторам после заключения редколлегии.

10. Статьи, направляемые на доработку, должны быть возвращены в исправленном виде с ответами на все вопросы.

11. Редакция журнала предоставляет возможность первоочередного опубликования статей, представленных лицами, которые осуществляют послевузовское обучение (аспирантура, докторантура, соискательство) в год завершения обучения.

12. Авторы несут ответственность за направление в редакцию статей, уже опубликованных ранее, или статей, принятых к публикации другими изданиями.

13. Редакция оставляет за собой право на редакционные изменения, не искажающие основное содержание статьи.

Журнал «Информатика» включен Высшей аттестационной комиссией Республики Беларусь в список научных изданий для опубликования результатов диссертационных исследований.

Индексы

00827

для индивидуальных подписчиков

008272

для предприятий и организаций