

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ,
МОЛОДЕЖИ И СПОРТА УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ
УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

ISSN 0135-1710

АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ И ПРИБОРЫ АВТОМАТИКИ

**Всеукраинский межведомственный
научно-технический сборник**

Основан в 1965 г.

Выпуск 157

Харьков
2011

СОДЕРЖАНИЕ

ЧУМАЧЕНКО С.В., ХАХАНОВ В.И., МУРАД АЛИ АББАС, ГОРОБЕЦ О.А. ГЕНЕРИРОВАНИЕ БУЛЕАНА ДЛЯ СИНТЕЗА КВАНТОВОГО ПРОЦЕССОРА.....	4
ВАРЕЦА В.В., ЛОПУХИН Ю.В., КАКУРИН Н.Я., КОБЫЛИН О.А., МАКАРЕНКО А.Н. СТАТИСТИЧЕСКИЙ ПОДХОД К ВЫБОРУ ШАГОВ ПРЕОБРАЗОВАНИЯ В ПРЕОБРАЗОВАТЕЛЯХ КОДОВ ПО МЕТОДУ НАКОПЛЕНИЯ ЭКВИВАЛЕНТОВ.....	17
БАРАННИК В.В., КРАСНОРУЦКИЙ А.А., ХАХАНОВА А.В. ПОЗИЦИОННОЕ СТРУКТУРНО-ВЕСОВОЕ КОДИРОВАНИЕ БИНАРНОГО ПРЕДСТАВЛЕНИЯ ТРАНСФОРМАНТ.....	23
АЛЬХАЙЕК Р., УДОВЕНКО С.Г., ШАМРАЕВ А.А. ОПТИМИЗАЦИЯ РЕЖИМОВ АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ КОМПРЕССОРНОЙ СТАНЦИЕЙ МАГИСТРАЛЬНОГО ГАЗОПРОВОДА.....	29
ШЕВЧЕНКО Е.Л., ШЕВЧЕНКО А.Ю., БОГДАН А.Н. ОСОБЕННОСТИ ПОСТРОЕНИЯ ПРИКЛАДНЫХ ПРОГРАММНЫХ СИСТЕМ НА ОСНОВЕ ОНТОЛОГИЧЕСКИХ БАЗ ЗНАНИЙ.....	36
ЧЕПЕНКО Т.Е. МЕТОДЫ ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ НА ОСНОВЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ С ЭЛЕМЕНТАМИ ВРЕМЕННОЙ ЗАДЕРЖКИ.....	41
ОСТРОУХ А.В., НИКОЛАЕВ А.Б., ЕФИМЕНКО Д.Б., ЖАНКАЗИЕВ С.В. НАУЧНЫЙ ПОДХОД К РАЗРАБОТКЕ АВТОМАТИЗИРОВАННОЙ НАВИГАЦИОННОЙ СИСТЕМЫ ДИСПЕТЧЕРСКОГО КОНТРОЛЯ И УЧЕТА РАБОТЫ ТРАНСПОРТА НЕФТЕДОБЫВАЮЩИХ И НЕФТЕПЕРЕРАБАТЫВАЮЩИХ ПРЕДПРИЯТИЙ.....	48
БАРАННИК В.В., ДОДУХ А.Н. ТЕХНОЛОГИЯ СЖАТИЯ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ НА ОСНОВЕ ДВУХКОМПОНЕНТНОГО КОДИРОВАНИЯ.....	58
КИРИЧЕНКО Л.О. СРАВНИТЕЛЬНЫЙ МУЛЬТИФРАКТАЛЬНЫЙ АНАЛИЗ ВРЕМЕННЫХ РЯДОВ МЕТОДАМИ ДЕТРЕНДИРОВАННОГО ФЛУКТУАЦИОННОГО АНАЛИЗА И МАКСИМУМОВ МОДУЛЕЙ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ.....	66
БУЛАВИН Д. А., ХАРИТОНОВ И. А. АНАЛИЗ МЕТОДОВ РАСПОЗНАВАНИЯ И ПРЕОБРАЗОВАНИЯ АУДИОИНФОРМАЦИИ В НОТЫ.....	78
МАЛИЙ О.Ю., ФАРАФОНОВ О.Ю., ДЕМ'ЯНЕНКО О.А. МЕТОД ОПИСУ ИНЖЕНЕРНЫХ ЗАДАЧ ЗА ДОПОМОГОЮ УНІВЕРСАЛЬНОЇ МОВИ ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ.....	82
САРАНЧА С.Н. МЕТОДИКА ОПРЕДЕЛЕНИЯ ПАРАМЕТРОВ АППАРАТНОГО ГЕНЕРАТОРА СЛУЧАЙНЫХ ЧИСЕЛ РЕАЛИЗОВАННОГО В ПЛИС АРХИТЕКТУРЫ FPGA.....	89
ГВОЗДИНСКИЙ А.Н., БУШНОВ С.В. ИССЛЕДОВАНИЕ МЕТОДОВ ОПТИМИЗАЦИИ В СИСТЕМАХ ПРИНЯТИЯ РЕШЕНИЙ ПРИ ПЛАНИРОВАНИИ ТОВАРООБОРОТА.....	95
ГВОЗДИНСКИЙ А.Н., МАЛЫШКИН В.А., ТИЩЕНКО И.В. ПРИМЕНЕНИЕ МЕТОДОВ ЭВОЛЮЦИОННОЙ ОПТИМИЗАЦИИ В ЗАДАЧАХ УПРАВЛЕНИЯ ЗАПАСАМИ.....	102
ГЕРАСИМЕНКО К.Е., ХАХАНОВ В.И. ПОВЫШЕНИЕ ТЕСТОПРИГОДНОСТИ КРИТИЧЕСКИХ СИСТЕМ УПРАВЛЕНИЯ.....	107
ГРИШКО А.А. ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ В СТОХАСТИЧЕСКИХ СИСТЕМАХ УПРАВЛЕНИЯ ЗАПАСАМИ.....	117
РЕФЕРАТЫ.....	124
ПРАВИЛА ОФОРМЛЕНИЯ СТАТЕЙ ДЛЯ АВТОРОВ НАУЧНО-ТЕХНИЧЕСКОГО СБОРНИКА	130

ГЕНЕРИРОВАНИЕ БУЛЕАНА ДЛЯ СИНТЕЗА КВАНТОВОГО ПРОЦЕССОРА

Предлагаются кубитные (квантовые) структуры данных и вычислительных процессов для существенного повышения быстродействия при решении задач дискретной оптимизации. Анализируются работы по генерированию булеана, как существенного компонента при решении комбинаторных задач. Описываются аппаратно-ориентированные модели параллельного (за один цикл) вычисления булеана (множества всех подмножеств) на универсуме из n примитивов для решения задач покрытия, минимизации булевых функций, сжатия данных, синтеза и анализа цифровых систем за счет реализации процессорной структуры в форме диаграммы Хассе.

1. Актуальность

Задача генерирования булеана как множества всех подмножеств часто встречается на практике [1-20]. В связи с этим представляют интерес порождающие алгоритмы, к которым предъявляются следующие требования: 1. Полнота – порождение алгоритмом всего множества элементов. 2. Непротиворечивость – порождение алгоритмом только тех элементов, которые относятся к множеству. 3. Неповторимость – порождение алгоритмом элементов только один раз. Последнее свойство вызвано возможностью ухудшения временных характеристик и усложнением проверки условий остановки алгоритма. Задача порождения множества всех подмножеств может быть реализована с использованием кодов принадлежности [3] и алгоритмов генерации k -элементных подмножеств (сочетаний) [10, 12, 14-16]. 4. Практическая направленность. Квантовые вычисления в последние годы становятся интересными для анализа кибернетического пространства, создания новых Интернет-технологий, благодаря их некоторой альтернативности существующим моделям вычислительных процессов. Кроме того, рыночная привлекательность квантовых или кубитных моделей основывается на высоком параллелизме решения практически всех задач дискретной оптимизации, факторизации, минимизации булевых функций, эффективном сжатии, компактном представлении и телепортации данных, отказоустойчивом проектировании за счет существенного повышения аппаратных затрат. Но такая плата в настоящее время вполне допустима, поскольку существуют проблемы заполнения площадей силиконового кристалла, который содержит до 1 миллиарда вентиляей.

2. Линейный код и код Грея

Код принадлежности. Пусть дано множество $X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$. Код принадлежности для подмножества $\sigma \subseteq X$ определяется как набор $K = \langle k_1, k_2, \dots, k_i, \dots, k_n \rangle$, где

каждый компонент k_i есть $k_i = \begin{cases} 1, & x_i \in \sigma; \\ 0, & x_i \notin \sigma. \end{cases}$ Очевидно, что количество всех подмножеств

равно 2^n .

Линейный код. Алгоритм порождает подмножества следующим образом: на 0-м ярусе – код пустого множества; на 1-м ярусе – коды всех подмножеств, состоящих из одного элемента; на 2-м ярусе – коды всех двухэлементных подмножеств; далее – аналогично, на k -м ярусе – коды всех k -элементных подмножеств; на n -м ярусе – код n -элементного подмножества, т.е. самого множества как несобственного подмножества самого себя. Ярусы связаны дугами с номерами, которые соответствуют порождающим операциям $N = 2^0, 2^1, 2^2, \dots, 2^n$, выполняемым по следующим правилам.

1. Для N имеется $K_{lev+1} = K_{lev} \vee N_2$, где K_{lev} – порождающий код принадлежности к ярусу с номером lev (level – уровень), K_{lev+1} – порожденный код, N – натуральное число, N_2 – двоичный код числа N .

2. Если K_{lev} содержит 1, совпадающую с соответствующим разрядом N_2 , порождение прекращается.

Алгоритм с такими правилами последовательно продвигается по ярусам, сначала порождая подмножества из одного элемента, затем – из двух, трех и далее аналогично (рис. 1).

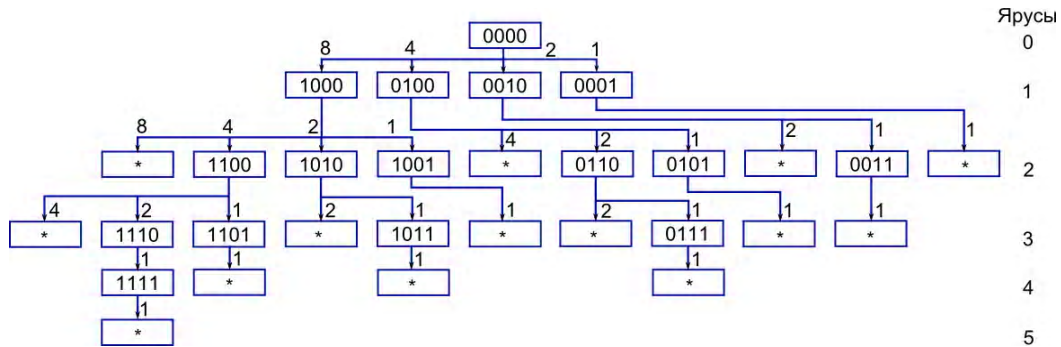


Рис. 1. Алгоритм порождения булеана

Коды Грея. Как и для линейного кода, каждый ярус содержит коды принадлежности булеану. Количество элементов каждого кода определяется номером яруса. В частности, ярус 3 содержит $2^3=8$ кодов принадлежности всех подмножеств, построенных из трех элементов (рис. 2).

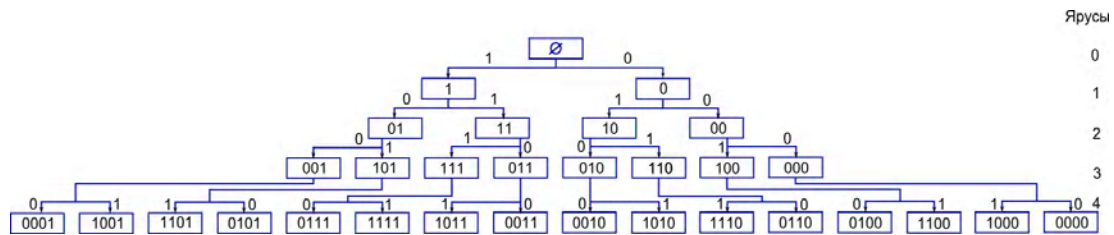


Рис. 2. Генерирования кода Грея

Код принадлежности на ярусе $lev+1$ получается приписыванием слева к коду принадлежности lev числа λ , которое попеременно принимает значения 0 или 1. Полученная на каждом ярусе последовательность кодов принадлежности называется n -элементным кодом Грея, где n – номер яруса. Код Грея обладает двумя следующими свойствами:

1. Два соседних набора в коде Грея отличаются на 1 в единственной позиции, наборы находятся на расстоянии по Хэммингу, равном единице. *Расстояние по Хэммингу* в общем случае определяется количеством позиций, по которым отличаются двоичные наборы.

2. Отдельные бинарные наборы кода являются вершинами n -мерного двоичного куба, а последовательность двоичных наборов есть гамильтонов цикл, по которому можно обойти вершины, включая каждую только один раз (рис. 3).

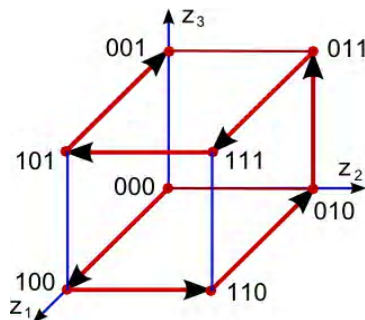


Рис. 3. Генерирование соседнего кода

3. Генерация k -элементных подмножеств (сочетаний)

В комбинаторике такие подмножества называют сочетаниями из n элементов по k и обозначают C_n^k . Их количество вычисляется по формуле:

$$C_n^k = \frac{n!}{k!(n-k)!}, 0 \leq k \leq n. \quad (1)$$

Однако при программировании гораздо удобнее использовать следующие рекуррентные соотношения:

$$\begin{aligned} C_n^k &= C_{n-1}^{k-1} + C_{n-1}^k, 0 < k < n; \\ C_n^k &= \frac{n}{k} C_{n-1}^{k-1}, 0 < k \leq n. \end{aligned} \quad (2)$$

Поскольку в формуле (1) числитель и знаменатель растут очень быстро, не всегда возможно точно вычислить значение C_n^k , даже когда последнее не превосходит максимально представимое целое число. При фиксированном значении n число сочетаний достигает максимального значения при $k = n/2$. Для четного n максимум $k = n/2$ (индекс k соответствует номеру среднего члена разложения бинома Ньютона), а для нечетного – максимум достигается на двух соседних значениях $k = [n/2]$ и $k = [n/2] + 1$. Поэтому полезной оказывается следующая оценка для четных n (очевидно, что при нечетных n отличия будут минимальными), основанная на формуле Стирлинга [4]:

$$C_n^{n/2} = \frac{2^n}{\sqrt{\pi n/2}} (1 + O(1/n)). \quad (3)$$

Если допустить, что за время, отведенное для решения задачи, можно перебрать около 10^6 вариантов, то из формулы (3) следует, что генерацию всех сочетаний из n элементов для любого фиксированного k можно проводить для $n \leq 24$. Далее предлагается рассмотреть алгоритмические вопросы, связанные с порождением сочетаний. Для множества $M = \{1, 2, \dots, n\}$ порождение всех сочетаний (подмножеств) описывается несколькими алгоритмами [1, 2, 6, 8, 11, 13, 18].

3.1. Алгоритм «Сочетания»

1) Дано: множество M . Найти: последовательность подмножеств множества M . 2) Начало: пустое множество \emptyset . 3) Пусть порождено множество S . Тогда следующее множество S' получается включением в S наименьшего не принадлежащего ему элемента и удалением из него всех элементов, меньших включенного. Если таких элементов нет, выполняется завершение алгоритма. Справедливость алгоритма доказывается индукцией по n : сначала алгоритм порождает все подмножества множества $M' = \{1, 2, \dots, n-1\}$, затем повторяет эту последовательность, добавляя к каждому подмножеству элемент n . Последним членом в последовательности будет само множество M , и алгоритм автоматически остановится.

3.2. Алгоритм «Сочетания добавлением/исключением одного элемента»

Наряду с рассмотренным выше существует алгоритм, который порождает все подмножества множества M так, что каждое следующее отличается от предыдущего не более, чем на один элемент. Данный алгоритм определяется индуктивно.

1) Если $n = 1$, то результатом алгоритма есть последовательность $\emptyset, \{1\}$.

2) Пусть для $n-1$ результатом алгоритма является последовательность L_{n-1} . Пусть $\overline{L_{n-1}}$ – записанная в обратном порядке последовательность L_{n-1} . Тогда для n результатом алгоритма будет последовательность $L_n = L_{n-1}, \overline{L_{n-1}} \vee \{n\}$, где $\overline{L_{n-1}} \vee \{n\}$ означает, что к каждому члену последовательности $\overline{L_{n-1}}$ добавлен элемент n .

Пример 1. $L_2 = \{\emptyset, \{1\}, \{1,2\}, \{2\}\}$; $L_3 = \{\emptyset, \{1\}, \{1,2\}, \{2\}, \{2,3\}, \{1,2,3\}, \{1,3\}, \{3\}\}$. Справедливость алгоритма подтверждается индукцией по n , если учесть, что L_{n-1} заканчивается множеством $\{n-1\}$. Первая половина последовательности L_n совпадает с L_{n-1} и по индукции удовлетворяет необходимым требованиям. Следующий элемент после L_{n-1} есть $\{n-1, n\}$, и затем проходит последовательность L_{n-1} в обратном порядке с добавлением элемента $\{n\}$, и требования также выполняются. Поскольку первый элемент в L_{n-1} есть \emptyset , то последний элемент в L_n есть $\{n\}$. Если длина L_{n-1} есть 2^{n-1} , то длина L_n есть $2 \cdot 2^{n-1}$.

3.3. Алгоритм порождения сочетаний в лексикографическом порядке

Как правило, генерация всех k -элементных подмножеств выполняется в лексикографическом порядке, тем более что это не приводит к усложнению алгоритма или увеличению его вычислительной трудоемкости. Порядок подмножеств называется лексикографическим, если для любых двух подмножеств раньше должно быть сгенерировано то из них, из индексов элементов которого можно составить меньшее k -значное число в n -ричной системе счисления (в десятичной, для $n < 10$). Например, при $n=6$ и $k=3$ сочетание из третьего, первого и пятого элементов должно быть сгенерировано раньше, чем из второго, третьего и четвертого, поскольку $135 < 234$.

Рассмотрим рекурсивный алгоритм решения данной задачи. Идея ее сведения к задаче меньшей размерности заключается в следующем. Первым элементом подмножества может быть любой из них, начиная с первого и заканчивая $(n-k+1)$ -м элементом. После того, как индекс первого элемента подмножества зафиксирован, остается выбрать $k-1$ элемент из множества с индексами большими, чем у первого. Далее – аналогично. Когда выбран последний элемент, то достигнут конечный уровень рекурсии, и выбранное подмножество можно использовать по назначению.

Ниже приведен алгоритм, который порождает все r -сочетания n -элементного множества $M = \{1, 2, \dots, n\}$ в лексикографическом порядке. Каждое r -сочетание представляется числами $a_1 < a_2 < \dots < a_r$. 1) Начало алгоритма: $a_1 = 1, a_2 = 2, \dots, a_r = r$. 2) Пусть порождено r -сочетание a_1, a_2, \dots, a_r . Тогда определяется наибольшее a_i такое, что $a_i + 1$ не входит в данное сочетание. Если таких a_i нет (это происходит только при $a_1 = n - r + 1, a_r = n$), то стоп. 3) Если такое a_i есть, то образуем новое сочетание a'_1, a'_2, \dots, a'_r , полагая

$$a'_1 = a_1, \dots, a'_{i-1} = a_{i-1}, \dots, a'_i = a_i + 1, a'_{i+1} = a_i + 2, \dots, a'_r = a_i + r - 1.$$

Пример 2. Алгоритм для $n = 4, r = 2$:

Шаги	Сочетания	a_i
1	1,2	2
2	1,3	3
3	1,4	1
4	2,3	3
5	2,4	2
6	3,4	Стоп

В предлагаемом ниже листинге 1 массив a содержит значения элементов исходного множества и может быть заполнен произвольным образом. В массиве p формируется очередное сочетание из k элементов.

Листинг 1:

```
const nmax = 24;
type list = array[1..nmax] of integer;
var k, i, j, n, q: integer;
a, p: list;
```

```

procedure print(k: integer);
var i: integer;
begin
for j := 1 to k do
write(p[j]:4);
writeln
end; {print}
procedure cnk(n, k: integer);
procedure gen(m, L: integer);
var i: integer;
begin
if m = 0 then print(k)
else
for i := L to n - m + 1 do
begin
p[k - m + 1] := a[i];
gen(m - 1, i + 1)
end
end; {gen}
begin {cnk}
gen(k, 1)
end; {cnk}
begin {main}
readln(n,k);
for i := 1 to n do
a[i] := i;
{заполнить массив можно и по-другому}
cnk(n,k)
end

```

Генерация сочетаний выполняется в рекурсивной подпрограмме gen со следующими параметрами: m – количество элементов, которые осталось выбрать из множества; L – элемент исходного множества, начиная с которого следует выбирать эти m элементов. Вложенная структура описания процедур cnk и gen позволяет не передавать при рекурсивных вызовах значения n и k, а из основной программы обращаться к процедуре cnk с параметрами, соответствующими постановке задачи.

4. Генерация всех подмножеств данного множества (булеана)

4.1. Перебор по возрастанию/убыванию числа элементов

При отыскании минимального/максимального подмножества, состоящего как можно из меньшего/большого числа элементов, эффективнее всего организовать перебор так, чтобы сначала проверялись все подмножества из одного элемента, затем – из двух, трех и далее элементов (для максимального подмножества — в обратном порядке). В этом случае первое подмножество, удовлетворяющее условию задачи, будет искомым, и дальнейший перебор следует прекратить. Для реализации такого перебора можно воспользоваться описанной выше процедурой cnk и ввести в нее еще один параметр – логическую переменную flag, которая будет идентифицировать соответствие текущего сочетания элементов условию задачи. При получении очередного сочетания необходимо обратиться к процедуре его проверки check, которая определяет значение флага. Тогда начало процедуры gen следует переписать следующим образом (листинг 2).

Листинг 2:

```

procedure gen(m, L: integer);
var i: integer;
begin
if flag then exit;
if m = 0 then

```

```

begin check(p,k,flag);
end
else ...

```

Далее процедура совпадает с предыдущей версией (см. листинг 1). В основной программе единственное обращение к данной процедуре следует заменить следующим фрагментом (листинг 3).

```

Листинг 3:
k:=0;
flag:=false;
repeat k:=k+1;
cnk(n,1,flag) until flag or (k=n);
if flag then print(k)
else writeln ('no solution');

```

Видно, что в основной программе запрос значения переменной k теперь не производится.

4.2. Перебор по двоичной системе (бинарному или двоичному коду)

Рассматривается альтернативный подход к перебору всех подмножеств заданного множества. Любое подмножество можно охарактеризовать, указав относительно каждого элемента исходного множества его принадлежность данному подмножеству и поставив в соответствие каждому элементу множества 0 или 1. Каждому подмножеству соответствует n-значное число в двоичной системе счисления. Поскольку числа могут начинаться с произвольного количества нулей, которые не являются значащими цифрами, в соответствии ставятся n или менее значные числа. Отсюда следует, что полный перебор всех подмножеств исходного множества соответствует перебору всех чисел в двоичной системе счисления от $\underbrace{0\dots0}_n 1$ до $\underbrace{1\dots1}_n$.

Теперь можно подсчитать количество различных подмножеств данного множества. Оно равно $2^n - 1$ или 2^n с учетом пустого множества.

Таким образом, сопоставляя два способа перебора всех подмножеств данного множества, можно получить следующую формулу:

$$C_n^0 + C_n^1 + \dots + C_n^n = 2^n, \quad (4)$$

которая известна как сумма биномиальных коэффициентов [7].

Итак, в рамках сделанной выше оценки на количество допустимых вариантов перебора можно рассмотреть все подмножества исходного множества только при $n \leq 20$.

Применение программ, соответствующих второму способу перебора, целесообразно, когда: 1. Необходимо перебрать все подмножества данного множества, например, требуется найти все решения, удовлетворяющие заданному условию. 2. По условию задачи не имеет значения количество элементов, которое должно входить в искомое подмножество. На примере такой задачи далее приводится программа генерации всех подмножеств исходного множества в лексикографическом порядке [5].

Пример 3. Даны целочисленный массив $a[1, \dots, N]$, $N \leq 20$ и число M. Найти подмножество элементов массива $a[i_1], a[i_2], \dots, a[i_k]$ такое, что $1 \leq i_1 < i_2 < \dots < i_k \leq N$ и $a[i_1] + a[i_2] + \dots + a[i_k] = M$.

В качестве решения приводится процедура генерации всех подмножеств, которые можно составить из элементов массива, и функция проверки конкретного подмножества на соответствие условию задачи (листинг 4).

```

Листинг 4:
function check(j: longint): boolean;
var k: integer; s: longint;
begin
s := 0;
for k := 1 to n do

```

```

{данное условие означает, что в k-й справа позиции числа j во 2-й системе стоит 1}
if ((j shr (k — 1)) and 1) = 1
then s := s + a[k];
if s = m then
begin
for k := 1 to n do
if ((j shr (k — 1)) and 1) = 1
then write(a[k]:4);
writeln
end
end;
procedure subsets(n: integer);
var q,j: longint;
begin
{помещаем в q число 2^n}
q := 1 shl n;
{цикл по всем подмножествам}
for j := 1 to q — 1 do
if check(j) then exit
end

```

Следует заметить, что если все элементы в массиве положительные, то решение приведенной выше задачи можно сделать более эффективным путем изменения порядка рассмотрения подмножеств, а именно: если сумма элементов какого-либо подмножества уже больше M , то рассматривать подмножества, включающие его в себя, уже не имеет смысла. Пересчет сумм может быть оптимизирован, если каждое следующее сгенерированное подмножество будет отличаться от предыдущего не более чем на один элемент [4]. Приведенная программа достаточно проста, но обладает одним недостатком: с ее помощью нельзя перебрать все подмножества множеств из более, чем 30 элементов, что обусловлено максимальным числом битов (32), отводимых на представление целых чисел. На самом деле перебор всех подмножеств множеств большей размерности вряд ли возможен за время, отведенное для решения той или иной задачи.

Пример 4. Для трехэлементного множества $M = \{1, 2, 3\}$ все подмножества следующие: $\{\}, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$. Первым в списке идет пустое множество, которое является подмножеством любого множества, последним – исходное множество как несобственное подмножество самого себя. Рассматриваются числа в диапазоне от 0 до $2^n - 1$ в двоичной системе счисления с разрядностью n , которые можно увеличить при необходимости незначащими нулями: 0 (000); 1 (001); 2 (010); 3 (011); 4 (100); 5 (101); 6 (110); 7 (111).

Каждый из этих двоичных кодов можно использовать как маску подмножества, а именно: единица в i -м бите характеризует наличие i -го элемента множества в подмножестве, ноль – его отсутствие. Например: $101 = \{1, 3\}$. Таким образом, для генерации подмножеств необходимо организовать цикл от 0 до $2^n - 1$, при этом на каждой итерации представить значение счетчика в виде бинарного кода и на его основе составить подмножество. Реализация на языке Си для четырехэлементного множества $M = \{3, 6, 7, 9\}$ представлена ниже.

```

Листинг 5:
#include <stdio.h>
#include <stdlib.h>
//—для возведения в степень
#include <math.h>
int main()
{
int M[4] = {3, 6, 7, 9}; //—множество
int w = 4; //—кол-во элементов множества
int i, j, n;

```

```

n = pow(2, w);
for ( i = 0; i < n; i++ ) //—перебор битовых масок
{
printf("{");
for ( j = 0; j < w; j++ ) //—перебор битов в маске
if ( i & (1 << j) ) //—если j-й бит установлен
printf("%d ", M[j]); //—то выводим j-й элемент множества
printf("{\n}");
}
return 0;
}

```

Результат представлен следующим набором подмножеств: {}, {3}, {6}, {3, 6}, {7}, {3, 7}, {6, 7}, {3, 6, 7}, {9}, {3, 9}, {6, 9}, {3, 6, 9}, {7, 9}, {3, 7, 9}, {6, 7, 9}, {3, 6, 7, 9}. Порядок в данном случае не имеет значения. Программная реализация вывода подмножеств достаточно проста. В процедуре `if (i & (1 << j))` оператор `<<` есть логический сдвиг влево. Чтобы узнать значение j-го бита числа i (биты отсчитываются слева направо, нумерация начинается с нуля), следует сначала сдвинуть единицу на j. Допустим, i=18 (10010), j=2. После логического сдвига влево единицы (00001) на 2 получается число 00100. Затем число i логически умножается на полученную после сдвига маску. Маска представляет совокупность нулей с установленным в единицу j-м битом благодаря сдвигу. При умножении, если j-й бит в числе i равен единице, получается некоторое число, равное маске; если j-й бит равен нулю, то результат умножения будет нулем: 10010, 00100. Побитовое умножение двух чисел – логическая операция `and` – дает число 00000. Это говорит о том, что 2-й бит числа 18 (согласно порядку нумерации битов) выставлен в ноль. Данный алгоритм позволяет довольно просто сгенерировать случайное подмножество множества. Для этого необходимо взять случайное число в диапазоне от 0 до $2^n - 1$, использовать его как битовую маску и сгенерировать на ее основе подмножество.

4.3. Процедура генерирования булеана

Задано множество из четырех компонентов: A, B, C, D. Необходимо выбрать подмножество, состоящее из нескольких примитивов, обладающее некоторым свойством. Способ решения задачи заключается в генерировании всех возможных подмножеств данного множества и для каждого из сгенерированных подмножеств выполняется проверка, удовлетворяет ли оно заданному свойству. Альтернативный вариант решения задачи – вычислить все подмножества данного множества, обладающие заданным свойством. Например, для множества из четырех символов A, B, C, D булеан включает следующие множества: пустое множество; одноэлементные множества: {A}, {B}, {C}, {D}; двухэлементные множества: {A, B}, {A, C}, {A, D}, {B, C}, {B, D}, {C, D}; трехэлементные множества: {A, B, C}, {A, B, D}, {A, C, D}, {B, C, D}; четырехэлементное множество: {A, B, C, D}.

В случае, когда порядок генерации подмножеств не имеет значения, алгоритм генерации множества всех подмножеств выглядит следующим образом [20]. Иницируется вектор B, состоящий из четырех разрядов, каждый из которых может принимать значение 0 или 1. Единица указывает на включение соответствующего по номеру компонента исходного множества в искомое. Значение 0 указывает на то, что элемент отсутствует. Пусть имеется последовательность двоичных чисел от 0 до 15 и соответствующие им подмножества:

```

{0000} – пустое множество;
{0001} – A, {0010} – B, {0100} – C, {1000} – D;
{0011} – AB, {0101} – AC, {0110} – BC, {1001} – AD, {1010} – BD, {1100} – CD;
{0111} – ABC, {1011} – ABD, {1101} – ACD, {1110} – BCD;
{1111} – ABCD.

```

Таким образом, имеется 16 различных подмножеств универсума из 4 элементов. В общем случае множество всех подмножеств множества из N элементов содержит 2^N элементов. Алгоритм, обеспечивающий генерацию множества всех подмножеств из N элементов, может быть неформально описан следующим образом: формирование вектора,

состоящего из N нулей и соответствующего пустому множеству. Для получения следующего подмножества из текущего обрабатывается текущий вектор следующим образом: справа от первого элемента массива к последнему ищется первое число, равное 0. Если такое число не найдено, то текущее подмножество является решением – множеством, состоящим из всех элементов. Если же элемент, равный 0, найден, то он заменяется на 1, а все числа справа от него, если такие имеются, заменяются на нули. Данный алгоритм может быть записан следующим образом:

```

Ввод (N)
Обнуление массива В из N+1 элемента
Вывод (пустое множество)
Пока В[N+1]=0
  i:=1
  Пока В[i]=1 делать В[i]=0, i=i+1
  В[i]=1
Вывод (множество, определяемое массивом В)
Соответствующая программа имеет следующий вид (листинг 6).

```

Листинг 6:

```

Программа генерации множества всех подмножеств
const
alphabet : string[26] = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
var
b : array [1..100] of byte;
N,i : byte;
begin
readln(N);
for i:=1 to N+1 do b[i]:=0;
writeln ('Пустое множество');
while (b[N+1]=0) do
begin
i:=1;
while B[i]=1 do
begin B[i]:=0; inc(i); end;
B[i]:=1;
for i:=1 to n do
if b[i]=1 then write(alphabet[i]);
writeln;
end;
end

```

5. Квантовый процессор оптимального покрытия

Цель создания кубит-процессора – существенное уменьшение времени при решении задач оптимизации путем параллельного вычисления векторных логических операций над множеством всех подмножеств от примитивных компонентов за счет увеличения памяти для хранения промежуточных данных.

Задачи: 1) Определение структур данных для взятия булеана при решении задачи покрытия столбцов матрицы $M = \begin{bmatrix} M_{ij} \end{bmatrix}, i = \overline{1, m}; j = \overline{1, n}$ единичными значениями строк. В частности, при $m = n = 8$, необходимо выполнить параллельно-логическую операцию над 256 вариантами всех возможных сочетаний векторов (строк матрицы), составляющих булеан. 2) Система команд процессора должна включать следующие операции (and, or, xor) над векторами (словами), размерности m. 3) Разработка архитектуры кубит-процессора для параллельного вычисления $2^n - 1$ вариантов сочетаний, направленных на оптимальное решение NP-полной задачи покрытия. 4) Реализация прототипа кубит-процессора на базе программируемой логики PLD и верификация (валидация) аппаратного решения на примерах минимизации булевых функций. 5) Приведение других практических задач дискретной оптимизации к форме задачи покрытия для последующего решения на кубит-процессоре.

В качестве примера предлагается решить задачи поиска оптимального единичного покрытия всех столбцов минимальным числом строк матрицы M , представленной ниже:

M	1	2	3	4	5	6	7	8
a	1	1	.	.
b	.	.	1	.	.	.	1	.
c	1	.	.	.	1	.	1	.
d	.	1	.	1	.	.	.	1
e	.	1	.	.	1	.	.	.
f	1	.	1	.	.	1	1	.
g	.	1	.	1	.	.	.	1
h	.	.	1	.	1	.	.	.

Для этого необходимо сделать перебор всех 255 сочетаний: из восьми по одной строке, по двум, трем, четырем, пяти, шести, семи и восьми. Минимальное количество примитивов (строк), формирующее покрытие, есть оптимальное решение. Таких решений может быть несколько. Диаграмма Хассе есть компромиссное предложение относительно времени и памяти, или такая стратегия решения задачи покрытия, когда ранее полученный результат впоследствии используется для создания более сложной суперпозиции. Поэтому для каждой таблицы покрытия, содержащей n примитивов (строк), необходимо генерировать собственную мультипроцессорную структуру в форме диаграммы Хассе, которая далее должна быть использована для почти параллельного решения NP-полной задачи. Например, для четырех строк таблицы покрытия диаграмма Хассе – структура мультипроцессора – будет иметь вид, представленный на рис. 4.

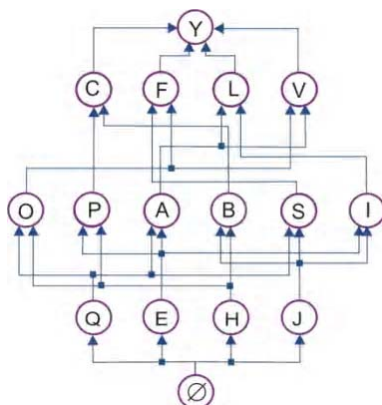


Рис. 4. Квантовая структура вычислительных процессов

Оптимальные решения задачи покрытия для матрицы M , которая генерирует 255 вариантов возможных сочетаний, представлены строками в форме ДНФ: $C = fgh \vee efg \vee cdf$.

Автомат управления вычислительным процессом для квантовой структуры путем восходящего анализа вершин графа основан на последовательном выполнении следующих шагов:

1. Занесение информации о примитивах в регистры (матрицы) $L_i^1 = P_i$ первого уровня с последующим анализом качества покрытия каждым примитивом в двоичном формате (1 – есть покрытие, 0 – нет покрытия). Если один из примитивов организует покрытие $\bigwedge_{j=1}^m L_{ij}^1 = 1$, процесс анализа структуры Хассе заканчивается. Иначе – выполнение перехода ($r = r + 1$) на следующий уровень графа:

$$L_i^1 = P_i \rightarrow \bigwedge_{j=1}^m L_{ij}^1 = \begin{cases} 0 \rightarrow n = n + 1; \\ 1 \rightarrow \text{end.} \end{cases}$$

2. Инициирование команды обработки очередного (второго) уровня. Последовательное выполнение векторных (матричных) операций ($\text{or } L_i^r = L_{ij}^{r-1} \bigwedge_{j=1}^m L_{ij}^{r-1}$, and $\bigwedge_{j=1}^m L_{ij}^r = 1$) в целях анализа покрывающей способности сочетаний примитивных элементов r -уровня. Здесь

$t = \overline{1, m}, i = \overline{1, m}, r = \overline{1, n}$, n – число ярусов или количество строк в таблице покрытия, m – число столбцов в ней. Если существует сочетание на рассматриваемом уровне, создающее полное покрытие, которое формирует оценку, равную 1, то обработка всех последующих уровней процессора не выполняется. В противном случае выполняется переход для анализа следующего уровня процессорной структуры:

$$L_i^r = L_{ij}^{r-1} \bigwedge_{j=1}^m L_{ij}^{r-1} \rightarrow \bigwedge_{j=1}^m L_{ij}^r = \begin{cases} 0 \rightarrow r = r + 1; \\ 1 \rightarrow \text{end}. \end{cases}$$

Для поиска оптимального покрытия всегда достаточно двух элементов нижнего уровня, что означает – все операционные вершины имеют два регистровых (матричных входа), что существенно уменьшает стоимость аппаратных затрат. Количество временных тактов для обработки структуры процессора в худшем случае равно n . Можно создать алгоритм поиска оптимального покрытия путем нисходящего анализа вершин графа. В этом случае при нахождении полного покрытия в одном из ярусов необходим еще один спуск по структуре, чтобы убедиться в отсутствии в нижнем соседнем ярусе полного покрытия. При положительном ответе на данный вопрос полученное решение является оптимальным. Иначе – необходимо выполнять спуск до такого уровня, когда, более нижний, соседний ярус не будет содержать полного покрытия.

Вершины процессорной структуры могут иметь более одной бинарной (унарной) регистровой логической операции. Тогда необходимо создавать простейший дешифратор команд для активизации, например, операций: and, or, xor, not.

Таким образом, достоинства кубитного Хассе процессора (Qubit Hasse Processor – QHP) заключаются в возможности использовать не более, чем двухвходовые схемы векторных логических операций (and, or, xor), а значит, в существенном уменьшении стоимости по Квайну реализации процессорных элементов (вершин) и памяти за счет применения последовательных вычислений и незначительного увеличения времени обработки всех вершин графа Хассе. Для каждой вершины используется критерий качества покрытия – наличие всех единиц в координатах вектора-результата. Если критерий качества выполняется, то все остальные вычисления можно не производить, поскольку диаграмма Хассе есть строго иерархическая структура по числу сочетаний в каждом ярусе. Это означает, что самое лучшее решение находится на более низком уровне иерархии. Варианты одного уровня равнозначны по реализации (стоимости), поэтому полученное

первое качественное покрытие ($Q = \sum_{i=1}^n q_i = n$) есть лучшее решение, предполагающее остановку всех последующих вычислений по стратегии диаграммы Хассе. С учетом последовательно-параллельной стратегии анализа вершин графа, время (цикл) обработки всех примитивов QH-процессора определяется числом уровней иерархии, функционально зависящим от количества битов в кубитной переменной: $T = \log_2 2^n \times t = t \times n$. При этом длина m строки таблицы покрытия не влияет на оценку быстродействия. Анализ вершины включает две команды: логическую (and, or, xor) и операцию вычисления критерия качества покрытия в форме скаляра путем применения функции and ко всем разрядам вектора-результата:

$$m_{ir,j} = M_{i,j} \vee M_{r,j}, (j = \overline{1, n}; \{i \neq r\} = \overline{1, m});$$

$$m_{ir}^s = \bigwedge m_{ir,j} = \bigwedge (M_{i,j} \vee M_{r,j}).$$

Аппаратные затраты на реализацию QH-процессора зависят от суммарного числа вершин графа Хассе и от количества битов (разрядов) в строке таблицы покрытия:

$$H = 2^n \times k \times m,$$

где k – коэффициент аппаратной реализации (сложности) одного разряда бинарной векторной логической операции и последующей команды вычисления критерия качества покрытия.

Таким образом, высокое быстродействие решения задачи покрытия достигается существенным повышением аппаратных затрат (в $2^n \times k \times m / k \times m \times n = 2^n / n$ раз по сравне-

нию с последовательной обработкой графовых вершин), которое обеспечивает компромиссный вариант между полностью параллельной структурой вычислительных процессов (здесь затраты аппаратуры определяются числом примитивов в каждой вершине $H = k \times m \times n \times 2^n$, а увеличение аппаратуры составляет 2^n раз) и последовательными вычислениями однопроцессорного компьютера (здесь быстродействие обработки графа Хассе равно $T = t \times 2^n$, а аппаратные затраты равны $H^* = k \times m \times n$). Уменьшение аппаратуры по сравнению с параллельным вариантом обработки графа составляет $Q^H = k \times m \times n \times 2^n / k \times m \times 2^n = n$. Как следствие существенной аппаратной избыточности, уменьшение времени анализа вершин графа по сравнению с последовательной обработкой структуры имеет следующую оценку:

$$Q^T = \frac{t \times 2^n}{t \times n} = \frac{2^n}{n}.$$

6. Практическая реализация кубит-процессора оптимального покрытия

Модель квантового устройства разработана на языке Verilog. Элементарная ячейка процессора состоит из двух регистровых вентилях. Регистровый элемент выполняет логическую операцию над двумя векторами, формируя вектор результата. Регистровый вентиль and выполняет свертку всех битов вектора по операции and, формируя однобитовый элемент, идентифицирующий единичным значением оптимальное решение задачи покрытия. Представлено формирование значений для вершин диаграммы Хассе шести уровней, где каждый элемент в схеме имеет на выходе примитив анализа качества покрытия в виде функции and. Реализация вычислительного устройства выполнена на основе кристалла FPGA фирмы Xilinx xc3s1600e-4-fg484, где основные параметры имеют следующий вид:

Листинг 7:

Map-report

Logic Utilization:

Number of Slice Flip Flops: 2,286 out of 29,504 7%

Number of 4 input LUTs: 2,715 out of 29,504 9%

Logic Distribution:

Number of occupied Slices: 1,514 out of 14,752 10%

Number of Slices containing only related logic: 1,514 out of 1,514 100%

Number of Slices containing unrelated logic: 0 out of 1,514 0%

*See NOTES below for an explanation of the effects of unrelated logic.

Total Number of 4 input LUTs: 2,715 out of 29,504 9%

Number of bonded IOBs: 321 out of 376 85%

Number of BUFGMUXs: 1 out of 24 4%

Timing parameters of project:

Tclk_to_clk = 4.672 ns

Tclk_to_pad_max = 11.552 ns

Period = max{ Tclk_to_clk, Tclk_to_pad_max };

Period = 11.552 ns

Fclk = 86,5 МГц

Заключение

Представлен обзор и классификация моделей, структур данных и методов генерирования булеана для решения практических задач дискретной оптимизации.

Научная новизна – предложена модель данных и структура аппаратной реализации квантового компьютера, которая характеризуется использованием диаграммы Хассе, что дает возможность существенно ($\times 100$) повысить быстродействие решения комбинаторных задач дискретной оптимизации.

Практическая значимость – существенное повышение быстродействия при решении задач покрытия и других задач дискретной оптимизации за счет увеличения аппаратных затрат для параллельного выполнения векторных логических операций на Хассе-структуре квантового вычислительного устройства.

Список литературы: 1. *Burris, Stanley N., Sankappanavar H.P.* A Course in Universal Algebra. — Springer-Verlag, 1981. 332 p. 2. *Peter Jipsen, Henry Rose.* Varieties of Lattices – Lecture Notes in Mathematics 1533, Springer Verlag, 1992. 162 p. 3. *Носов В.А.* Комбинаторика и теория графов. М.: Изд-во МГИЭМ, 1999. 112 с. 4. *Стив Тейксейра, Ксавье Пачеко.* Borland Delphi 6. Руководство разработчика (Steve Teixeira, Xavier Pacheco. Delphi 6 Developer's Guide First Edition). М.: Издательский дом "Вильямс", 2002. 1120 с. 5. *Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн.* Алгоритмы: построение и анализ. М.: Издательский дом "Вильямс", 2012. 1296 с. 6. *Комбинаторный анализ.* Задачи и упражнения / Под ред. Рыбникова К.А. М.: Наука, 1982. 184 с. 7. *Гаврилов Г.П., Сапоженко А.А.* Сборник задач по дискретной математике. М.: Наука, 1977. 368 с. 8. *Липский В.* Комбинаторика для программистов: Пер. с польск. М.: Мир, 1988. 213 с. 9. *Романовский И.В.* Дискретный анализ. СПб: Невский Диалект; БХВ Петербург, 2003. 320 с. 10. *Ахо А., Хопкрофт Дж., Ульман Дж.* Структуры данных и алгоритмы. М.: Издательский дом «Вильямс», 2000. 384 с. 11. *Вирт Н.* Алгоритмы и структуры данных / Пер. с англ. М.: Мир, 1989. 360 с. 12. *Окулов С.М., Ашихмина Т.В., Бушмелева Н.А.* и др. Задачи по программированию / Под ред. С.М. Окулова. М.: БИНОМ. Лаборатория знаний, 2006. 820 с. 13. *Кормен Т., Лейзерсон Ч., Ривест Р.* Алгоритмы: построение и анализ. М.: МЦНМО, 1999. 960 с. 14. *Кнут Д.* Искусство программирования. Сортировка и поиск. Т. 3. М.: «Вильямс», 2007. 824 с. 15. *Кнут Д.* Искусство программирования. Генерация всех кортежей и перестановок. Т. 4, вып. 2. М.: «Вильямс», 2008. 160 с. 16. *Кнут Д.* Искусство программирования, том 4, выпуск 3. Генерация всех сочетаний и разбиений. М.: «Вильямс», 2007. 208 с. 17. *Грэхем Р., Кнут Д., Паташник О.* Конкретная математика. Основание информатики / Пер. с англ. М.: Мир, 1998. 703 с. 18. *Липский В.* Комбинаторика для программистов. М.: Мир, 1988. 77 с. 19. *Рейнгольд Э., Нивергельт Ю., Део Н.* Комбинаторные алгоритмы: теория и практика. М.: Мир, 1980. 476 с. 20. *Долинский М.С.* Алгоритмизация и программирование на Turbo Pascal: от простых до олимпиадных задач: Учебное пособие. Изд-во: Питер, 2006. 366 с. 21. *Beth T.* Quantum computing: an introduction // Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS. 2000. Geneva. Vol. 1. P. 735 – 736. 22. *Jonker P., Jie Han.* On quantum and classical computing with arrays of superconducting persistent current qubits // Proceedings of Fifth IEEE International Workshop on Computer Architectures for Machine Perception. 2000. P. 69 – 78. 23. *Keyes R.W.* Challenges for quantum computing with solid-state devices // Computer. Jan. 2005. Vol. 38, Issue 1. P. 65 – 69. 24. *Инфраструктура мозгоподобных вычислительных процессов / М.Ф. Бондаренко, О.А. Гузь, В.И. Хаханов, Ю.П. Шабанов-Кушнаренко.* Харьков: Новое Слово. 2010. 160 с. 25. *Mark Gregory Whitney.* Practical Fault Tolerance for Quantum Circuits. PhD Dissertation in Computer Science. Berkeley: University of California. 2009. 206 p. 26. *Хаханов В. И., Литвинова Е. И., Чумаченко С. В., Гузь О. А.* Логический ассоциативный вычислитель // Электронное моделирование. 2011. № 1. С. 73-90. 27. *Hahanov V., Wajeb Gharibi, Litvinova E., Chumachenko S.* Information analysis infrastructure for diagnosis // Information. An international interdisciplinary journal. 2011. Japan. Vol. 14, No 7. P. 2419-2433. 28. *Хаханов В.И.* Проектирование и тестирование цифровых систем на кристаллах / В.И. Хаханов, Е.И. Литвинова, О.А. Гузь. Харьков: ХНУРЭ, 2009. 484 с.

Поступила в редколлегию 14.12.2011

Чумаченко Светлана Викторовна, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: математическое моделирование, теория рядов, методы дискретной оптимизации. Увлечения: путешествия, любительское фото. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: ri@kture.kharkov.ua.

Хаханов Владимир Иванович, декан факультета КИУ ХНУРЭ, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanov@kture.kharkov.ua.

Мурад Али Аббас, аспирант кафедры АПВТ ХНУРЭ. Научные интересы: компьютерные системы и сети. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.

Горобец Олег Александрович, магистрант Центра последипломного образования ХНУРЭ. Научные интересы: аналитика программного обеспечения, облачные вычисления и социальные сети. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, e-mail: gorobetsu@gmail.com.

СТАТИСТИЧЕСКИЙ ПОДХОД К ВЫБОРУ ШАГОВ ПРЕОБРАЗОВАНИЯ В ПРЕОБРАЗОВАТЕЛЯХ КОДОВ ПО МЕТОДУ НАКОПЛЕНИЯ ЭКВИВАЛЕНТОВ

Рассматривается базирующийся на программном средстве "TRANSFORMATION" статистический подход к определению основных характеристик преобразователей кодов, использующих метод накопления эквивалентов и параллельную стратегию преобразования чисел.

1. Постановка задачи

Преимуществами преобразователей кодов по методу накопления эквивалентов {ПК НКЭ} являются умеренные аппаратные затраты и возможность изменения в широких пределах соотношения между числом тактов преобразования (быстродействием) и аппаратными затратами путем выбора числа шагов преобразования, оптимальных значений шагов, числа блоков разбиения многоразрядного ПК НКЭ и стратегии использования шагов преобразования (последовательной или параллельной).

Стратегия применения шагов преобразования может быть последовательной или параллельной [1,2].

При последовательной стратегии в многоразрядных ПК НКЭ преобразование чисел ведется путем уменьшения значений преобразуемых цифр (в тех разрядах показания, где значение равно или превышает значение шага) на величину большего шага. При этом формируется соответствующий эквивалент, который на каждом такте преобразования добавляется к содержимому накапливающего сумматора.

Если же во всех преобразуемых разрядах значения цифр оказываются меньше большего шага, то происходит переход на более меньший шаг и т. д. вплоть до использования шага 1 и обнуления всех разрядных счетчиков. Значения шагов преобразования должны выполняться при следующих ограничениях:

$$1 < a < b; a < b < c; b < c, K-1, \quad (1)$$

где K – основание системы счисления на входе; a, b, c – соответственно второй, третий и четвертый шаги преобразования (первый шаг преобразования всегда равен 1).

Последовательная стратегия по сравнению с параллельной структурно реализуется достаточно просто. Оценки сверху для числа тактов преобразования ПК НКЭ являются несколько завышенными, что привело к необходимости разработки программного средства, позволяющего найти не оценки сверху для числа тактов преобразования, а математическое ожидание M и дисперсию D числа тактов преобразования.

Применение параллельной стратегии преобразования чисел в ПК НКЭ приводит к уменьшению числа тактов преобразования (увеличению быстродействия) из-за того, что в различных преобразуемых разрядах разрешается одновременное вычитание различных шагов [3].

Параллельная стратегии применима только в ПК НКЭ с числом шагов преобразования равным двум и более. Для нахождения основных статистических характеристик параллельной стратегии (математического ожидания и дисперсии) и сравнения этих характеристик для последовательной и параллельной стратегий разработано программное средство "TRANSFORMATION" [4].

Целью настоящей работы является:

- рассмотрение назначения и возможностей разработанного программного средства;
- анализ функционирования программного средства в режиме "СТАТИСТИКА";
- разработка алгоритма выбора основных характеристик (наборов шагов преобразования) преобразователей кодов с параллельной стратегией преобразования чисел на основе статистических свойств параллельной стратегии.

2. Структура и функционирование ПК параллельного типа

Реализация логики управления в двухшаговом ПК параллельного типа выполнена таким образом, чтобы запретить возможность вычитания шага 1, если в этом разряде имеется возможность вычитания шага а. И наоборот, если значение разряда x_m находится в пределах $1 \leq x_m < a$, то следует разрешить опрос вентиля, управляющего вычитанием 1 из разрядного счетчика, хранящего x_m .

Дальнейшего повышения быстродействия ПК можно достичь или путем увеличения числа шагов в наборе до трех 1, а, b (b – третий шаг) при сохранении принципа последовательного использования шагов преобразования (вначале b, затем а и в заключение шаг 1), или путем параллельного (одновременного) вычитания различных шагов из различных разрядных цифр. Усиление локального параллелизма достигается за счет допущения возможности одновременного использования различных шагов преобразования в различных старших разрядах числа.

Выражение для формулы максимального числа тактов преобразования двухшагового ПК с параллельным применением шагов преобразования получить в явном виде затруднительно. Определить же значение максимального числа тактов преобразования можно путем моделирования процесса преобразования для различных наборов шагов или по формуле:

$$N_2^{\text{max}}(K) = [(x_i - (a - 1) \cdot x_i / a], (a - 1)], x_i = \overline{(K - 1)}, a, \quad (2)$$

где K – основание системы счисления на входе; $[\]$ – означают округление до меньшего целого; а скобки $[\]$ – означают выбор большего из чисел, каждое из которых заключено в круглые скобки.

Анализ формулы (2) показывает, что для $K = 12$ в двухшаговом ПК с параллельным использованием шагов преобразования минимальное значение числа тактов преобразования равно пяти для наборов (1, 3); (1, 4); (1, 5). Подобный анализ, проведенный для наборов 1,6 - 1,11, дает рост числа тактов преобразования до шести – десяти.

Повышение быстродействия двухшаговых ПК с параллельным использованием шагов преобразования достигается ценой некоторого усложнения ФЭ и ЛЭ для управления вычитанием шагов а и 1 из содержимого разрядных счетчиков.

Этот принцип управления реализуется в двухшаговом ПК параллельного типа путем замены (n-1) входного элемента ИЛИ-НЕ блоком инверторов (инверторы 17₁, 17₂). Вход каждого из них связан с единичным выходом соответствующего триггера старшего регистра состояний РГ₄, а выход соответствующего инвертора соединяется с управляющими входами схем И 8₁, И 8₂, на информационные входы которых поступают сигналы с единичных выходов триггеров этого разряда младшего регистра состояний РГ₃ (рис. 1).

Закон функционирования ФЭ для двухшагового ПК параллельного типа для набора шагов 1,2 и K=12 приведен в табл. 1.

Таблица 1

Номер набора	Состояние триггеров		Общий вид эквивалента	Десятичный код эквивалента	Двоичный код эквивалента		
	Второй группы	Первой группы			У ₁₂ У ₁₁ У ₁₀ У ₉	У ₈ У ₇ У ₆ У ₅	У ₄ У ₃ У ₂ У ₁
Z _i	D ₃ D ₂ D ₁	C ₃ C ₂ C ₁	S	S ₁₀	У ₁₂ У ₁₁ У ₁₀ У ₉	У ₈ У ₇ У ₆ У ₅	У ₄ У ₃ У ₂ У ₁
0	00	00	X ₀	X ₀	Трансляция младшей тетрады		
1	00	01	K ¹	10	0000	0000	1010
2	00	10	K ²	10	0000	0110	0100
3	00	11	K ² + K ¹	110	0000	0110	1110
4	01	01	aK	40	0000	0010	1000
5	01	11	K ² + aK	140	0000	1000	1100
6	10	10	aK ²	400	0001	1001	0000
7	10	11	aK ² + K	410	0001	1001	1010
8	11	11	aK ² + aK	440	0001	1011	1000

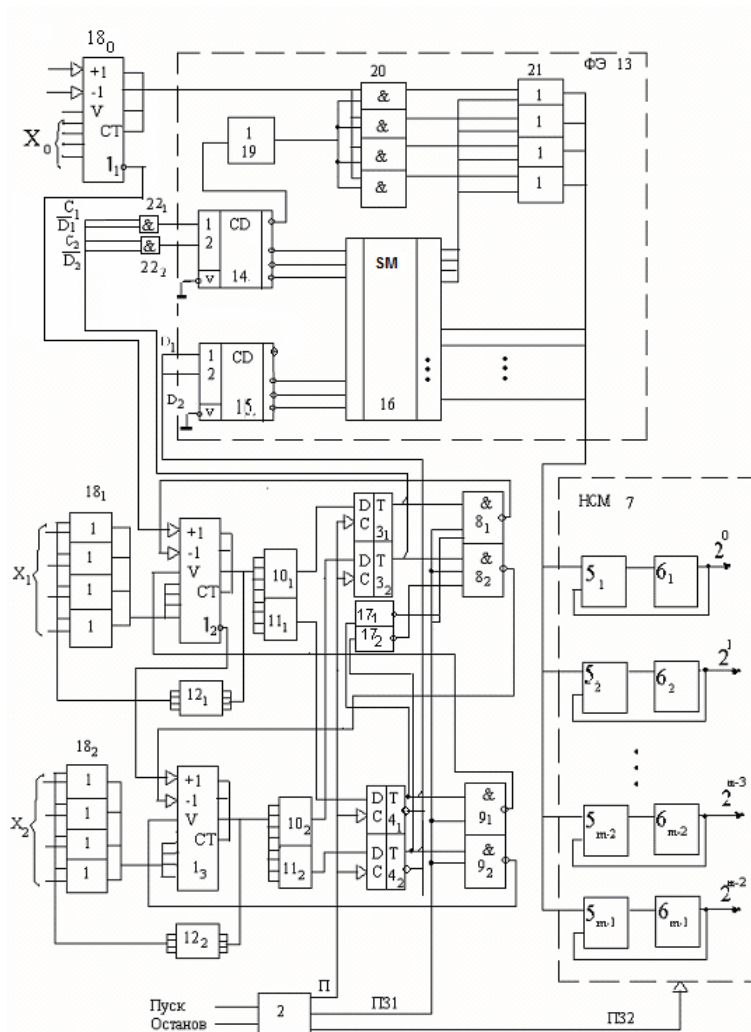


Рис. 1. Двухшаговый ПК с параллельным использованием шагов

3. Назначение и возможности программного средства ” TRANSFORMATION”

Программное средство ”TRANSFORMATION” имеет следующие возможности :

- имитация работы ПК НКЭ с числом шагов преобразования в наборе от 1 до 4 с основанием системы счисления на входе от 3 до 15 и разрядностью n преобразуемых цифр от 2 до 12;

- реализация двух режимов работы: режима преобразования заданного числа (или числа, сформированного генератором случайных чисел) и режима “ СТАТИСТИКА” для нахождения математического ожидания числа тактов преобразования и дисперсии.

Математическое ожидание и дисперсия определяются для всего множества преобразуемых чисел. Для разработки данного программного продукта была выбрана наиболее оптимальная по скорости проектирования и эффективности визуальная среда программирования QT для Windows XP/7, программа реализована на языке C++.

Для функционирования программы в составе компьютера необходимо иметь процессор с быстродействием не менее 1ГГц, 512 Мб ОЗУ, 50 Мб дискового пространства, Windows XP/Vista/7. Разработанное программное обеспечение сформировано как независимый модуль и для своего функционирования не требует дополнительного программного обеспечения. После запуска программного средства следует активизировать режим “ СТАТИСТИКА”. Код функции, реализующий режим статистики, приведен ниже:

```
do
{
int digit = 0;
for (int i = 0; i < amount_of_digits; ++i)
```

```

if (value[i] > digit)
digit = value[i];
int amountOfCycles = 0;
do
{
int equivalent = 0;
for (int j = 0; j < stepsCount; ++j)
{
quint8 step = stepsSizes.at(j).toInt();
if (step <= digit && step > equivalent)
equivalent = step;
}
digit -= equivalent;
if (equivalent == 0)
continue;
amountOfCycles++;
totalAmountOfCycles++;
}
while (digit > 0);
convertedValues[amountOfCycles]++;
amountOfCyclesPerConversation.append(amountOfCycles);
if (amountOfCycles > maxAmountOfCycles)
maxAmountOfCycles = amountOfCycles;
totalAmountOfValues++;
}
}

```

После активации режима “ СТАТИСТИКА ” появляется окно этого режима.

В этом окне имеется пять текстовых полей для ввода данных, три из которых снабжены кнопками инкремента и декремента. С помощью этих кнопок следует задать значение основания, количество преобразуемых цифр, число шагов, величину шагов.

Когда все значения введены, следует нажать поле “Собрать данные”. Программа после вычисления отобразит значения математического ожидания, дисперсии, количества всех проанализированных чисел и распределение количеств чисел по числу тактов преобразования, на основании которого можно построить гистограмму распределения .

Результат работы программного средства в режиме “ СТАТИСТИКА ” для $K=10$ и $n=6$ приведен на рис. 2 .

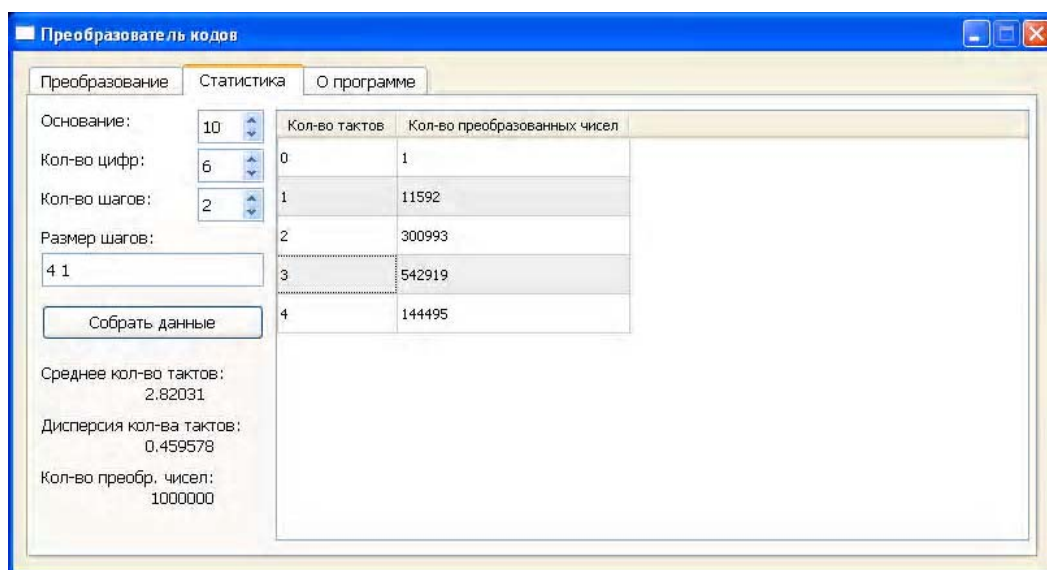


Рис. 2. Результаты статистических расчетов для $K=10$ и $n=6$

Математическое ожидание M рассчитывается по формуле

$$M = \left(\sum_{i=0}^{i=N_{\max}} i \cdot N_i \right) / 2^n, \quad (3)$$

где M – математическое ожидание числа тактов преобразования i ; N_i – количество чисел, преобразуемых за i тактов; N_{\max} – максимальное значение числа тактов преобразования (зависит от основания системы счисления K и значений шагов преобразования a, b, c); n – число входных преобразуемых разрядов.

Затем рассчитывается дисперсия преобразования D по формуле

$$D = \left(\sum_{j=1}^{j=2^n} (M - N_j)^2 \right) / 2^n (2^{n-1}), \quad (4)$$

здесь N_j – количество тактов, за которое преобразуется число Aa_j .

Анализ табл.2 показывает, что параллельная стратегия использования шагов для $K=10$, $n = 2-8$ и набора шагов 1 и 4 дает существенный среднестатистический выигрыш по быстродействию порядка 30%.

Таблица 2

T_i	Разрядность n							
	2	3	4	5	6	7	8	
0	1	1	1	1	1	1	1	
1	12	18	384	2122	11592	61868	325344	
2	33	327	3201	31143	300993	2889687	27564801	
3	39	435	4719	50763	542919	5774836	61097439	
4	15	169	1695	15967	144495	1273609	11012405	
T_{cp}	2,55	2,7	2,77	2,81	2,82	2,827	2,83	
M	0,85	0,69	0,59	0,52	0,46	0,41	0,37	

4. Алгоритм выбора основных параметров и системных характеристик ПК НКЭ параллельного типа с помощью программного средства “TRANSFORMATION”

Алгоритм состоит из следующих шагов:

1. По заданному основанию системы счисления на входе определяем возможные наборы шагов.

2. Активируем режим “СТАТИСТИКА”.

3. Задаем исходные параметры режима “СТАТИСТИКА”: основание системы счисления, количество шагов преобразования, размер шагов, разрядность преобразуемых чисел, устанавливая при первом шаге минимальное значение разрядности.

4. Инициуем активность кнопки “СОБРАТЬ ДАННЫЕ” и получаем значения T_{cp} и M .

5. Переходим к п.3 и увеличиваем разрядность на единицу, не изменяя значений основания системы счисления на входе, количества шагов преобразования и размера шагов.

6. Проверяем значение разрядности преобразуемого числа. Если оно является максимальным, то переходим к п.3 и устанавливаем новое значение размера шагов и минимальное значение разрядности преобразуемого числа, не меняя основания системы счисления и количества шагов преобразования. Если значение разрядности преобразуемого числа не является максимальным значением разрядности, то переходим к п. 5.

7. Процесс получения статистических значений T_{cp} и M заканчивается при достижении максимального значения разрядности анализируемого преобразования на наибольшем исследуемом наборе шагов.

В табл.3 приведены значения T_{cp} и M для $K=10$, а в табл.4 – для $K=12$.

Таблица 3

Значения шагов		Разрядность n						
		2	3	4	5	6	7	8
1 и 2	T _{ср}	3,35	3,78	4,03	4,21	4,34	4,43	4,51
	M	1,53	1,08	0,81	0,64	0,58	0,44	0,38
1 и 4	T _{ср}	2,55	2,7	2,77	2,81	2,82	2,827	2,83
	M	0,85	0,69	0,59	0,52	0,46	0,41	0,37
1 и 8	T _{ср}	3,63	3,56	3,33	3,09	2,86	2,66	2,5
	M	4,83	5,15	5,24	5,03	4,61	4,09	3,55

Таблица 4

Значения шагов		Разрядность n					
		2	3	4	5	6	7
1 и 2	T _{ср}	4,014	4,52	4,83	5,03	5,18	5,297
	M	2,14	1,49	1,11	0,86	0,697	0,581
1 и 4	T _{ср}	3,15	3,48	3,70	3,87	4,01	4,13
	M	1,49	1,33	1,22	1,12	1,03	0,94
1 и 8	T _{ср}	3,60	3,55	3,45	3,39	3,358	3,36
	M	3,43	3,087	2,67	2,23	1,83	1,50

Анализ указанных таблиц показывает, что наиболее выгодным набором шагов по критериям быстродействия и дисперсии для обоих оснований является набор шагов 1 и 4.

Для систем счисления с более высокими значениями оснований набор шагов преобразования с лучшими статистическими характеристиками может быть другим. Например, для $K=24$ меньшее значение среднего времени преобразования соответствует набору шагов 1 и 8.

Выводы

Научная новизна выполненного исследования заключается в следующем.

1. Рассмотрена и проанализирована параллельная стратегия использования шагов преобразования в двухшаговых ПК НКЭ.
2. Получена формула для вычисления числа тактов преобразования 2-шагового ПК НКЭ параллельного типа .
3. Проанализировано назначение и возможности программного средства "TRANSFORMATION" в режиме "СТАТИСТИКА".
4. Проведен сравнительный анализ статистических характеристик (математического ожидания и дисперсии) в ПК НКЭ с параллельной стратегией преобразования чисел для ряда оснований систем счисления на входе и разрядностей.

Практическая значимость исследования состоит в доказательстве преимущества параллельной стратегии перед последовательной и в возможности быстрого автоматизированного расчета основных статистических характеристик ПК НКЭ с параллельной стратегией преобразования чисел на этапе системного проектирования .

Список литературы: 1. А.С. 1647908 5НОЗМ 7/12. Преобразователь двоично-К-ичного кода в двоичный код / Н.Я.Какурин, Ю.К. Кирьяков, А.Н. Макаренко // Открытия, изобретения. 1991. №17. С. 262-263. 2. Какурин Н.Я., Лопузин Ю.В., Быкова Н.Н. Программное средство для анализа преобразований чисел // АСУ и приборы автоматики. 2007. Вып. 140. С.96-102. 3. Какурин Н.Я., Вареца В.В., Коваленко С.Н. Параллельная стратегия использования шагов в двухшаговых преобразователях кодов // АСУ и приборы автоматики. 2007. Вып. 141. С.29-36. 4. Какурин Н.Я., Бочаров Е.В., Вареца В.В., Полежаев К.В., Замалеев Ю.С. Программное средство для анализа преобразований чисел в преобразователях кодов параллельного типа // АСУ и приборы автоматики. 2011. Вып. 154. С.83-90.

Поступила в редколлегию 22.11.2011

Вареца Виталий Викторович, аспирант кафедры АПВТ ХНУРЭ. Научные интересы: проектирование программного обеспечения, автоматизация проектирования цифровых устройств. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

Лопухин Юрий Владимирович, ст. преподаватель кафедры АПВТ ХНУРЭ. Научные интересы: проектирование программного обеспечения, автоматизация проектирования цифровых устройств. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

Какурин Николай Яковлевич, канд. техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: прикладная теория цифровых автоматов, автоматизация проектирования цифровых устройств. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-326.

Кобылин Олег Анатольевич, канд. техн. наук, доцент кафедры информатики ХНУРЭ. Научные интересы: спектральный анализ изображений, программирование. Адрес: Украина, 61166, Харьков, пр.Ленина, 14, тел. 70-21-419.

Макаренко Анна Николаевна, канд. техн. наук, доцент кафедры информационных технологий Харьковского института банковского дела. Научные интересы: информационные технологии, анализ и синтез преобразователей код-код. Адрес: Украина, 61074, Харьков, пр. Победы, 55, тел. 336-05-64.

ПОЗИЦИОННОЕ СТРУКТУРНО-ВЕСОВОЕ КОДИРОВАНИЕ БИНАРНОГО ПРЕДСТАВЛЕНИЯ ТРАНСФОРМАНТ

Рассматриваются особенности кодирования битового представления трансформанты с учетом выявленных закономерностей двоичных структур на основе позиционного структурно-весового (ПСВ) кодирования. Доказывается, что технология ПСВ кодирования обладает двумя механизмами компенсации влияния структурных особенностей двоичного формата трансформанты на количество бит сжатого представления, приходящегося в среднем на один элемент, а именно: формирование длин для бинарных серий; построение системы оснований ПСВ числа для каждого массива длин бинарных серий.

1. Введение

Развитие инфокоммуникационных систем осуществляется в соответствии с концепцией создания сетей следующего поколения (NGN). При этом основная доля расширения услуг связана с предоставлением мультимедийной информации [1, 2]. Проведенный анализ особенностей доставки видеoinформации с использованием инфокоммуникационных систем показал, что заданное время передачи обеспечивается только для кадров с низким пространственным разрешением. Для преодоления сложившегося дисбаланса между временем передачи кадров изображений и необходимым временем требуется обеспечить снижение скорости передачи видеопотока на основе использования систем компрессии.

Анализ подходов относительно построения систем сжатия видеоданных в инфокоммуникациях выявил, что они базируются на использовании JPEG ориентированных технологий [3 – 4]. С помощью таких систем сжатия обеспечивается передача изображений нормального SD качества по высокоскоростным каналам связи. Однако такая возможность достигается только при наличии потерь качества реконструируемых изображений на уровне пикового отношения сигнал/шум, равного 30дБ.

Значит, существующие технологии компрессии изображений с использованием JPEG ориентированных технологий не обеспечивают требуемого уровня битовой скорости сжатого видеопотока. Следовательно, снижение битовой скорости компрессированных видеоданных для повышения качества предоставления услуг с использованием средств телекоммуникаций является *актуальной научно-прикладной задачей*.

Снижение битовой скорости трансформированных изображений предлагается организовывать на основе совершенствования технологии кодирования бинарного представления, которая обладает потенциалом для снижения битового потока и времени обработки транс-

формант в условиях заданного качества реконструкции изображений для различной степени их насыщенности мелкими объектами.

Процесс сжатия трансформанты $Y_{m,n}$, рассматриваемой на битовом уровне описания, заключается в выявлении закономерностей для двоичных структур. Поэтому в первую очередь необходимо обосновать концептуальный подход для системы преобразований, включающей следующие составляющие [5]:

1) бинаризация компонент $Y_{\xi,\chi}$ трансформанты, т.е. формирование для ее компонентной структуры битового представления;

2) выявление структурных закономерностей для двоичного представления трансформанты $[Y_{m,n}]_2$;

3) кодирование битового представления трансформанты с учетом выявленных закономерностей двоичных структур.

При этом требуется обеспечить:

– снижение битовой скорости v_c компактного представления трансформанты без внесения дополнительных искажений;

– уменьшение количества операций на обработку трансформанты.

Наиболее ключевой является третья концептуальная составляющая. Отсюда *цель исследований* заключается в разработке кодирования битового представления трансформант для повышения степени сжатия изображений в условиях сохранения заданного качества визуализации изображений, предназначенных для передачи в инфокоммуникационных системах реального времени.

2. Построение позиционного структурно-весового кодирования бинарного представления трансформант

Выбор технологии кодирования требуется проводить с учетом особенностей выявления структурных характеристик для бинарных плоскостей трансформанты. Здесь необходимо учитывать, что путем описания одномерных структур одинаковых двоичных элементов длинами их серий достигается сокращение количества обрабатываемых данных и снижение битовой скорости. В то же время одномерные структуры выявляемых ограничений не учитывают наличие двумерных закономерностей.

Поэтому технологию кодирования последовательностей длин бинарных серий необходимо осуществлять в соответствии со следующими принципами:

1) обеспечивать снижение битовой скорости для различной частоты двоичных перепадов в плоскостях двоичных структур трансформант (ДСТ);

2) дополнительно учитывать избыточность, обусловленную двумерностью структурных закономерностей, которая остается не выявленной в случае одномерных двоичных серий;

3) обеспечивать возможность устойчивости к неконтролируемым искажениям в процессе передачи кодовых комбинаций по каналам связи с ошибками;

4) организация кодирования не должна приводить к увеличению количества операций, затрачиваемых на трансформацию сегмента. Рост количества операций должен находиться в линейной зависимости от увеличения количества обрабатываемых данных.

Среди различных стратегий кодирования наибольшее сокращение избыточности достигается в результате выявления пространственных ограничений на последовательности длин двоичных серий. Это обусловлено тем, что поэлементное кодирование заведомо менее эффективное, чем кодирование блоков. Поэтому чем больше длина обрабатываемой последовательности, на которой выявляются ограничения, тем потенциально большее количество избыточности можно устранить.

Реализовать такой подход можно на основе учета неравномерности динамических диапазонов длин бинарных серий, расположенных в пределах как одной двоичной плоскости, так и в различных плоскостях ДСТ. Одним из вариантов для учета такой особенности является формирование для последовательностей длин бинарных серий полиадических чисел (ПЧ) [5].

Полиадическим числом A называется такое число $\{\ell_1, \dots, \ell_\theta, \dots, \ell_\Theta\}$, основаниями g_θ элементов которого могут быть произвольные целые числа, так чтобы выполнялись неравенства

$$\ell_\theta < g_\theta, \theta = \overline{1, \Theta}. \quad (1)$$

Понятно, что в общем случае между основаниями элементов ПЧ выполняется условие

$$g_\xi \neq g_\gamma, \text{ где } \xi \neq \gamma, \xi, \gamma = \overline{1, \Theta}. \quad (2)$$

В целях сокращения количества оснований, как правило, они формируются сразу для нескольких элементов обрабатываемых данных. В этом случае образование полиадических чисел проводится для одномерных последовательностей, рассматриваемых как структурные составляющие массивов данных. В соответствии с этим на основе последовательности L длин бинарных серий, выявленных для ДСТ, строятся двумерные массивы $A_{k,u}$, $A_{k,u} = \{\ell_{s,p}\}$. Здесь $A_{k,u}$ – массив длин бинарных серий, сформированный для $(k;u)$ -й трансформанты; $\ell_{s,p}$ – $(s;p)$ -й элемент массива $A_{k,u}$; p – длина строки (количество столбцов) в массиве $A_{k,u}$; S – длина полного столбца; s' – длина неполного столбца.

Последний столбец массива может быть заполненным не полностью, т.е. $s' \leq S$. Это вызвано тем, что количество Φ бинарных серий для ДСТ не всегда будет кратным заранее выбранной длине столбца S .

В этом случае полиадические числа строятся на основе отдельных столбцов массива ДБС. В результате для одного массива получаем p полиадических чисел, т.е. $A_{k,u} \rightarrow \{A^{(1)}, \dots, A^{(p)}, \dots, A^{(p)}\}$. При этом последнее полиадическое число $A^{(p)}$ может состоять из s' элементов.

При таком варианте образования ПЧ основания определяются для строк массива $A_{k,u}$. Тогда основание g_s будет общим для элементов s -й строки массива $A_{k,u}$. По определению основания полиадического числа должно выполняться неравенство (1), тогда для величины g_s должно выполняться условие $g_s > \max_{1 \leq p \leq P} \{\ell_{s,p}\}$, $s = \overline{1, S}$. Отсюда наименьшее допустимое значение для основания ПЧ будет вычисляться как

$$g_s = \max_{1 \leq p \leq P} \{\ell_{s,p}\} + 1 \text{ для } s \leq s'; \quad g_s = \max_{1 \leq p \leq P-1} \{\ell_{s,p}\} + 1 \text{ для } s > s'$$

или, если ввести обобщающую функцию $P' = P - (1 - \text{sign}(\text{sign}(s' - s) + 1))$, то

$$g_s = \max_{1 \leq p \leq P'} \{\ell_{s,p}\} + 1. \text{ Здесь получаем } P' = \begin{cases} P, & \rightarrow s \leq s'; \\ P-1, & \rightarrow s > s'. \end{cases}$$

Отсюда для массива $A_{k,u}$ формируется одномерная система оснований G , включающая в себя S компонент, а именно $G = \{g_1, \dots, g_s, \dots, g_S\}$. Для предложенной системы оснований весовые коэффициенты W_s элементов ПЧ определяются как накопленное произведение

оснований младших элементов, т.е. $W_s = \prod_{\xi=s+1}^S g_\xi$. Весовой коэффициент элемента ПЧ

зависит от его позиции в числе и в отличие от позиционного числа имеет неравномерное приращение, зависящее от значений оснований между рассматриваемыми элементами. Например, для s -го элемента относительно $(s+1)$ -го элемента весовой довесок равен

$$g_{s+1}, \quad g_{s+1} = W_s / W_{s+1} = \prod_{\xi=s+1}^S g_\xi / \prod_{\xi=s+2}^S g_\xi.$$

С другой стороны, весовой довесок для $(s-1)$ -го элемента относительно s -го элемента

равен g_s , $g_s = W_{s-1} / W_s = \prod_{\xi=s}^S g_\xi / \prod_{\xi=s+1}^S g_\xi$. Отсюда в соответствии с условием (2) получим

$$g_s \neq g_{s+1}.$$

Поэтому полиадическое число в общем случае трактуется как позиционно-неравновесное число (ПНЧ) – весовой коэффициент элементов зависит не только от их позиции, но и от значений оснований младших элементов.

В соответствии с данным определением кодовое отображение C_p полиадического числа находится по следующей формуле:

$$C_p = \ell_{1,p} \prod_{\xi=2}^S g_{\xi} + \dots + \ell_{s,p} \prod_{\xi=s+1}^S g_{\xi} + \dots + \ell_{S-1,p} g_S + \ell_{S,p} = \sum_{s=1}^S \ell_{s,p} \prod_{\gamma=s+1}^S g_{\gamma}, \quad (3)$$

или $C_{p,1} = \ell_{1,p} \prod_{\xi=2}^S g_{\xi}$; $C_{p,2} = C_{p,1} + \ell_{2,p} \prod_{\xi=3}^S g_{\xi}$; \dots ; $C_p = \sum_{\xi=1}^{S-1} C_{p,\xi} + \ell_{S,p}$.

Следовательно, на каждом шаге кодирования формируется слагаемое $C_{p,\xi}$ на основе старшего из необработанных элементов ПЧ. Это позволяет заключить, что кодирование полиадических чисел осуществляется по схеме кодирования старших элементов позиционно-неравновесного числа.

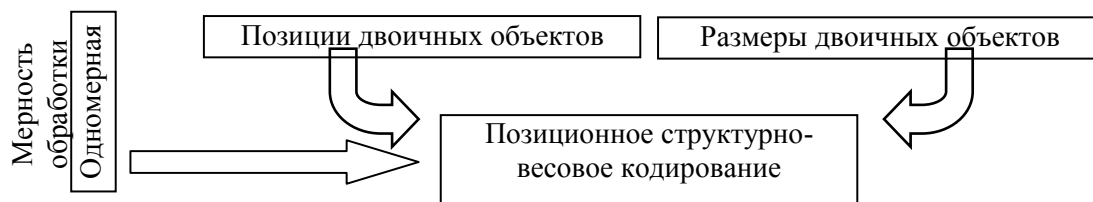
Сформулируем интерпретацию кодирования ПНЧ для обработки двоичных структур трансформант. В выражении (3) величины $\ell_{s,p}$ представляют собой одномерную структурную единицу содержания битовых плоскостей, а именно длину бинарной серии. Поэтому в данном случае полиадическое кодирование можно рассматривать как интегрированное представление взвешенных структурных составляющих двоичного формата трансформанты. При этом весовые характеристики структурных составляющих зависят от их позиционирования в ДСТ. Следовательно, такое кодирование можно интерпретировать как позиционно-структурно-весовое кодирование.

Определение 1. Позиционным структурно-весовым (ПСВ) числом называется такое позиционное число, когда его элементами являются взаимодополняющие (непересекающиеся) структурные составляющие обрабатываемых форматов данных и в общем случае выполняется условие $g_{\xi} \neq g_{\gamma}$, где $\xi \neq \gamma$, $\xi, \gamma = 1, \Theta$.

Определение 2. Позиционным структурно-весовым кодированием называется интегрированное представление взвешенных структурных составляющих, заданное выражением (3).

Позиционное структурно-весовое кодирование как технологию обработки двоичных форматов данных можно рассматривать в виде варианта обобщения двух типов структурных характеристик, а именно позиционирования и размеров двоичных объектов (рисунок). Размеры двоичных областей учитываются в результате выявления длин двоичных серий, а позиции областей – на основе зависимости весовых коэффициентов длин серий от их позиции в битовом представлении.

Тип выявляемых характеристик



Место позиционно-неравновесного представления в системе классификации методов битовой обработки

В свою очередь значение кода ПСВ числа является количественной характеристикой частоты бинарных перепадов. Для фиксированной длины столбца (длина ПСВ числа фиксирована, количество бинарных элементов ДСТ является переменным) значение ПСВ кода будет уменьшаться по мере увеличения частоты двоичных перепадов, т.е. уменьшения длин серий. Однако с другой стороны будет увеличиваться количество ПСВ кодов.

Для фиксированного количества бинарных элементов двоичной структуры трансформанты (в этом случае формируются ПСВ числа переменной длины) значение ПСВ кода будет, наоборот, увеличиваться с ростом частоты двоичных перепадов. Если теперь зафиксируем длину машинного слова, то для переменной длины ПЧ мы уложим туда больше элементов ДСТ, чем будут больше длины построенных для них бинарных серий.

Объем $V(C_p)_c$ сжатого представления столбца массива бинарных серий оценивается как количество разрядов на представления соответствующего полиадического кода C_p ,

$$V(C_p)_c = [\ell \log_2 C_p] + 1 = [\ell \log_2 (\sum_{s=1}^S \ell_{s,p} \prod_{\gamma=s+1}^S g_\gamma)] + 1.$$

Величина кода полиадического числа ограничена сверху величиной накопленного произведения оснований его элементов, т.е.

$$C_p \leq (\prod_{s=1}^S g_s) - 1,$$

где $\prod_{s=1}^S g_s$ – накопленное произведение оснований элементов ПЧ.

Максимальное значение $v(\max)_c$ объема кодового представления столбца массива ДБС оценивается по следующей формуле:

$$v(\max)_c = [\ell \log_2 (\prod_{s=1}^S g_s - 1)] + 1 \leq [\ell \log_2 (\prod_{s=1}^S g_s)] + 1 = [\sum_{s=1}^S \ell \log_2 g_s] + 1.$$

Усреднение объемов $V(C_p)_c$ и $v(\max)_c$, приходящееся на один элемент ДСТ, оценивается по формулам:

$$\bar{v}_b = V(C_p)_c / S \sum_{s=1}^S \ell_{s,p} = ([\ell \log_2 (\sum_{s=1}^S \ell_{s,p} \prod_{\gamma=s+1}^S g_\gamma)] + 1) / S \sum_{s=1}^S \ell_{s,p};$$

$$\bar{v}(\max)_b = ([\sum_{s=1}^S \ell \log_2 g_s] + 1) / S \sum_{s=1}^S \ell_{s,p},$$

где \bar{v}_b , $\bar{v}(\max)_b$ – среднее и максимальное количество бит, приходящихся на один двоичный элемент бинарного формата трансформанты.

Анализ выражений показывает, что:

1) с одной стороны, с ростом длины бинарной серии величина кода ПСВ числа будет увеличиваться. Но с другой стороны за счет увеличения длины бинарной серии среднее количество бит на один элемент двоичной структуры трансформанты будет уменьшаться, т.е. будет происходить компенсация повышения значения кода ПСВ числа;

2) наоборот, при сокращении длины бинарной серии уменьшаются значения оснований ПСВ числа и, как следствие снижается значение его кода, что приводит к компенсации роста среднего количества бит, приходящегося на один элемент ДСТ.

Значит, предложенная технология ПСВ кодирования обладает двумя механизмами компенсации влияния структурных особенностей двоичного формата трансформанты на количество бит сжатого представления, приходящегося в среднем на один элемент ДСТ, а именно:

- формирование длин для бинарных серий;
- построение системы оснований ПСВ числа для каждого массива длин бинарных серий.

В итоге это приводит к снижению битовой скорости сжатого представления ДСТ для различного структурного содержания двоичного формата трансформанты, т.е. для различной частоты двоичных перепадов.

Сжатие изображений в результате позиционного структурно-веса кодирования достигается за счет сокращения следующих видов избыточности: структурной избыточности, обусловленной наличием бинарных серий в двоичном формате трансформанты; комбинаторной избыточности, вызванной наличием неравномерности в длинах бинарных серий как для двоичных плоскостей, так и между ними.

Таким образом, разработаны базовые составляющие концепции сжатого представления бинарного формата трансформанты, а именно:

- 1) бинаризация компонент трансформант проводится на основе их разложения с использованием полинома по основанию два;
- 2) выявление структурных характеристик в двоичном представлении трансформанты осуществляется путем формирования длин бинарных серий;
- 3) сокращение пространственной избыточности в битовом представлении трансформанты без внесения погрешности с учетом выявленных структурных закономерностей организуется на основе позиционного структурно-неравновесного кодирования.

Выводы

1. Кодирование битового представления трансформанты с учетом выявленных закономерностей двоичных структур проводится на основе позиционного структурно-весового кодирования. В этом случае реализуется интегрированное представление взвешенных структурных составляющих двоичного формата трансформанты, причем весовые характеристики структурных составляющих зависят от их позиционирования в ДСТ. Это позволяет учитывать неравномерность динамических диапазонов длин бинарных серий, расположенных в пределах как одной двоичной плоскости, так и в различных плоскостях ДСТ.

2. Построенная технология ПСВ кодирования обладает двумя механизмами компенсации влияния структурных особенностей двоичного формата трансформанты на количество бит сжатого представления, приходящегося в среднем на один элемент ДСТ, а именно: формирование длин для бинарных серий; построение системы оснований ПСВ числа для каждого массива длин бинарных серий. Это приводит к снижению битовой скорости сжатого представления ДСТ для различного структурного содержания двоичного формата трансформанты, т.е. для различной частоты двоичных перепадов.

Список литературы: 1. *Аудиовизуальные системы связи и вещания: новые технологии третьего тысячелетия, задачи и проблемы внедрения в Украине* / [О.В. Гофайзен, А.И. Ляхов, Н.К. Михалов и др.] // Праці УНДІРТ. 2000. № 3. С. 3-40. 2. *Олифер В.Г.* Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. / В.Г. Олифер, Н.А. Олифер. СПб.: Питер, 2006. 958 с. 3. *Gonzales R.C.* Digital image processing / R.C. Gonzales, R.E. Woods. Prentice Inc. Upper Saddle River, New Jersey 2002. 779 p. 4. *Баранник В.В.* Кодирование трансформированных изображений в инфокоммуникационных системах / В.В. Баранник, В.П. Поляков. Х.: ХУПС, 2010. 212 с. 5. *Баранник В.В.* Метод сжатия изображений на основе неравновесного позиционного кодирования битовых плоскостей / В.В. Баранник, Н.К. Гулак, Н.А. Королева // Радіоелектронні і комп'ютерні системи. Х.: ХНАУ "ХАІ", 2009. Вип. 1. С. 55–61.

Поступила в редколлегию 05.12.2011

Баранник Владимир Викторович, д-р техн. наук, профессор, ведущий научный сотрудник научного центра Харьковского университета Воздушных Сил им. Ивана Кожедуба. Научные интересы: системы, технологии преобразования, кодирования, защиты и передачи информации, семантической обработки изображений. Адрес: Украина, 61023, Харьков, ул. Сумская, 77/79, тел. 8 050-3038971.

Красноруцкий Андрей Александрович, инженер Харьковского университета Воздушных Сил им. Ивана Кожедуба. Научные интересы: системы, технологии преобразования, кодирования, защиты и передачи информации. Адрес: Украина, 61023, Харьков, ул. Сумская, 77/79.

Хаханова Анна Владимировна, канд. техн. наук, доцент кафедры АПВТ ХНУРЭ. Научные интересы: технологии преобразования, кодирования и передачи информации. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, комн. 321.

ОПТИМИЗАЦИЯ РЕЖИМОВ АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ КОМПРЕССОРНОЙ СТАНЦИЕЙ МАГИСТРАЛЬНОГО ГАЗОПРОВОДА

Рассматривается подход к оптимизации режимов управления компрессорными станциями магистрального газопровода с помощью нечетких моделей, позволяющих эффективно описывать процесс компримирования газа с учетом действующих возмущений и существующих ограничений. Идея заключается в создании модульного имитатора работы автоматизированных систем управления технологическими процессами компрессорной станции, что позволяет повысить качество подготовки операторов-технологов, в диалоговом режиме осуществляющих принятие решений по управлению ТП КС.

1. Введение

Основными объектами оперативного контроля и управления на магистральном газопроводе (МГ) являются:

- головная компрессорная станция и промежуточные компрессорные станции (КС), предназначенные для сообщения головному потоку дополнительной энергии для передачи газа на дальние расстояния. Компрессорные станции, как правило, являются многоцеховыми и оборудованы разнотипными агрегатами. При оперативном управлении на КС осуществляется изменение режимов работы агрегатов, а также подключение или отключение некоторых из них;

- трасса газопровода (участки между КС и отводы), на которой расположены линейные краны и станции электрохимической защиты;

- станции подземного хранения газа, с помощью которых выравнивается загрузка газопровода при суточных и сезонных изменениях потребления газа;

- газораспределительные станции (ГРС), предназначенные для редуцирования давления газа при подаче его потребителям. В настоящее время ГРС относят к автономному объекту управления и в схему оперативного контроля и управления не включают.

Функционирование МГ как сложной непрерывной транспортной системы связано с частыми изменениями в режиме работы управляемых агрегатов, вызванными неравномерностью поставок газа непосредственно с газопромысла, ситуациями на отдельных участках газопровода, нестабильностью в потреблении газа потребителями.

Компрессорная станция как локальный объект управления представляет собой сложный комплекс компрессорного оборудования [1]. На КС предусмотрены различные способы регулирования и управления режимами газопередачи. Одной из главных задач управления технологическим режимом работы КС является поддержание давления на входе КС на заданном уровне при оптимальном (в смысле некоторого критерия) распределении нагрузки между агрегатами. Центральное место в системе управления технологическими процессами занимает оператор-технолог, на долю которого приходится самая ответственная функция – принятие окончательного решения по управлению процессом, а его деятельность протекает в условиях высокой психической напряженности, ответственности и дефицита времени. Наличие пяти основных состояний технологического процесса (норма, отклонение, предавария, авария и катастрофа) требуют от оператора умения выполнения различных функциональных задач, отличающихся целями и критериями эффективности управления.

В иерархии автоматизированных систем управления технологическими процессами компрессорной станции (АСУ ТП КС) выделяют два основных уровня управления:

- верхний интерактивный уровень, предусматривающий выработку оператором-технологом текущих заданий (уставок) для локальных систем управления нижнего уровня. Здесь используются программно-аппаратные средства для оперативного определения режимных параметров ТП КС;

– нижний уровень, объединяющий локальные системы управления, предназначенные для стабилизации работы КС в соответствии с текущими уставками.

Эффективность реализации иерархического управления КС во многом зависит от качества решений, принимаемых оператором-технологом.

Целью данного исследования является разработка методов определения технологических режимов на верхнем уровне АСУ ТП КС. В соответствии с этим необходимо решить следующие задачи: определить последовательность расчета режимов с использованием методов нечеткой оптимизации; разработать структуру модульного имитатора работы АСУ ТП КС, который позволил бы повысить качество подготовки операторов-технологов.

2. Оптимизация технологических режимов в АСУ ТП КС

Рассмотрим задачу оптимизации, заключающуюся в нахождении технологического режима, при котором удовлетворяются требования к ТП компримирования газа. Для того чтобы найти оптимальный (с точки зрения стабильности давления на выходе КС) режим управления, необходимо определить зависимость между выходными и входными параметрами процесса. Основным технологическим объектом КС является газоперекачивающий агрегат (ГПА) с газотурбинным приводом, приводящим в движение центробежный нагнетатель, на котором непосредственно осуществляется компримирование газа.

В общем случае задачу нечеткой оптимизации можно сформулировать как определение входного вектора X , при котором некоторый нечеткий функционал стремится к нечеткому максимуму, т.е.:

$$\tilde{y}^* = \tilde{f}(Y) \rightarrow \text{m}\tilde{\alpha} \text{ при } \tilde{\varphi}_i(Y) \subseteq \tilde{X}_i, \tilde{\varphi}_i(Y) \subseteq \tilde{X}_i, \quad (1)$$

где \tilde{f} , $\tilde{\varphi}_i$ – нечеткие функции, определенные в m -мерном пространстве действительных чисел; \tilde{X}_i – нечеткие числа.

Задача нечеткой оптимизации компонент вектора выходных параметров Y ТП КС может быть представлена в следующем виде:

$$\tilde{F}(\tilde{y}^1, \tilde{y}^2, \dots, \tilde{y}^r, \tilde{x}_1^1, \tilde{x}_2^1, \dots, \tilde{x}_m^1, \tilde{x}_1^2, \tilde{x}_2^2, \dots, \tilde{x}_m^2, \dots, \tilde{x}_1^r, \tilde{x}_2^r, \dots, \tilde{x}_m^r) \rightarrow \text{m}\tilde{\alpha}(\text{m}\tilde{\alpha} \text{in}), \quad (2)$$

здесь нечеткая функция $\tilde{F}(\cdot)$ зависит от компонент вектора выходных параметров Y и компонент вектора входных параметров X ТП КС.

Содержание вектора X , формализующего сведения о параметрах технологического процесса, определяют следующие множества: X_1 – множество, определяющее давление газа на входе КС; X_2 – множество, определяющее температуру газа на входе КС; X_3^i – множество, определяющее давление на входе ГПА; X_4^i – множество, определяющее давление на выходе ГПА; X_5^i – множество, определяющее температуру на входе ГПА; X_6^i – множество, определяющее температуру на выходе ГПА; X_7^i – множество, определяющее количество оборотов ГПА; i – номер ГПА, $i=1, N$ (N – количество ГПА на КС).

Компоненты вектора выходных параметров Y определяют искомые значения уставок для локальных систем управления нижнего уровня.

Для принятия оператором-технологом решений по оптимизации работы КС необходима информация о нечетких множествах функции (2), определяющих давление газа на входе КС P_1 , давление газа на выходе КС P_2 , температуру газа на входе КС T_1 , температуру газа на выходе ГПА T_2 и количество оборотов ГПА V для каждого момента времени:

$$C[P_m, P_{1i}, P_{2i}, T_{1i}, T_{2i}, V_i, t], \quad i=1, N. \quad (3)$$

При этом нечеткие множества должны отражать факторы эффективности управления и допустимости расхода газа и соответствующих давлений с позиций увеличения пропускной способности газопровода. Так как цели, ограничения и состояние системы являются нечеткими, множество, соответствующее совокупности данных (3), может быть задано функцией принадлежности следующего вида:

$$\mu_p [P_m, P_{1i}, P_{2i}, T_{1i}, T_{2i}, V_i / t]. \quad (4)$$

В свою очередь нечеткое множество режимов работы i -го ГПА КС должно задавать совокупность частных ограничений и координирующих функций, учитывающих особенности ТП. Для отдельных ГПА функции принадлежности зададим в следующем виде:

$$\mu_p [P_m, P_{1i}, P_{2i}, T_{1i}, T_{2i}, V_i / t] = \mu_1 [P_m, V_i / t] \wedge \mu_2 [P_{1i}, P_{2i}, T_{1i}, T_{2i}, V_i / t] \wedge \mu_3 [P_m, P_{1i}, P_{2i}, T_{1i}, T_{2i}, V_i / t], \quad (5)$$

где $\mu_1 [P_m, V_i / t]$ – функция ограничения режима работы ГПА с учетом возможного разрушения участка МГ или поломки оборудования; $\mu_2 [P_{1i}, P_{2i}, T_{1i}, T_{2i}, V_i / t]$ – функция технологических ограничений режима работы i -го ГПА; $\mu_3 [P_m, P_{1i}, P_{2i}, T_{1i}, T_{2i}, V_i / t]$ – координирующая функция, характеризующая степень принадлежности режима работы i -го ГПА к множеству эффективных режимов, определяемому условиями работы всего МГ.

Определим нечеткие интервалы задания параметров модели. Для этого необходимо иметь информацию о законе распределения параметров $F(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_7)$ на плоскости и взаимном расположении его отдельных фрагментов с площадью d_i , имеющих различные весовые коэффициенты b_i . Будем считать, что для каждой d_i плотность распределения вероятностей постоянна. Тогда поставленную задачу можно свести к поиску максимума функции распределения следующего вида:

$$\varphi(\Delta x_1, \Delta x_2, \dots, \Delta x_n) = \sum_{i=1}^s b_i d_i f(\tilde{x}_{1i}, \Delta x_1; \tilde{x}_{2i}, \Delta x_2; \dots; \tilde{x}_{ni}, \Delta x_n), \quad (6)$$

где n – количество входных параметров.

Предположим, что контролируемые параметры независимы и распределены по нормальному закону, а $b_i = m_{b_i}$ (математическое ожидание весовых коэффициентов). В этом случае для ТП КС функция (6) преобразуется к следующему виду:

$$\varphi(\Delta x_1, \Delta x_2, \dots, \Delta x_7) = \sum_{i=1}^s m_{b_i} d_i f(\tilde{x}_{1i}, \Delta x_1; \tilde{x}_{2i}, \Delta x_2; \dots; \tilde{x}_{7i}, \Delta x_7). \quad (7)$$

Значения оптимального сдвига координат отдельных фрагментов закона распределения $f(\tilde{x})$ можно определить по методу оптимума номинала [2].

Алгоритмическая процедура формирования $f(\tilde{x})$ задается такой последовательностью операций:

- наблюдаемые значения компонент вектора входных параметров (функции принадлежности) X разбиваем по соответствующим интервалам таким образом, чтобы случайные колебания параметра внутри нечеткого интервала не оказывали существенное влияние на ТП при неизменном давлении на выходе КС. Сужение диапазона случайных изменений входных параметров снижает влияние измеряемых факторов на закон распределения управляющих параметров $f(\tilde{x})$;

- для каждого интервала по статистическим данным определяем соответствующие распределения компонент вектора входных параметров X КС. При этом для каждого значения давления на выходе КС определяется область возможных значений управляющих параметров, ограниченная n -мерным законом распределения компонент вектора входных параметров;

- разбиваем все пространство выходных параметров Y на области, соответствующие различным режимам работы КС.

Оценивая влияние каждой такой области на общую эффективность ТП, можно определить вид и параметры функции $f(\tilde{x})$.

Введем величину h_o – «степень сжатия», которую будем рассматривать как значение обобщенной лингвистической переменной $\tilde{X} = (\tilde{X}_7^1, \tilde{X}_7^2, \dots, \tilde{X}_7^i)$, заданной на области определения $\tilde{X}_7^1 \times \tilde{X}_7^2 \times \dots \times \tilde{X}_7^i$ и принимающей базовые значения h_{o_k} с функциями принадлежности μ_{o_k} , $k=1, K$, $K = |T_{X_7^1}| \times |T_{X_7^2}| \times \dots \times |T_{X_7^i}|$, где i – количество ГПА в цеху, \tilde{X}_7^i – множество, определяющее количество валов ГПА.

Эффективность режима функционирования КС можно оценить величиной:

$$\varphi = \sum_{i=1}^s b_i d_i f(h_o), \quad (8)$$

где b_i – весовые коэффициенты для ГПА.

Задача оптимизации сводится к определению такого расположения кривой $f(h_o)$, при котором величина φ достигает максимального значения φ_o .

Качество ТП считается удовлетворительным, если давление P_2 находится в диапазоне [9-10] МПа. Для определения значений b_i необходимо задать диапазоны i -х интервалов изменения давления, которые характеризуют чувствительность процесса компримирования газа к изменению степени сжатия h_o . Анализ ретроспективных данных по ТП компримирования газа позволил получить математическую модель процесса в виде линеаризованного уравнения следующего вида:

$$P_{out} = b_0 + b_1 h,$$

где P_{out} – выборочная оценка для давления газа на выходе КС; b_0, b_1 – выборочные оценки для управляющего фактора.

В соответствии с существующими нормами эксплуатации КС с центробежным нагнетателем зададим нулевой уровень степени сжатия равным 1,205. Затем, изменяя с заданным шагом Δh значения степени сжатия в интервале [1,205-1,42], получим данные, необходимые для определения b_0 и b_1 . На рис.1 приведен вид функции принадлежности, аргументом которой является обобщенная лингвистическая переменная «степень сжатия» (h_o) с разбиением диапазона ее изменения на полосы шириной 0,0165.

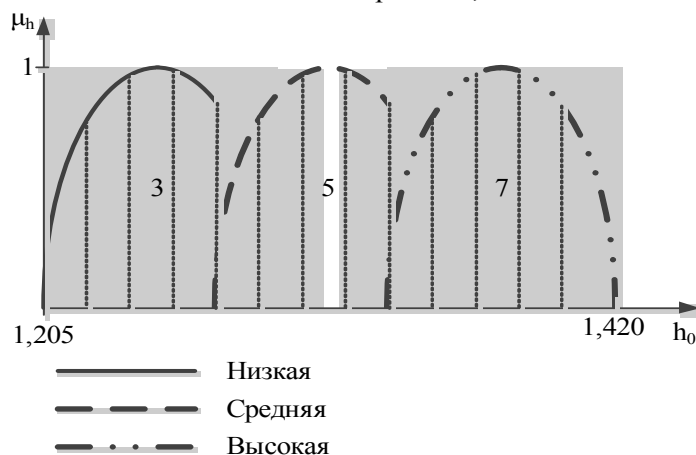


Рис. 1. Функции принадлежности для оценки режимов КС по степени сжатия

Полосы 3, 5, 7 соответствуют максимальным значениям функции принадлежности, однако только при значении степени сжатия, соответствующей полосе 7, на выходе КС будет поддерживаться давление от 9 до 10 МПа. Следовательно, эта полоса имеет большой вес и в относительных единицах оценивается величиной «+1», оставшимся полосам присваиваем в простейшем случае отрицательную оценку «-1». Используя полученные оценки, получаем оптимальные (в смысле критерия (8)) значения степени сжатия газа, гарантирующие эффективные режимы работы КС с поддержанием заданного давления на его выходе.

Следует отметить, что при управлении работой КС кроме задачи поддержания заданного давления на выходе должны быть решены также задачи стабилизации температуры газа на выходе КС и оптимизации расхода газа.

Для выбора оптимального (по расходу топливного газа) планового режима работы КС на интервале управления следует провести оценку эффективности функционирования КС МГ для различных комбинаций включения ГПА с учетом построенных для них функций принадлежности. В качестве оценки эффективности КС можно использовать математическое ожидание суммарных затрат топливного газа на компримирование заданного количества природного газа на интервале управления [5].

Таким образом, задачу оптимизации режимов работы КС МГ можно представить в виде оптимизации функционала (8), дополненную условиями стабилизации температуры газа на выходе КС и оптимизации расхода газа. При этом приоритетным моментом является получение требуемого значения давления газа на выходе КС.

Результаты моделирования подтверждают работоспособность предложенного подхода для оперативного выбора режимов управления ГПА КС.

3. Модульный компьютерный имитатор работы АСУ ТП КС

Разработанная схема модульного компьютерного имитатора автоматизированного управления режимами работы КС тренажерной системы состоит из рабочего места оператора, имитатора режимов и автоматизированной системы обучения оператора. Дисплейный модуль выполняет функции систем отображения информации и средств управления ТП КС. На экран дисплейного модуля выносятся задания, тексты, информация о ходе технологического процесса с изображением мнемосхем, графики и таблицы. Пакет режимной имитации содержит набор подпрограмм для моделирования системы в различных режимах функционирования технологического процесса [2]. Автоматизированная система обучения состоит из генератора заданий, программ контроля и оценки действий обучаемого. Общая модель системы включает в себя модель ТП, модели исполнительных механизмов и преобразователей, а также модель двухуровневой системы автоматизированного управления. Имитатор предусматривает возможность применения для управления ТП КС моделей с нечетким описанием исходных данных и нечетким логическим выводом. Режим работы компрессорной станции зависит от типов, схемы соединения, оборотов ГПА, температуры и состава газа и задается с использованием описанных выше рекомендаций. В блоке локального регулирования реализованы алгоритмы работы нечеткого цифрового регулятора с инверсной моделью, предложенные в [3] для управления ГПА.

Модель технологического процесса разделяется на три модели по последовательности расчета наблюдаемых технологических переменных процесса (положение исполнительных механизмов – физические выходные переменные процесса – измеренные значения). В ходе моделирования этот блок получает необходимую входную информацию от блока исполнительных механизмов и, таким образом, остается независимым от способов выработки управляющих воздействий. Модель процесса может подвергаться вмешательствам оператора m_4^s , имитирующего неисправности оборудования, изменения внешних условий, свойств сырья. Вырабатываемые физические значения переменных процесса $x=X(v, m_4^s)$ поступают в блок измерения информации для измерения переменных и расчета вспомогательных параметров.

Модель системы исполнительных механизмов получает из блока базового регулирования воздействия (выходы) на клапаны ГПА и дискретные команды по изменению скорости вращения валов ГПА. При этом она может подвергаться вмешательствам оператора m_3^s в целях имитации отказов исполнительных механизмов (клапанов, насосов, переключателей), вырабатывая необходимые для блоков модели ТП и измерения информации значения дискретных и непрерывных исполнительных органов $v=V(u, m_3^s)$. В блок измерения информации поступают выходные физические значения переменных от модели процесса и положения исполнительных механизмов из соответствующего блока (1). Оператор может вмешиваться в работу блока измерения (m_5^s) путем имитации отказов контрольно-измерительной аппаратуры. Вырабатываемые в этом блоке сигналы наблюдений $y=Y(x, v, m_5^s)$ используются в блоках базового регулирования и усовершенствованного управления и передаются для отображения в операторский и инструкторский интерфейсы.

Модель системы управления предложенного имитатора по иерархическому признаку данных разделена на нижний и верхний уровни, соответствующие локальному и режимному управлению. Нижний уровень получает транслируемые операторским диалогом сигналы оператора d_1 , выходные сигналы алгоритмов определения уставок для локальных подсистем и фактические значения наблюдаемых переменных. Кроме того (в отличие от реальных систем), работа нижнего уровня системы управления подвержена вмешательствам со стороны инструктора m_1^s , имитирующего отказы автоматики. Вырабатываемые блоком

локального регулирования управления $u=U(d_1, z, y, m_1^S)$ передаются в модель технологического процесса, блок управления режимами ТП КС, диалоговое окно оператора и интерфейс инструктора. В простейшем случае ($z = m_1^S = 0$, d_1 – уставка оператора) задача управления сводится к стандартному регулированию по отклонению.

Верхний уровень системы получает из операторского диалога сигнальные и символьные потоки операторских вмешательств d_2 , d_2^S , управления u , наблюдаемые и вспомогательные переменные y , а также вмешательства инструктора m_2^S . Данный блок вырабатывает сигнальные выходы (уставки, задания и команды для нижнего уровня АСУ ТП КС) $z=Z(d_2, d_2^S, u, m_2^S)$ и символьные выходы (данные для оценки технико-экономических показателей, материальные и энергетические балансы, вспомогательные рекомендации).

Центральное место в разработанном имитаторе занимает алгоритм непосредственного обучения оператора за пультом дисплея. Несмотря на разнообразие формулировок учебно-тренировочных задач (УТЗ), весь процесс обучения на тренажере может быть представлен в виде универсального обучающего алгоритма. Под универсальностью алгоритма понимается как возможность его применения в процессе обучения и контроля, так и его использование для обучения любому из заданных классов задач.

Обучающий алгоритм реализован в виде модуля, управляющего диалогом между обучаемым и имитационной моделью. Обучение ведется в терминах проблемной области и включает следующие типы диалога: меню, вопросы, требующие ответа ДА/НЕТ, шаблоны. Выполнение каждого шага диалога сопровождается сообщением на экране дисплея.

Имитатор поддерживает работу в следующих режимах: обучения, подсказки, контроля знаний и исследовательский режим (рис. 2).

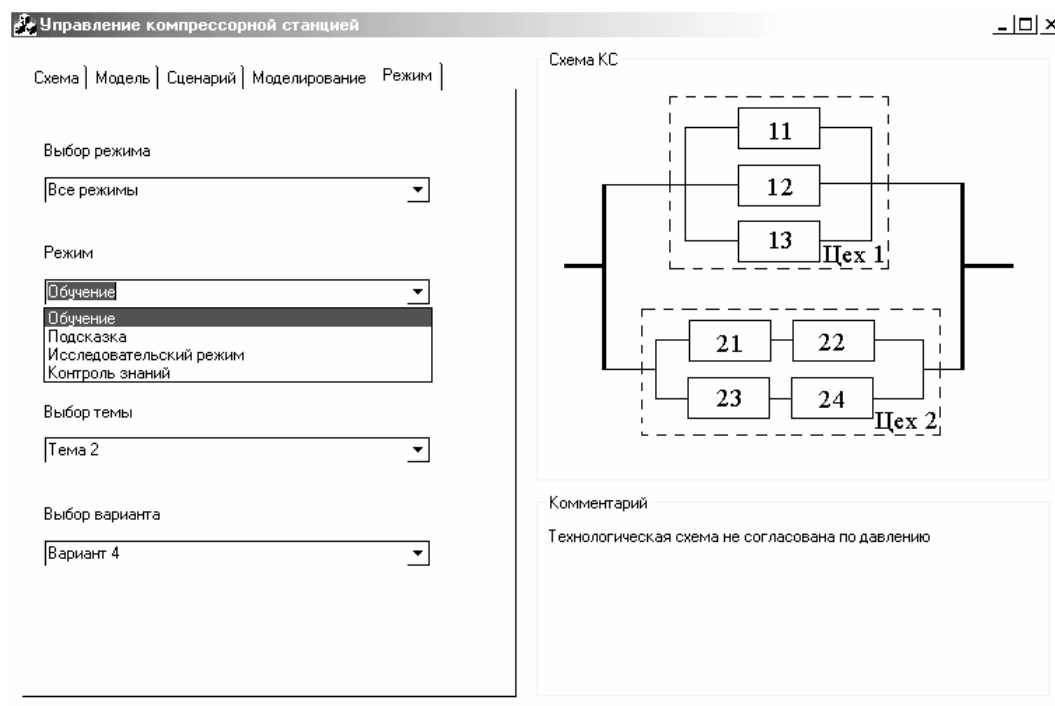


Рис. 2. Выбор режимов работы имитатора автоматизированного управления ТП КС

Режим «обучение» предполагает проведение учебного процесса с оператором, работающим самостоятельно за персональным дисплеем. В этом режиме для реализации конкретной УТЗ оператору предоставляется возможность обращаться за рекомендациями по управлению к модулю подсказок имитатора. Все действия оператора фиксируются в протоколе обучения. В режиме «контроль» подсказки по управлению запрещены.

Режим «исследование» предназначен для имитации в учебных условиях реально возможных ситуаций, позволяющих обучаемым обрабатывать действия, которые в этих ситуациях должны быть предусмотрены. Этот режим ориентирован также на приобретение обучаемыми навыков управления ГТС в условиях, приближенных к реальным, т.е. позволя-

ет работать с реальными объектами, их реальными характеристиками и параметрами. Отметим, что в этом режиме, также как и в режиме «обучение», можно обращаться за рекомендациями по управлению.

После того, как режим работы тренажера определен, осуществляется выбор УТЗ и формирование задания на обучение. Формирование задания, в зависимости от режима работы, заключается в выборе или задании конкретных исходных данных (например, параметров газового потока, технологических ограничений, граничных условий, начальных параметров управления и т.д.). По ходу выполнения задания фиксируется ряд параметров, характеризующих степень подготовленности обучаемого (например, количество нарушений технологических ограничений, количество попыток управления, время, затраченное на обдумывание и принятие управленческих решений и т.д.). Поэтому перед сеансом работы осуществляется обнуление счетчиков указанных параметров.

В имитаторе предусмотрены системные часы, которые фиксируют общее время выполнения задания и время, затраченное на обдумывание и принятие решений. Оператор, изменяя то или иное управляющее воздействие и выбирая цех КС или схему координации ГПА, может уже в исходной ситуации повлиять на режим работы моделируемого объекта. Для этого обучаемому оператору необходимо дать положительный ответ на вопрос об изменении управления.

Если текущий режим работы тренажера – не контроль знаний, то обучаемому предоставляется возможность воспользоваться для решения УТЗ рекомендациями по управлению, которые на основе сложившейся ситуации вырабатывает сам имитатор. Если обучаемый отказывается от подсказки, то управление вновь передается к блоку принятия решений оператором. В противном случае счетчик соответствующей подсказки увеличивается на единицу. На основе анализа ситуаций формируются и выдаются на экран монитора соответствующие рекомендации. Рассмотренный процесс будет повторяться до тех пор, пока задача не будет решена или (в случае контроля знаний) не будет исчерпано время на решение задачи. В том случае, если режим работы тренажера не является исследовательским, то на основе статистики, набранной в ходе выполнения УТЗ, формируется протокол обучения и выдается на экран монитора.

4. Выводы

Научная новизна полученных результатов состоит в совершенствовании методов оптимизации технологических режимов на верхнем уровне АСУ ТП КС с использованием нечетких моделей, позволяющих эффективно описывать процесс компримирования газа с учетом действующих возмущений и существующих ограничений. Эффективность реализации иерархического управления КС во многом зависит от качества решений, принимаемых оператором-технологом.

Практическая значимость заключается в создании модульного имитатора работы АСУ ТП КС, который позволяет повысить качество подготовки операторов-технологов, в диалоговом режиме осуществляющих принятие решений по управлению ТП КС.

Перспективным представляется развитие предложенного подхода с учетом состояния работоспособности оборудования компрессорных станций магистрального газопровода.

Список литературы: 1. *Исаков А.Т., Хохряков М.В., Фланчик Б.С. та ін.* Експлуатація і технічне обслуговування газорозподільних станцій магістральних газопроводів (довідник). Київ: "Росток", 2003. 411 с. 2. *Боженюк А.В., Шадрин В.В.* Нечеткая классификация ситуаций и принятие решений в системах магистрального транспорта // Известия ТРТУ - Таганрог: Изд-во ТРТУ, 2006. № 10 (65). С. 9-12. 3. *Альхайек Р., Удовенко С.Г.* Модифицированный метод построения инверсной нечеткой модели объекта цифрового управления // Системи управління, навігації та зв'язку. 2009. Вип. 4(12). С. 130–134.

Поступила в редколлегию 12.12.2011

Альхайек Ранем, аспирант кафедры электронных вычислительных машин ХНУРЭ. Научные интересы: нечеткая идентификация нелинейных систем, нейро-нечеткое управление. Адрес: Украина, 61166, Харьков, пр. Ленина, 14.

Удовенко Сергей Григорьевич, д-р техн. наук, профессор кафедры электронных вычислительных машин ХНУРЭ. Научные интересы: управление стохастическими процессами, методы вычислительного интеллекта. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 7021354.

Шамраев Анатолий Анатольевич, канд. техн. наук, доцент кафедры электронных вычислительных машин ХНУРЭ. Научные интересы: нейро-нечеткое управление, разработка и оптимизация микроконтроллерных систем. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 7021354.

ОСОБЕННОСТИ ПОСТРОЕНИЯ ПРИКЛАДНЫХ ПРОГРАММНЫХ СИСТЕМ НА ОСНОВЕ ОНТОЛОГИЧЕСКИХ БАЗ ЗНАНИЙ

Выделяются и анализируются этапы разработки приложения на основе онтологического хранилища знаний, проводится аналогия с процессом создания приложения, основанного на реляционной БД. Показывается принципиальная близость этих подходов и акцентируются значимые отличия.

Введение

В настоящее время онтологические системы могут быть применены для решения различных задач в сфере представления знаний в Интернет и создания интеллектуальных систем. Подобные системы, основанные на знаниях (Knowledge Based Systems), используют методы искусственного интеллекта для помощи в принятии решений, обучения и проч. Достоинством онтологических систем является универсальное представление информации, позволяющее применять одни и те же знания в разных предметных областях и разных задачах. Система, основанная на онтологическом принципе, легко расширяется. Такие системы позволяют проводить интеллектуальный анализ и обработку информации, используя всю выразительную мощь таких языков как SWRL и SPARQL.

Недостатком онтологических систем является большая вычислительная сложность при работе на больших объемах данных. Учитывая последние тенденции при создании серверов баз знаний и анализ, проведенный в [1], видно, что для построения производственных систем, основанных на онтологиях, возможно использовать такие RDF-хранилища как Virtuoso, которые делают применение онтологических хранилищ близкими по производительности и сложности к использованию традиционных СУБД.

Целью исследования является разработка метода создания производственных программных продуктов на основе онтологических хранилищ знаний. Такие приложения потенциально имеют следующие преимущества:

- масштабируемость относительно количества источников данных и размера онтологии;
- расширяемость (минимизация затрат, связанных с адаптацией приложения при расширении схемы базы знаний);
- облегчение интеграции онтобазированных приложений в корпоративные информационные системы в связи с использованием универсальной стандартизированной модели представления знаний, не зависящей от конкретного приложения.

Для достижения поставленной цели были сформулированы следующие задачи:

- исследование возможностей и методик внедрения онтологических хранилищ знаний в высоконагруженные программные продукты, формирование универсального метода;
- апробация предложенного метода на тестовом приложении.

1. Анализ предметной области

Шаблон MVC стал де-факто стандартом при проектировании приложений, основанных на данных [2]. В нем Model – абстракция, соответствующая графу данных предметной области, инкапсулирует бизнес-логику их обработки, а также в некоторых интерпретациях – логику сохранения/восстановления состояния модели между перезапусками системы (рис. 1).

Часто среды разработки предоставляют целые фреймворки (ORM) для автоматической генерации кода, реализующего логику сохранения/восстановления состояния модели на основе традиционных источников данных, например, таких как реляционные базы данных. Так (среди прочего), для платформы .NET разработан EntityFramework; для Java – Hibernate, JPA; для Python – Django.

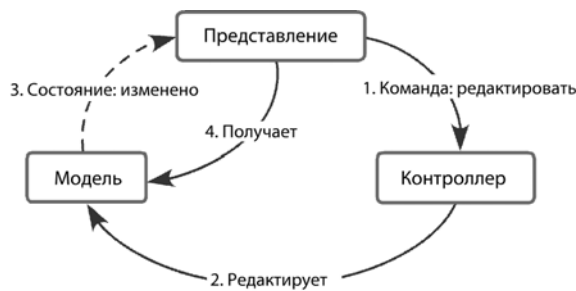


Рис. 1. Шаблон проектирования MVC

К сожалению, для инновационных в настоящее время источников данных (таких, как RDF-хранилища) выбор ORM-библиотек и их производительность значительно отстает. Например, известно решение на основе Tinkerpop, заточенное под Neo4J [3] или jenabean [4]. Но поскольку результат сравнения RDF-хранилищ [1] позволил выделить явного лидера в этой области Openlink Virtuoso, то практический интерес представляют способы обращения к графам, поддерживаемые именно этим сервером.

Доступ к RDF-хранилищу Virtuoso осуществляется с помощью языка запросов SPARQL. Разработчики Virtuoso предоставляют несколько вариантов подачи SPARQL запросов к серверу (рис.2): 1) через HTTP REST Endpoint; 2) через традиционные API, такие как ODBC, JDBC, .NET или OLE/DB (Virtuoso имеет возможность исполнять SPARQL запросы, являющиеся составной частью SQL); 3) через специализированные RDF Data-провайдеры Virtuoso посредством API распространенных RDF-фреймворков (Jena, Sesame, Redland, Linq2Rdf, dotnetRDF).

Как первое, так и второе – довольно низкоуровневые по сравнению с ORM подходы, подразумевающие большой объем рутинной работы для повышения уровня абстракции (в первом случае добавляется еще и низкоуровневая работа с протоколом HTTP). В третьем случае мы имеем дело с мощным API, по своим возможностям значительно перекрывающим классические задачи ORM. Хотя при оптимальной реализации первый подход может показать значительный выигрыш в производительности, по сравнению с последним, третий вариант по своей сути более близок к традиционным ORM-библиотекам. Этот вариант и был выбран базовым.

В качестве тестового приложения выбрана задача по созданию универсального модуля для проведения опросов. Языком программирования выступает Java, а в качестве среды Eclipse SDK. RDF-хранилище создано на Openlink Virtuoso. Доступ к онтологии производится через java-фреймворк Jena. Для доступа к RDF был применен наиболее высокоуровневый вариант через специализированный RDF Data-провайдер для Jena. Интерфейс приложения создается при помощи GWT (Google Web Toolkit).

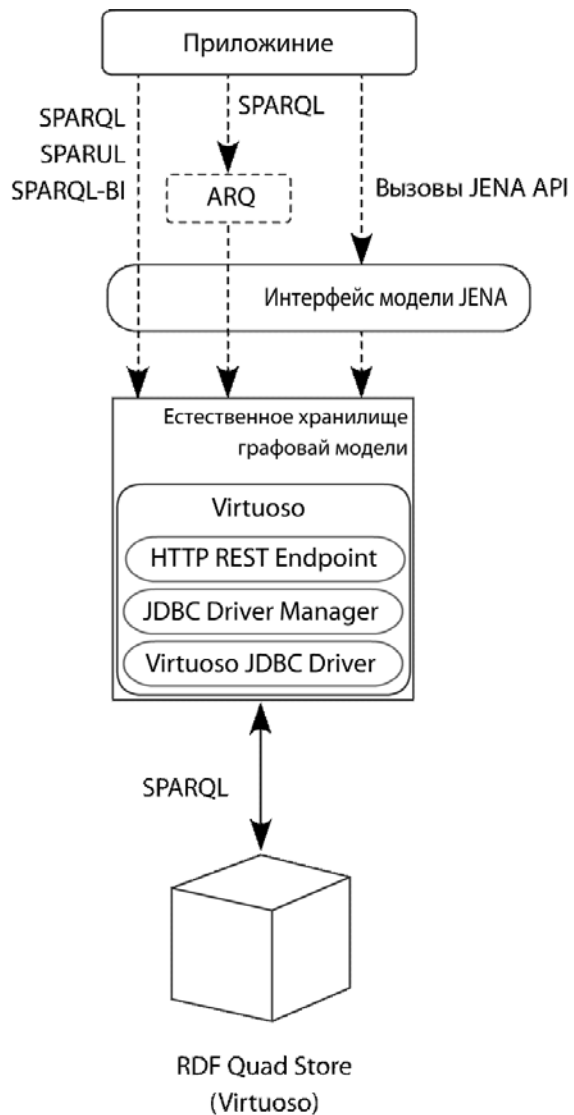


Рис. 2. Интерфейсы доступа к RDF-хранилищу Virtuoso

2. Метод создания программных продуктов на основе онтологических баз знаний

Все имеющиеся вопросы вместе с вариантами ответов предполагается хранить в виде OWL онтологии, сюда же будут заноситься ответы пользователей. Посредством SPARQL запросов вопросы с вариантами ответов будут выбираться из хранилища для отображения пользователю. Таким же образом результаты опроса будут возвращаться обратно в хранилище. Далее перечислены этапы, которые разработчик должен выполнить для реализации приложения, основанного на онтологиях.

Этап 1. Создание схемы онтологии. С помощью Protege была смоделирована предметная область. Выделен класс «Question», соответствующий одному вопросу. Класс «ResponseOption» представляет вариант ответа на конкретный вопрос; соответствие варианта ответа и вопроса задает слот «questionLink», ссылающийся на экземпляр класса «Question».

Класс «User» представляет в онтологии сущность «пользователь», конкретные его экземпляры (instances) – это пользователи, принимавшие участие в опросе. Каждый экземпляр класса «User» кроме данных о пользователе содержит и ответы на вопросы в виде ссылок на выбранный вариант ответа для каждого вопроса. Готовая онтология сохранена в формате OWL (рис.3).

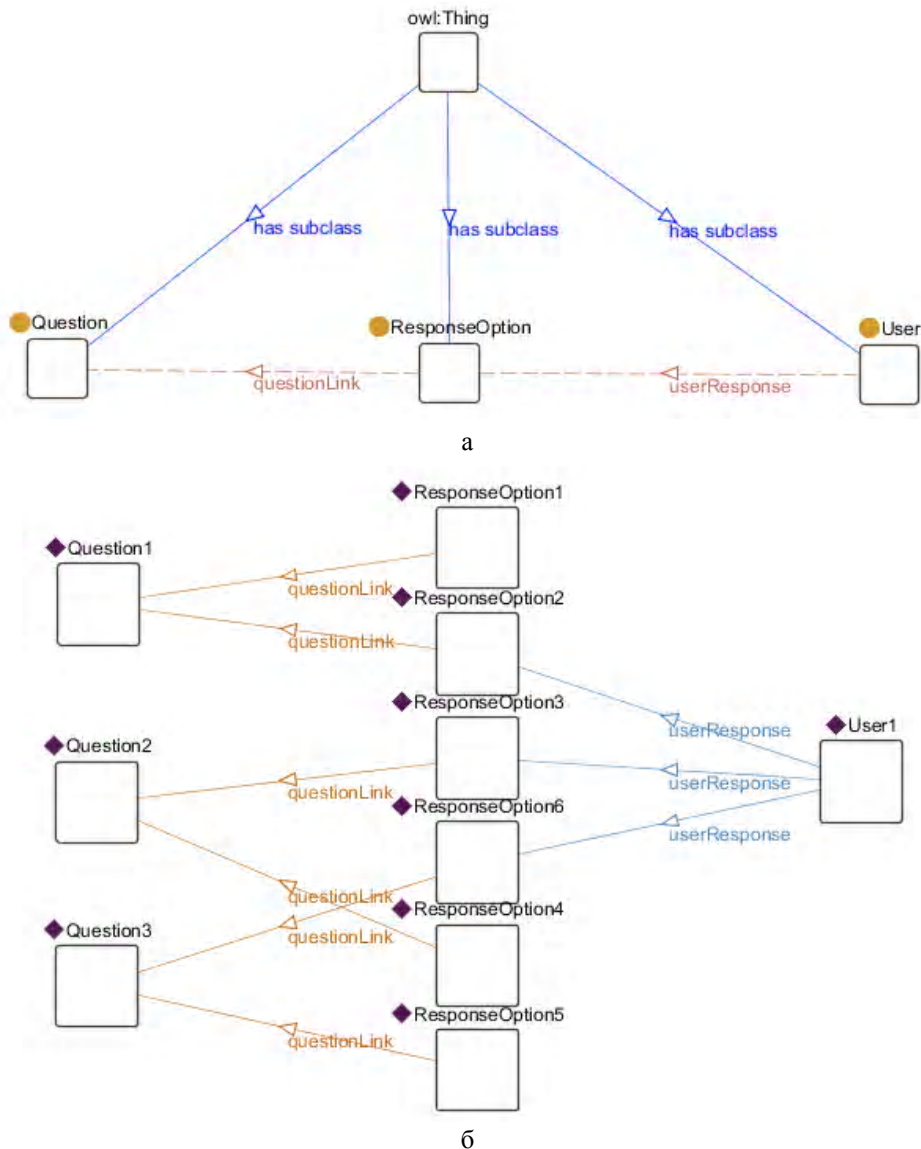


Рис. 3. Граф из редактора онтологий: а – только TBox; б – только ABox

Этап 2. Интеграция онтологии в Virtuoso. После загрузки и установки Virtuoso необходимо импортировать созданную онтологию в хранилище. Это осуществляется через консоль управления, которая при стандартных настройках после запуска Virtuoso Service Manager доступна по адресу <http://localhost:8890/>.

Для загрузки онтологии следует воспользоваться WebDAV Browser, где для любого пользователя возможно создать папку WebDAV-типа и в ней разместить данные в формате RDF или OWL, а затем загрузить их в хранилище триплетов. Необходимо помнить, что в Virtuoso каждая запись онтологии имеет дополнительный атрибут – идентификатор графа, которому принадлежит триплет. Поэтому на самом деле хранилище Virtuoso представляет собой quad store. Это позволяет содержать в Virtuoso множество отдельных онтологий-задач.

Далее нужно проверить правильность загрузки онтологии следующим образом: необходимо перейти по адресу <http://localhost:8890/sparql>, в качестве URI графа по умолчанию указывается путь к онтологии. Для стандартного пользователя путь будет выглядеть так: http://localhost:8890/DAV/home/dba/rdf_sink/, где dba – это имя пользователя. В поле ввода SPARQL запроса можно вписать, например: “SELECT DISTINCT ?p WHERE {?s ?p ?o} LIMIT 10”, в результате выполнения данного запроса должен отобразиться список предикатов онтологии с ограничением не более десяти.

Этап 3. Разработка архитектуры приложения. Внутренняя структура приложения, использующего онтологическое хранилище, ничем не ограничена и может быть построена согласно любой методологии. Единственное специфическое требование – это необходимость создания прослойки доступа к знаниям. Для взаимодействия с сервером, на котором расположен репозиторий онтологии, потребуются библиотеки фреймворка Jena, которые можно скачать на сайте OpenLink Software [5]. Общая модель архитектуры приложения приведена на рис. 4.

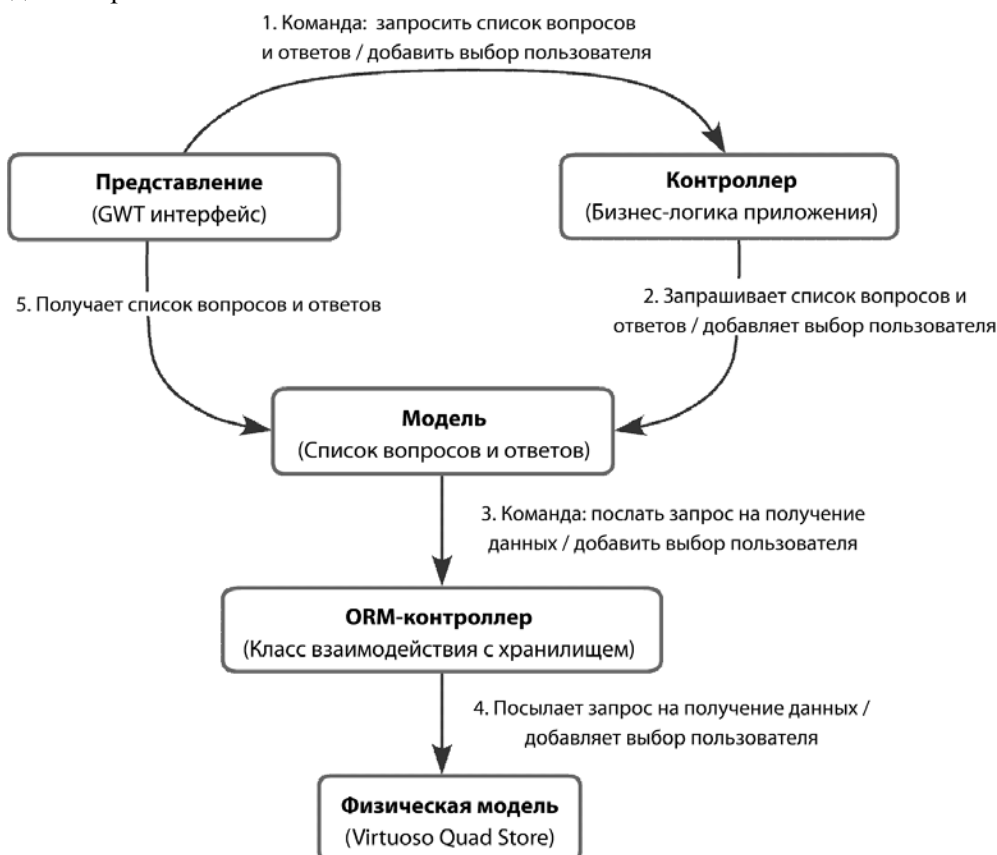


Рис. 4. Схема разработанного приложения в терминах MVC

Для упрощения работы с данными онтологии были созданы серверный класс «Question» и «ResponseOption» - элементы модели, отображающиеся на аналогичные классы в онтологии, в которых в виде объектной модели уровня приложения хранятся вопросы с вариантами

ми ответов и ссылки на эти данные в онтологии. Таким образом, упрощается процедура отображения вопросов и вариантов ответов на веб-форме, а также отправка результатов опроса на сервер и записи их в онтологию.

Как уже было сказано, для связи с репозиторием ORM-контроллер применяет фреймворк Jena. Он инкапсулирует использование библиотек Jena, производит установку связи с сервером и выполнение запросов для получения или отправки данных на сервер.

Ниже приведен пример запроса на получение списка вопросов из репозитория:

```
String queryStringQuestions =  
    "PREFIX table: <http://www.owl-ontologies.com/interview.owl#>" +  
    "select * from <http://localhost:8890/DAV/home/dba/rdf_sink/>" +  
    "where {?s table:question ?o}";
```

API отправки запроса на сервер и получение результатов аналогично стандартным классам для выполнения SQL-запросов:

```
queryExec = new QueryEngineHTTP(endpoint, queryStringQuestions);  
resultQuestions = queryExec.execSelect();
```

Здесь *endpoint* – аналог сущности Connection для работы с СУБД.

Аналогично производится запрос на добавление триплетов в онтологию:

```
String queryString =  
    "PREFIX table: <http://www.owl-ontologies.com/interview.owl#>" +  
    "INSERT INTO GRAPH <http://localhost:8890/DAV/home/dba/rdf_sink/> "+  
    "{table:User1 table:userResponse ?o." +  
    " table:User1 table:userResponse table:ResponseOption6.}";
```

Здесь под параметром ?o подразумевается объект, который будет добавлен в хранилище как свойство «вариант ответа» (table:userResponse) определенного пользователя (table:User1). Строка с параметром определяет шаблон, по которому сначала находится «место» в онтологии, куда необходимо добавить объект. В нашем случае пользователю добавляем вариант ответа.

Этап 4. Создание интерфейса системы опросов при помощи GWT. Технология создания пользовательского интерфейса, по сути, значения не имеет. Поскольку созданное тестовое приложение является веб-приложением, бизнес-логика приложения заключена в сервлет. Для создания интерфейса был выбран фреймворк Google Web Toolkit. На веб-форме располагаются поля для отображения вопроса и выпадающие списки с вариантами ответа. Элементы интерфейса формируются динамически в зависимости от количества вопросов в репозитории, количество вариантов ответа также может быть любым. На рис. 5 отображена веб-форма приложения для ввода ответов.

The image shows a web form titled "Qualification survey" with a light gray header. Below the header, there are seven numbered questions, each followed by a dropdown menu. The questions and their corresponding dropdown values are:

- 1) **Lecture material corresponds to the current state of knowledge in this subject area** (Good)
- 2) **Effective use of modern learning technologies (multimedia, telecommunications, etc.)** (Unsatisfactorily)
- 3) **Material is explained clearly, uses examples** (Satisfactorily)
- 4) **Involves students in a discussion** (No)
- 5) **Able to generate interest of the audience to the subject** (Unsatisfactorily)
- 6) **Change the tasks according to the degree of preparedness of students and their abilities** (Satisfactorily)
- 7) **Evaluates the knowledges objectively** (Do not know)

At the bottom of the form, there is a "Submit" button.

Рис. 5. Интерфейс системы опросов

Выводы. Практическая значимость предложенного в статье метода состоит в том, что он позволяет повысить уровень интеллектуализации высоконагруженных производственных приложений путем использования онтологических хранилищ знаний. Дальнейшие исследования должны быть направлены на оценку производительности таких приложений и поиск путей её повышения.

Список литературы: **1.** *Europeana* RDF Store Report [Текст]: The results of qualitative and quantitative study of existing RDF stores in the context of Europeana / Multimedia Information Systems Research Group, CS Faculty, University of Vienna; В. Haslhofer. Vienna, 2011. 30 p. **2.** Фаулер М. Шаблоны корпоративных приложений [Текст]/Фаулер М. М.: «Вильямс», 2009. С. 544. **3.** *RDF* data in Neo4J – the Tinkerpop story / Портал DZone для разработчиков. Режим доступа: [www / URL: http://java.dzone.com/news/rdf-data-neo4j-tinkerpop-story](http://java.dzone.com/news/rdf-data-neo4j-tinkerpop-story). 20.04.2012. Загл. с экрана. **4.** *Binding* Java Objects to RDF / Портал группы-участника W3C SemanticWeb. Режим доступа: [www / URL: http://semanticweb.com/binding-java-objects-to-rdf_b10682?red=su](http://semanticweb.com/binding-java-objects-to-rdf_b10682?red=su). 20.12.2011. Загл. с экрана. **5.** *OpenLink* Virtuoso Open-Source Edition: Downloads/ Официальный сайт Openlink Software. Режим доступа: [www / URL: http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VOSDownload#Jena%20Provider](http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VOSDownload#Jena%20Provider). 20.12.2011. Загл. с экрана.

Поступила в редколлегию 16.11.2011

Шевченко Елена Леонидовна, канд. техн. наук, доцент кафедры ПИ ХНУРЭ. Научные интересы: онтологический инжиниринг, формальные методы в разработке ПО. Адрес: Украина, 61000, Харьков, ул. Космическая, 11, кв. 37, тел. 0505748372.

Шевченко Александр Юрьевич, канд. техн. наук, доцент кафедры ИИ ХНУРЭ. Научные интересы: онтологический инжиниринг, распределенные интеллектуальные системы. Адрес: Украина, 61000, Харьков, ул. Космическая, 11, кв. 37, тел. 0506192638.

Богдан Александр Николаевич, студент, гр. КН-08-4, ХНУРЭ. Научные интересы: онтологический инжиниринг. Адрес: Украина, 61000, Харьков, ул. Блюхера, 42, кв. 96, тел. 0671201964.

МЕТОДЫ ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ НА ОСНОВЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ С ЭЛЕМЕНТАМИ ВРЕМЕННОЙ ЗАДЕРЖКИ

Рассматривается задача обучения специализированных нейронных сетей, предназначенных для прогнозирования рядов показателей, описывающих поведение сложных многосвязных систем. Описывается метод обучения искусственных нейронов многослойных нейросетей с прямой передачей информации, обладающий робастными свойствами в условиях возмущений с неизвестным распределением. Обобщаются методы обучения прогнозирующих рекуррентных нейросетей на динамических нейронах-фильтрах с конечной и бесконечной импульсной характеристиками, отличающиеся высоким быстродействием и повышенными сглаживающими свойствами.

1. Введение

Проблема обработки больших объемов разнообразной, подчас неполной и противоречивой диагностической информации выходит на передний план в различных областях, таких как прогнозирование и моделирование, управление, оптимизация, анализ данных, прикладная статистика [1-3]. Решение этих задач связывают с новыми информационными технологиями [4,5]. Так как многие системы реального времени демонстрируют комплексные нелинейные характеристики, к ним неприменимы способы обработки, характерные при использовании теории линейных систем [6-9]. Нейронные сети - это один из самых универсальных методов для решения задач прогнозирования в ситуациях, когда в экспериментальных данных отсутствует часть информации, а имеющаяся информация зашумлена. Все сказанное выше определяет актуальность исследования нейросетевых технологий обработки сигналов.

Развитие в области теории и приложений нейронных сетей идет в самых разных направлениях – это поиск новых нелинейных элементов, которые могли бы реализовывать слож-

ное поведение нейронов, новые архитектуры нейронных сетей, поиск новых областей приложения нейротехнологий. Значительное место в данных исследованиях традиционно занимает математическое моделирование. В связи с этим возникает необходимость проанализировать возможности моделирования структуры нейросетей, рекуррентные методы настройки этих нейросетевых моделей, а также свойства сходимости изучаемых процедур обучения, их преимущества и недостатки.

Цель исследования - обобщение методов обучения искусственных нейронных сетей с элементами временной задержки, предназначенных для прогнозирования нестационарных рядов показателей, описывающих поведение многосвязных систем.

Задачи исследования:

– анализ методов обучения искусственных нейронных сетей и обоснование возможности применения в задачах моделирования и прогнозирования сложных многосвязных систем;

– обобщение методов обучения искусственных нейросетей с элементами временной задержки.

2. Обучение искусственной нейронной сети

Типичная искусственная нейросеть состоит из множества простых вычислительных элементов, размещаемых послойно и использующихся параллельно. Весовые коэффициенты, которые определяют силу связи между элементами, подстраиваются во время работы сети, тем самым обеспечивая хорошие эксплуатационные показатели. Нейросети образуют классы в соответствии с характеристиками элементов, архитектурой и правилами обучения. Так, нейросети, в которых входной сигнал распространяется через слои сети от входа к выходу, известны как аппроксимирующие нейросети прямого распространения. Эта разновидность сети способна к обучению комплексным отображениям входа-выхода.

Многослойная нейросеть является универсальным аппроксиматором, однако не существует конструктивных оценок, позволяющих определить точное количество нейронов в скрытом слое. Многослойная нейросеть может обладать произвольными начальными условиями, при этом необходимо начать процесс обучения. Таким образом, нейросеть обладает аппроксимирующими свойствами в том случае, если её весовые коэффициенты обучены с помощью того или иного правила настройки, минимизирующего принятый критерий обучения. Если есть заданный критерий обучения и есть соответствующая процедура, которая минимизирует этот критерий, то нейросеть будет обладать требуемыми аппроксимирующими свойствами.

Обучение в общем случае – многоэкстремальная задача оптимизации, для решения которой используются:

– процедуры локальной оптимизации с вычислением частных производных первого порядка (градиентная процедура; методы одномерной и двумерной оптимизации целевой функции в направлении антиградиента; метод сопряженных градиентов; методы, учитывающие направление антиградиента на нескольких шагах процедуры);

– процедуры локальной оптимизации с вычислением частных производных первого и второго порядка (метод Ньютона; методы оптимизации с разреженными матрицами Гессе; квазиньютоновские методы; метод Гаусса-Ньютона; метод Левенберга-Марквардта);

– стохастические процедуры оптимизации (поиск в случайном направлении; метод Монте-Карло (численный метод статистических испытаний); имитация отжига);

– процедуры глобальной оптимизации (перебор значений переменных, от которых зависит целевая функция).

Класс процедур обучения, известных как процедуры прогнозирования ошибки, может быть получен для многослойной нейросети при использовании методов нелинейной настройки систем.

Оценка параметров в нелинейных моделях основана на нелинейных методах оптимизации [3].

Процедуры, которые используются для обучения нейросетей – классические процедуры, такие как процедура обратного распространения ошибки во времени, обладают низкой скоростью сходимости. В связи с этим необходимо провести исследования, направленные на улучшение скорости сходимости. В ходе этих исследований нужно оптимизировать

процессы обучения нейросети, чтобы проводить обучение сети как в реальном времени, так и на коротких выборках. В силу того, что выборка реальных данных имеет, как правило, ограниченный объем, необходимо обеспечить максимальную скорость сходимости на короткой выборке. Так, можно ввести в структуру нейросети фактор времени, организовав на входе цепочку элементов чистой задержки. Однако такой подход ведет к тому, что количество синаптических весов резко возрастает, что неприемлемо для работы в условиях ограниченной выборки, т.е. в условиях, когда выборка ограничена, невозможно решить задачу прогнозирования с помощью классических нейросетей, у которых на входе – цепочка элементов чистой задержки. В связи с тем, что классические архитектуры нейронных сетей с элементами чистого запаздывания в нулевом слое не отвечают поставленным требованиям, необходимо обратиться к другим конструкциям, которые специальным образом ориентированы на работу с временными рядами. Именно к таким сетям и относятся нейросети, построенные на так называемых динамических нейронах, представляющих собой, по сути, нелинейные цифровые фильтры. Хотя подобные сети были разработаны достаточно давно, однако не поднимался вопрос оптимизации процессов обучения нейросетей во времени. Этот вопрос не является проблемным, если обучающая выборка может быть сколь угодно велика. Но для работы с реальными данными эта выборка всегда ограничена.

3. Обучение прогнозирующей нейросети с прямой передачей информации на динамических нейронах

Диапазон задач, решаемых с помощью нейротехнологий, в настоящее время достаточно велик, это обстоятельство не позволяет создавать универсальные нейросети, которые бы не были сформированы под определенную задачу. Структура нейросети выбирается в соответствии с особенностями и степенью сложности решаемых задач, таких как, например, обработка сигналов различной природы в технических, медицинских, экономических и прочих приложениях.

Для решения некоторых классов задач существуют оптимальные нейросетевые конфигурации. Если задача не может быть сведена ни к одному из известных классов задач, тогда решается проблема синтеза новой конфигурации.

Можно отметить, что чем сложнее архитектура нейросети, тем больше задач она способна решить. Процесс работы нейросети зависит также от настройки синаптических связей, поэтому необходимо найти оптимальные значения весовых коэффициентов, т.е. провести обучение. Кроме параметра качества подбора весов важную роль играет также время обучения. Эти два параметра могут быть связаны обратной зависимостью, поэтому на практике обычно требуется достичь определенного компромисса.

Рассмотрим случай, когда прогнозирование необходимо проводить в реальном времени по мере поступления новых данных. Используем нейросети, которые по прошлым наблюдениям позволяют восстанавливать нелинейное динамическое отображение вида:

$$x(k) = F(x(k-1), x(k-2), \dots, x(k-n_A)) + e(k) = \hat{x}(k) + e(k),$$

где $\hat{x}(k)$ – оценка (прогноз) значения $x(k)$, полученная на выходе искусственной нейронной сети (ИНС), представляющей в данном случае нелинейную авторегрессионную модель; $e(k)$ – ошибка прогнозирования.

В качестве основы для построения нелинейных авторегрессионных моделей обычно используются многослойные нейросети с прямой передачей информации, входной (нулевой) слой которых образован линиями элементов чистой задержки с отводами.

Стандартный статический нейрон реализует нелинейное отображение

$$x_j^{[l+1]} = \psi_j^{[l]}(u_j^{[l+1]}) = \psi_j^{[l]} \left(\sum_{i=0}^{n^{[l]}} u_{ji}^{[l+1]} \right) = \psi_j^{[l]} \left(\sum_{i=0}^{n^{[l]}} w_{ji}^{[l]} x_i^{[l]} \right),$$

синаптические веса $w_{ji}^{[l]}$ которого подлежат уточнению в процессе обучения нейронной сети.

Нелинейное отображение, реализуемое динамическим нейроном, можно записать в виде:

$$x_j^{[l+1]}(k) = \psi_j^{[l]}(u_j^{[l+1]}(k)) = \psi_j^{[l]} \left(\sum_{i=0}^{n_j^{[l]}} u_{ji}^{[l+1]}(k) \right) = \psi_j^{[l]} \left(\sum_{i=0}^{n_j^{[l]}} W_{ji}^{[l]} X_i^{[l]}(k) \right).$$

Общим недостатком прогнозирующих сетей, построенных на статических нейронах, является чрезвычайно большое количество настраиваемых весов и низкая скорость обучения, что, естественно, вызывает серьезные проблемы, особенно при работе в реальном времени.

Хотя количество параметров, которые содержит динамический нейрон, превышает количество синаптических весов стандартного нейрона, сеть, построенная из таких узлов, содержит намного меньше параметров, чем стандартная архитектура на статических нейронах с линиями задержки на входе.

Для обучения нейросетей на динамических нейронах в [10] была введена градиентная процедура, получившая название обратного распространения ошибок во времени:

1) Одношаговый критерий обучения

$$J(k) = \frac{1}{2} \|e(k)\|^2 = \frac{1}{2} \|d(k) - N(X^{[1]}, W^{[1]}, \dots, W^{[L]})\|^2 = \frac{1}{2} \|d(k) - \hat{x}(k)\|^2,$$

где $d(k)$ – обучающий сигнал, в качестве которого в задачах прогнозирования принимается текущее значение $x(k)$.

2) Процедура минимизации критерия обучения

$$W_{ji}^{[l]}(k+1) = W_{ji}^{[l]}(k) - \gamma^{[l]}(k) \frac{\partial J(k)}{\partial u_j^{[l+1]}(k)} \nabla_{W_{ji}^{[l]}} u_j^{[l+1]}(k),$$

здесь $\gamma^{[l]}(k)$ параметр, определяющий скорость сходимости обучения.

3) Локальная ошибка обучения

$$\frac{\partial J(k)}{\partial u_j^{[l+1]}(k)} = \delta_j^{[l]}(k).$$

4) Процедура настройки нейронов выходного слоя

$$\begin{aligned} W_{ji}^{[L]}(k+1) &= W_{ji}^{[L]}(k) + \gamma^{[L]}(k) e_j(k) \psi_j^{[L]'}(u_j^{[L+1]}(k)) X_i^{[L]}(k) = \\ &= W_{ji}^{[L]}(k) + \gamma^{[L]}(k) e_j(k) J_i^{[L]}(k). \end{aligned}$$

5) Локальная ошибка обучения для скрытых слоев сети

$$\begin{aligned} \delta_j^{[l]}(k) &= \frac{\partial J(k)}{\partial u_j^{[l+1]}(k)} = \sum_{q=1}^{n_{l+1}} \sum_{t=k}^{n_A^{[l+1]}+k} \delta_q^{[l+1]}(t) \frac{\partial u_q^{[l+1]}(t)}{\partial u_j^{[l+1]}(k)} = \\ &= \psi_j^{[l+1]'}(u_j^{[l+1]}(k)) \sum_{q=1}^{n_{l+1}} \sum_{t=k}^{n_A^{[l+1]}+k} \delta_q^{[l+1]}(t) \frac{\partial u_{jq}^{[l+1]}(t)}{\partial x_j^{[l]}(k)}. \end{aligned}$$

6) Адаптивная процедура

$$W_{ji}^{[l]}(k+1) = W_{ji}^{[l]}(k) + \frac{e_j^{[l]}(k) J_i^{[l]}(k)}{\beta + \|J_i^{[l]}(k)\|^2}, \quad 1 \leq l \leq L,$$

где $\beta > 0$ – регуляризующий параметр;

Процедура настройки процесса обучения в скрытых слоях оптимизирована по скорости.

7) Модифицированный метод обучения

$$\begin{cases} W_{ji}^{[1]}(k+1) = W_{ji}^{[1]}(k) + \frac{e_j^{[1]}(k) J_i^{[1]}(k)}{\beta_i^{[1]}(k)}, \\ \beta_i^{[1]}(k+1) = \alpha \beta_i^{[1]}(k) + \|J_i^{[1]}(k)\|^2, \quad 0 \leq \alpha \leq 1, \end{cases}$$

где α – параметр сглаживания. Метод модифицирован в целях придания сглаживающих свойств, необходимых при обработке “зашумленных” сигналов [7].

Следует отметить, что прогнозирующие нейронные сети на базе динамических нейронов могут применяться при решении задач, связанных с моделированием сложных нелинейных динамических систем, в частности, в задачах прогнозирования нестационарных временных рядов.

Предложенный метод обучения нейросетей на динамических нейронах-фильтрах [11] характеризуется повышенными сглаживающими свойствами и высоким быстродействием, что позволяет работать в режиме реального времени.

В основе большинства рекуррентных методов обучения лежит гипотеза о нормальном распределении помех, что привело к использованию методов, основанных на критерии наименьших квадратов, которые на практике оказываются чрезвычайно чувствительными к отклонениям фактического закона распределения от нормального.

При работе в реальном времени решение задач робастной настройки, основанных на минимизации критериев, отличных от квадратичного, затруднено в силу численной громоздкости. Преодоление указанных трудностей видится в использовании нейросетевых методов обработки информации. Вводя в рассмотрение теоретическую текущую ошибку настройки, вектор неизвестных коэффициентов, подлежащих определению, вектор независимых входных переменных, включающих в частном случае и параметр времени, и робастный критерий оценивания Р. Вэлша, получили метод обучения [5], являющийся многошаговой процедурой. Так как использование этого метода в режиме реального времени затруднено в связи с необходимостью обработки на каждом такте всей выборки наблюдений, был предложен [12] метод обучения на основе одношаговой процедуры.

4. Метод обучения локально-рекуррентной нейронной сети на динамических нейронах-фильтрах

Особенность нелинейной авторегрессионной модели состоит в том, что синапсы ее нейронов по сути являются адаптивными динамическими фильтрами. Вместе с тем, остается открытым вопрос о выборе порядка фильтра, определяющего динамические свойства синтезируемой прогнозирующей сети. В связи с этим были предложены [13] локально-рекуррентные нейронные сети, синапсы которых являются адаптивными рекуррентными фильтрами с бесконечной импульсной характеристикой.

Для обучения этих сетей предложена группа градиентных процедур, чья скорость сходимости оказалась явно недостаточной при обработке больших массивов информации.

Наиболее эффективной процедурой настройки локально-рекуррентных нейронных сетей на сегодня является процедура рекуррентного обратного распространения ошибок, обеспечивающая минимизацию в пакетном режиме (обучение по эпохам) критерия обучения:

$$J^k = \sum_{k=1}^N J(k) = \frac{1}{2} \sum_{k=1}^N \|e(k)\|^2 = \frac{1}{2},$$

где $e(k) = (e_1(k), e_2(k), \dots, e_{n[L+1]}(k))^T$; $e_j(k) = d_j(k) - x_j^{[L+1]}(k) = y_j(k) - \hat{y}_j(k)$; $d_j(k)$ – внешний обучающий сигнал, в качестве которого в задаче настройки используется значение выхода реального объекта.

Базовым элементом локально-рекуррентной нейронной сети является динамический нейрон-фильтр, синапсы которого представляют собой адаптивные фильтры с бесконечной импульсной характеристикой.

Отображение, реализуемое синапсами и динамическим нейроном l-го слоя сети, в целом может быть записано в виде

$$u_{ji}^{[l+1]}(k) = \sum_{p=1}^{n_W^{[l]}} w_{jip}^{[l]} x_i^{[l]}(k-p) + \sum_{p=1}^{n_V^{[l]}} v_{jip}^{[l]} u_i^{[l]}(k-p) = \frac{W_{ji}^{[l]}(z^{-1})}{V_{ji}^{[l]}(z^{-1})} x_i^{[l]}(k) = S_{ji}^{[l]T} X_i^{[l]}(k);$$

$$u_j^{[l+1]}(k) = \sum_{i=1}^{n^{[l]}} u_{ji}^{[l+1]}(k);$$

$$x_j^{[l+1]}(k) = \psi_j^{[l]}(u_j^{[l+1]}(k)),$$

$$\text{где } W_{ji}^{[l]}(z^{-1}) = \sum_{p=0}^{n_W^{[l]}} w_{jip}^{[l]} z^{-p}; \quad V_{ji}^{[l]}(z^{-1}) = \sum_{p=0}^{n_V^{[l]}} v_{jip}^{[l]} z^{-p};$$

$$S_{ji}^{[l]} = \left(w_{ji0}^{[l]}, w_{ji1}^{[l]}, \dots, w_{jinn_W^{[l]}}^{[l]}, v_{ji1}^{[l]}, v_{ji2}^{[l]}, \dots, v_{jinn_V^{[l]}}^{[l]} \right)^T;$$

$$X_i^{[l]}(k) = \left(x_i^{[l]}(k), x_i^{[l]}(k-1), \dots, x_i^{[l]}(k-n_W^{[l]}), u_{ji}^{[l]}(k-1), \dots, u_{ji}^{[l]}(k-n_V^{[l]}) \right)^T.$$

Динамический нейрон-фильтр является объединением линейного адаптивного рекуррентного фильтра со стандартным нелинейным статическим нейроном.

Адаптивный нейрон-фильтр фактически представляет собой настраиваемую модель авторегрессии - скользящего среднего вида

$$V(k, z^{-1})u(k) = W(k, z^{-1})x(k).$$

Здесь z^{-1} – оператор сдвига назад.

Для обучения нейронной сети используем процедуру послойной оптимизации критерия обучения.

Метод обучения многослойной локально-рекуррентной сети может быть реализован в виде такой последовательности шагов:

- 1) вычисление ошибки обратного распространения $e_j^{[l+1]}(k)$ для всех $k = 1, 2, \dots, N$:

$$e_j^{[l+1]}(k) = \begin{cases} e_j(k), l=L, \\ \sum_{q=1}^{n^{[l+1]}} \sum_{r=1}^{N-k} \delta_q^{[l+1]}(k+r) \frac{\partial u_{jq}^{[l+1]}(k+r)}{\partial x_j^{[l+1]}(k)}, 1 < l < L; \end{cases}$$

- 2) вычисление локальной ошибки $\delta_j^{[l+1]}(k)$ для всех $k = 1, 2, \dots, N$:

$$\delta_j^{[l+1]}(k) = e_j^{[l+1]}(k) \psi_j^{[l]'}(u_j^{[l+1]}(k));$$

- 3) вычисление производных сигнала внутренней активации $u_i^{[l+1]}(k)$ по настраиваемым синаптическим весам с помощью соотношений:

$$\frac{\partial u_j^{[l+1]}(k)}{\partial w_{jip}^{[l+1]}} = x_i^{[l]}(k-p) + \sum_{t=1}^{n_V^{[l]}} v_{jit}^{[l]} \frac{\partial u_j^{[l+1]}(k-t)}{\partial w_{jip}^{[l]}},$$

$$\frac{\partial u_j^{[l+1]}(k)}{\partial v_{jip}^{[l+1]}} = u_i^{[l]}(k-p) + \sum_{t=1}^{n_V^{[l]}} v_{jit}^{[l]} \frac{\partial u_j^{[l+1]}(k-t)}{\partial v_{jip}^{[l]}};$$

- 4) вычисление оптимального значения параметра шага [8] и уточнение вектора параметров каждого синапса $S_{ji}^{[l+1]}(k)$ с помощью соотношения:

$$\begin{cases} S_{ji}^{[l]}(k+1) = S_{ji}^{[l]}(k) + \frac{e_j^{[l+1]}(k) E_i^{[l]}(k)}{\alpha_i^{[l]}(k)}, 1 \leq l \leq L, \\ \alpha_i^{[l]}(k+1) = \beta \alpha_i^{[l]}(k) + \|E_i^{[l]}(k)\|^2, 0 \leq \beta \leq 1, \end{cases}$$

где $E_i^{[l]}(k) = w_j^{[l]'}(u_j^{[l+1]}(k)) \nabla_{S_{ji}} u_j^{[l+1]}$;

5) вычисление локальных приращений весов с помощью соотношений:

$$\Delta w_{jip}^{[l]}(k) = w_{jip}^{[l]}(k+1) - w_{jip}^{[l]}(k) = \gamma^{[l]}(k) \delta_j^{[l+1]}(k) \frac{\partial u_j^{[l+1]}(k)}{\partial w_{jip}^{[l]}};$$

$$\Delta v_{jip}^{[l]}(k) = \gamma^{[l]}(k) \delta_j^{[l+1]}(k) \frac{\partial u_j^{[l+1]}(k)}{\partial v_{jip}^{[l]}};$$

6) уточнение синаптических весов по всей обучающей выборке (за эпоху) с помощью соотношений:

$$\Delta w_{jip}^{[l]}(k) = \sum_{k=1}^N \Delta w_{jip}^{[l]}(k) = -\gamma^{[l]} \sum_{k=1}^N \frac{\partial J^k}{\partial u_j^{[l+1]}(k)} \cdot \frac{\partial u_j^{[l+1]}(k)}{\partial w_{jip}^{[l]}}, \quad \Delta v_{jip}^{[l]}(k) = \sum_{k=1}^N \Delta v_{jip}^{[l]}(k).$$

Разработанный метод [14] предназначен для решения задач настройки многомерных нестационарных стохастических динамических существенно нелинейных объектов, функционирующих в условиях априорной и текущей структурной и параметрической неопределенности, и обладает как сглаживающими, так и следящими свойствами, а также сочетает возможности как градиентных процедур, так и процедур оптимизации второго порядка.

Предложенный метод обучения нейронных сетей на нейронах-фильтрах отличается высокой скоростью сходимости благодаря оптимальному выбору шага поиска, что позволяет повысить стабильность при обработке зашумленных данных.

Выводы

Научная новизна. В результате проведенного исследования были проанализированы возможности моделирования структуры нейросетей и обобщены методы обучения нейронных сетей, предназначенных для прогнозирования нестационарных рядов показателей, описывающих поведение многосвязных систем.

Метод обучения прогнозирующих нейросетей с прямой передачей информации на динамических нейронах-фильтрах с конечной импульсной характеристикой имеет повышенные сглаживающие свойства и отличается высоким быстродействием, что позволяет использование в режиме реального времени.

Метод обучения нейросети с прямой передачей информации на основе минимизации робастного критерия оценивания Вэлша характеризуется вычислительной простотой, что дает возможность использования в режиме реального времени и работы в условиях негауссовских помех.

Метод обучения прогнозирующих локально-рекуррентных нейронных сетей на нейронах-фильтрах с бесконечной импульсной характеристикой отличается от известных методов высокой скоростью сходимости благодаря оптимальному выбору шага поиска, что позволяет повысить стабильность при обработке зашумленных данных.

Практическая значимость. Решение поставленной задачи исследования дает возможность предсказать нежелательные режимы поведения сложных многосвязных систем и улучшить качество процессов управления ими.

Перспективы исследования. Методы обучения прогнозирующих нейросетей, отличающиеся высоким быстродействием и повышенными сглаживающими свойствами, могут быть использованы для дальнейшего решения перспективных задач моделирования и прогнозирования поведения динамических многосвязных систем в условиях априорной и текущей неопределенности относительно их структуры и параметров.

Список литературы: 1. *Rey J. W. W.* Robust Statistical Methods Berlin-Heidelberg-New York: Springer-Verlag, 1978. 128 p. 2. *Welsh R. E.* Nonlinear statistical data analysis // Proc. Comp. Sci. and Statist. Tenth Ann. Symp. Interface. Held at Nat'l Bur. Stds. Gaithersburg, MD. 977. P. 77–86. 3. *Nelles O.* Nonlinear System Identification. Berlin: Springer, 2001. 785 p. 4. Applications and Innovations in Intelligent Systems XV: Proceedings of AI-2007, the Twenty-seventh SGA International Conference on Innovative Techniques and Applications of Artificial Intelligence / Birkhдuser, 2008. 350 p. 5. *Cichocki A., Unbehach R.* Neural Networks for Optimization and Signal Processing. Stuttgart: Teubner, 1993. 526 p. 6. *Chen S., Billings S.A., Grant P. M.* Non-linear systems identification using neural networks // International Journal of Control. 1990. V 51. P. 1191-1214. 7. *Bodyanskiy Ye., Kolodyazhniy V., Stephan A.* An adaptive learning algorithm for a neuro-fuzzy network // Computational Intelligence. Theory and Applications. Berlin - Heidelberg - New York: Springer, 2001. P. 68–75. 8. *Бодянский Е. В.* Обнаружение разладок в нелинейных стохастических последовательностях с помощью рекуррентных искусственных нейронных сетей // Проблемы бионики. 1998. Вып. 49. С. 23–33. 9. *Hornik K.* Multilayer feedforward networks are universal approximators / K. Hornik, M. Stindcombe, H. White // Neural Networks. 1989. 2. P. 359-366. 10. *Wan E.* Temporal backpropagation: an efficient algorithm for finite impulse response neural networks // Proceedings of the 1990 Connectionist Models Summer School: Morgan Kaufmann, 1990. P. 131–140. 11. *Бодянский Е. В., Попов С.В., Чепенко Т.Е.* Прогнозирующая адаптивная нейронная сеть с динамическими нейронами-фильтрами // Радиоэлектроника и информатика. 2003. №2 (23). С. 48–51. 12. *Плисс И.П., Чапланов А.П., Чепенко Т.Е.* Нейросетевая робастная идентификация объекта управления // Радиоэлектроника и информатика. 2002. №2 (19). С. 52 - 54. 13. *Back A.D., Tsoi A.C.* FIR and IIR synapses, a new neural network architecture for time series modeling // Neural Computation. 1991. № 3. P. 375-385. 14. *Бодянский Е. В., Попов С.В., Чепенко Т.Е.* Алгоритм обучения локально рекуррентной нейронной сети в задаче идентификации нелинейных динамических объектов // АСУ и приборы автоматки. 2004. Вып. 126. С.17-22.

Поступила в редколлегию 09.11.2011

Чепенко Татьяна Евгеньевна, научный сотрудник ПНИЛ АСУ ХНУРЭ. Научные интересы: искусственные нейронные сети, прогнозирование временных рядов. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (057) 702-18-90.

УДК 004.8:656:168.4

А.В. ОСТРОУХ, А.Б. НИКОЛАЕВ, Д.Б. ЕФИМЕНКО, С.В. ЖАНКАЗИЕВ

НАУЧНЫЙ ПОДХОД К РАЗРАБОТКЕ АВТОМАТИЗИРОВАННОЙ НАВИГАЦИОННОЙ СИСТЕМЫ ДИСПЕТЧЕРСКОГО КОНТРОЛЯ И УЧЕТА РАБОТЫ ТРАНСПОРТА НЕФТЕДОБЫВАЮЩИХ И НЕФТЕПЕРЕРАБАТЫВАЮЩИХ ПРЕДПРИЯТИЙ

Предлагается подход к разработке организационного обеспечения и комплекса технических решений автоматизированной навигационной системы диспетчерского контроля и учета работы транспорта нефтедобывающих предприятий. Предложенные решения основаны на принципах обеспечения функционирования целостной структуры системы, вычислительных средствах, устанавливаемых в подразделениях органов общего учета и контроля работы транспорта нефтедобывающих предприятий, а также использовании спутниковой связи. Работа выполнена при финансовой поддержке Правительства Российской Федерации (Минобрнауки России) в рамках реализации проекта по Договору № 13.G25.31.0064 от 22 октября 2010 г.

1. Введение

Развитие современных подходов к обеспечению эффективности транспортных процессов требует автоматизированного диспетчерского управления и сопровождения автомобильных транспортных средств.

Транспортное обслуживание нефтедобывающих предприятий находится в сфере доступности автомобильного транспорта, а использование других видов транспорта на указанных перевозках является, как правило, неоправданным. Рост нефтедобычи, усложнение условий возведения транспортной инфраструктуры из-за географического разнесения областей нефтедобычи обуславливают необходимость постоянного совершенствования технологий управления, координирования и контроля организации транспортного процесса.

Вместе с тем, возрастающие риски отклонения запланированных маршрутных траекторий движения транспорта, трудоемкость поисков вследствие возникновения нештатных ситуаций во время реализации транспортного процесса создают трудности при внедрении математических методов планирования перевозок, предъявляют повышенные требования к их регулированию и «гибкости» планирования, разрешить которые возможно при комплексном внедрении автоматизированных систем управления транспортным процессом [1].

Целью настоящего исследования является подготовка исходных и аналитических материалов для реализации проекта построения автоматизированной навигационной системы диспетчерского контроля (АНСДК) и учета работы транспорта нефтедобывающих предприятий – в рамках комплексного проекта: «Создание высокотехнологичного производства инновационной установки для поверки узлов учета нефтегазоводяной смеси на местах добычи, в нефтедобывающих предприятиях».

Для достижения указанной цели решаются следующие задачи:

1. Разработка организационной структуры АНСДК.
2. Выявление основных компонентов АНСДК и взаимосвязей между ними.
3. Разработка схемы взаимодействия отдельных участников перевозочного процесса внутри АНСДК.
4. Разработка комплекса технических решений для обеспечения эффективного функционирования АНСДК.

2. Организационное обеспечение автоматизированной навигационной системы диспетчерского контроля

Для организации эффективной работы транспорта нефтедобывающих предприятий необходимо очертить круг задач, решение которых требует эффективность функционирования. В рамках перевозочного процесса можно выделить следующие задачи [2]:

- подготовка исходной информации (определение кратчайших расстояний, компоновка распределения обслуживаемых территорий, микро- и макрозонирование, создание моделей транспортной сети);
- оптимизация грузопотоков, т.е. закрепление транспортных источников за объектом притяжения;
- маршрутизация (единичные и мелкопартионные отправки);
- выбор определенного типа мобильной поверочной установки (МПУ) для выполнения перевозок в заданных условиях;
- создание и ведение баз данных, выстроенных по алгоритмам предоставления полной информации о процессах добычи нефти.

Инновационный испытательный и контрольный комплекс, позволяющий исключить воздействие человеческого фактора на результат испытаний, с диагностической возможностью моделирования режимов эксплуатации на объекте представляет собой систему, сочетающую организационно-техническое объединение основных объектов, органов аппаратов и служб контроля, планирования, сопровождения, управления, анализа и пр.

При общем рассмотрении организационной структуры системы производится первичное построение основных компонентов:

- органы общего управления и контроля (областные Администрации, областные, региональные Управления по транспорту и связи и т.д.);
- службы планирования, мониторинга и управления транспортным процессом;
- объекты оперативного диспетчерского сопровождения транспорта нефтедобывающих предприятий (ЦМ, ОЦ, диспетчерские центры и пр.);
- региональные органы, осуществляющие сбор статистической информации о выполнении транспортной работы;
- органы контроля и учета качества процесса добычи полезных ископаемых;
- сервисный центр по техническому обслуживанию мобильного и стационарного оборудования системы;
- представители сегмента предприятий-потребителей добываемой продукции;
- служба технического обеспечения работоспособности поверочного оборудования;
- официальные представители транспортных организаций в мультимодальном сообщении;

- региональные службы обеспечения безопасности перевозочного процесса;
- технические комплексы, обеспечивающие выполнение функций управления;
- совокупность служб технического обеспечения МПУ;
- комплексы средств, сетей и линий связи, обеспечивающих связь между организационными элементами системы, передачу информации и управляющих воздействий для выполнения задач сопровождения перевозок транспортом нефтедобывающих предприятий.

Взаимодействие на различных уровнях системы осуществляется путем сопряжения комплекса распространенных элементов организационной структуры системы (рис. 1).



Рис. 1. Элементы организационной структуры системы

Комплекс элементов включает в себя совокупность органов (секторов, центров) различных целевых значений:

- Органы планирования. Оперативное планирование направлено на автоматизацию текущего планирования перевозочной деятельности автотранспортного предприятия (АТП) и предназначено для решения следующих задач: расчет провозных возможностей АТП; расчет оптимальных маршрутов движения мобильных поворочных установок; составление почасовых графиков работы МПУ; составление плана работ по участкам нефтедобычи; расчет предполагаемых затрат и необходимых ресурсов для выполнения транспортной работы; составление сменно-суточного плана работы АТП; составление графика выпуска МПУ на линию; оформление путевой документации. Входная информация формируется на основании данных о потребностях в перевозках на данный период времени исходя из заключенных договоров и поступивших разовых заявок на перевозки, оценки провозных возможностей предприятия, сделанных на основании данных о количестве исправных МПУ в общей структуре парка и готовых к работе водителей. Основными выходными документами являются сменно-суточный план, графики работы МПУ и путевые документы.

– Органы мониторинга. Важную роль в эффективности работы системы занимают средства сбора данных о работе МПУ на линии и средства передачи принятых решений исполнителям. Данные действия в основном закреплены за центром мониторинга. Он является связующим звеном транспортных средств и органов контроля и планирования транспортной деятельности, для которых рабочими являются данные, полученные от бортов МПУ. В этой связи входными данными служат оперативные отчеты от телематического оборудования, установленного на МПУ, функциональное обеспечение которого определяется исходя из требований к оценочным параметрам МПУ и объекта перевозки. Выходными данными являются отметки о движении МПУ, распределенные в соответствии с формами запросов, сообщения о нарушениях в графиках движения МПУ, оперативные отчеты и пр.

– Органы управления. Подсистема оперативного управления занимает ключевое место в организации перевозочного процесса. В систему в режиме реального времени поступает информация о выпуске МПУ на маршрут и оперативная информация о работе МПУ, исходящая от органов мониторинга перевозочного процесса. Поступающая оперативная информация сравнивается с запланированными графиками. В случае расхождения фактических результатов с запланированными выясняется причина срыва, проводится поиск оптимального решения для продолжения работы в изменившихся условиях и выполняется корректировка заданий водителям в автоматизированном режиме в соответствии с распределенными сюжетными алгоритмами принятия решения. В центре оперативного управления как минимум реализуется функция оперативного контроля, которая позволяет следить за ходом выполнения сменного-суточного плана в режиме реального времени. Для реализации этой функции достаточно тем или иным образом взаимодействовать с органами мониторинга, после чего имеется возможность сравнить фактические данные с запланированными. Больше возможностей для управления имеют системы, в которых реализована функция оперативного регулирования, которое позволяет вырабатывать управляющие воздействия на перевозочный процесс при расхождении фактических данных с запланированными. В этом случае система должна иметь программы построения оптимального плана работы, которые способны в режиме реального времени выполнить все необходимые расчеты на основе новых исходных данных. Новый план должен быть своевременно доведен до исполнителей.

– Органы проведения аналитических вычислений. Данная структура позволяет получить своевременную информацию о результате работы МПУ и выявить основные причины невыполнения запланированных работ, а также по запросу эксплуатирующего предприятия и других легитимных пользователей сформировать всю необходимую информацию о перевозочном процессе на определенный момент времени. Входной информацией являются данные с путевых листов и товарно-транспортных накладных, а также фактические результаты работы МПУ, зафиксированные в системе оперативного управления на основе оперативных отчетов органов мониторинга. Результатом обработки этой информации являются технико-эксплуатационные показатели работы МПУ; изменение заработной платы сотрудников за выполненную работу; накопление данных о работе водителей и АТС; уточненное значение размера оплаты за выполненную работу; величина доходов предприятия и фактическая себестоимость перевозок; рекомендации по улучшению работы парка.

– Региональные отделения и локальные операторские центры. Региональные отделения локальных операторских центров непосредственно контролируют работу и ведение информационного сопровождения в соответствии с территориальным распределением локальных зон движения МПУ в рамках маршрутной сети системы.

– Службы эксплуатации и ответственные представители организаторов перевозочного процесса.

Объектом оперативного управления является МПУ, работающая на маршрутной сети.

– Органы, осуществляющие сбор статистической информации о выполнении транспортной работы.

Пользователи информации: официальные представители заказчиков транспортной работы; администрации регионов, транспортных компаний сопутствующих направлений, социальной сферы, организации дорожного движения в регионах; Государственные органы проведения мероприятий в соответствии с регламентами позиционирования на уровнях

контроля, учета, анализа и управления закрепленными процессами; Региональные службы сопровождения; Службы немедленного реагирования (МЧС, ГУ ГОЧС, ГУВД, другие региональные и городские оперативные и аварийные службы).

Организация перевозок разрабатывается на разных уровнях функционирования системы. При разработке организационной структуры на уровне автомобильных парков эксплуатирующего предприятия, задействованных в общем перевозочном процессе, особое внимание уделяется методам решения задач, одними из которых являются:

1. Полное удовлетворение требований центральных органов управления перевозочным процессом в рамках проведения транспортировочных работ.

2. Обеспечение обслуживания деятельности нефтедобывающих предприятий на максимально эффективном уровне.

3. Выполнение существующих планов, разработанных в центре планирования, центральной диспетчерской службой.

4. Эффективное использование МПУ, повышение производительности труда, максимальное снижение транспортных расходов.

5. Удовлетворение требований потребителей в сегменте обеспечения поверочных установок согласно разработанным проектам и договоренностям.

6. Исключение воздействия человеческого фактора на результат испытаний с диагностической возможностью моделирования эксплуатации на объекте.

Архитектура системы диспетчерского управления автомобильным транспортом, контролируемым на основе использования спутниковой навигации, в значительной степени зависит от выстроенных организационных принципов, а именно от базовых компонентов, формирующих организационную структуру. Базовые компоненты организационной структуры автоматизированной навигационной спутниковой системы диспетчерского контроля и учета работы транспорта нефтедобывающих предприятий приведены на рис. 2.

При создании центральной диспетчерской системы (ЦДС), в основном, закладываются функции координации оперативного управления при нештатных ситуациях, а также анализа исполненной на линии работы в различных разрезах (по объектам притяжения, МПУ, по маршрутам). В дальнейшем на ЦДС могут быть возложены функции по координации текущего планирования перевозок.

Функциональная специализация автоматизированных рабочих мест специалистов автотранспортного предприятия, работающих в системе диспетчерского управления транспортом на основе использования спутниковой навигации, должна соответствовать определенным принципам построения:

1. Автоматизированное рабочее место (АРМ) технолога. На АРМ технолога в постоянно действующем режиме производится формирование нормативно-справочной информации о клиентах автотранспортного предприятия (грузоотправителях и грузополучателях), о расстояниях нулевых пробегов, о маршрутах перевозок, о МПУ и о водителях АТП, а также о комплексных контрольных пунктах. Кроме того, на АРМ технолога составляется первая часть сменно-суточного плана на следующие сутки.

2. АРМ диспетчера. На АРМ диспетчера производится корректировка сменно-суточного плана и графика выпуска МПУ.

Диспетчер на своем АРМ осуществляет постоянный мониторинг перемещений МПУ на линии. При этом автоматизированная система должна выдавать сообщения оперативному диспетчеру только при возникновении нештатных ситуаций: при опозданиях выпуска на линию, при простоях, сходах с линии и внеплановых возвратах в АТП. В случае необходимости, по инициативе диспетчера, с водителем может быть установлена голосовая связь. Кроме голоса, с АРМ диспетчера может быть реализована возможность передачи служебных сообщений водителю на дисплей бортового, мобильного модуля. В любой момент времени оперативный диспетчер должен иметь возможность сформировать оперативные справки об исполненном движении автомобилей на линии.

Анализ выполненной автомобилями работы на линии производится на АРМ старшего сменного диспетчера (или на специальном АРМ аналитика движения). Анализ проводится как за предыдущий, так и за любой другой прошедший день.

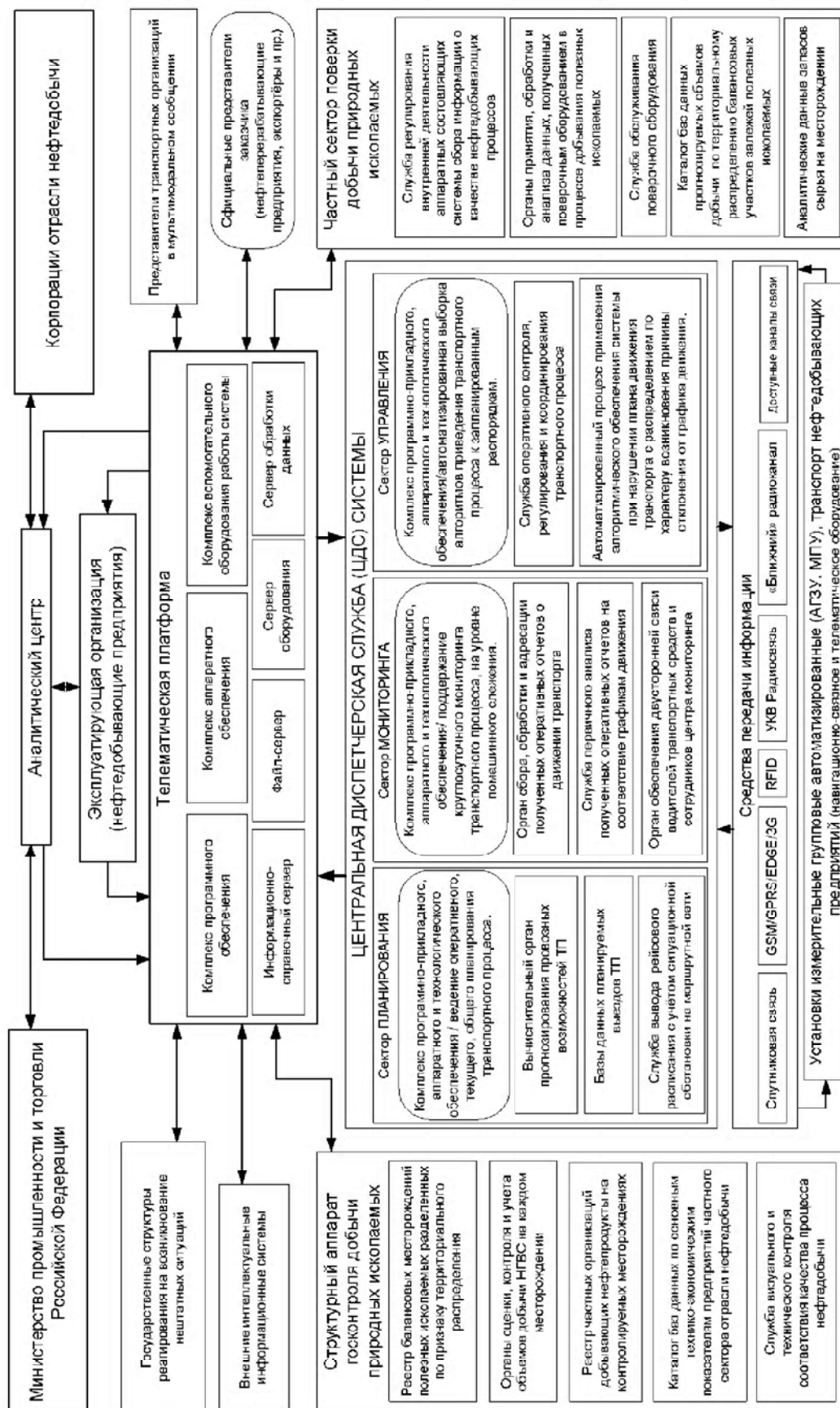


Рис. 2. Базовые компоненты организационной структуры системы

При подготовке к выезду из АТП водитель, кроме традиционных процедур медицинского осмотра и технического контроля состояния МПУ, проходит технический контроль мобильного навигационно-связного оборудования, получает у оперативного диспетчера задание (подробный график - расписание) своей работы на линии и путевой лист. Объектами оперативного управления являются МПУ, работающие на маршрутной сети, сформированной в соответствии с территориальным распределением участков нефтедобычи и производственных баз эксплуатирующего предприятия.

При рассмотрении условий взаимодействия задействованных в транспортных, производственных, контрольных и других процессах предприятий выделяют определенные сегменты. В рамках сегмента транспортного обеспечения потребностей в перевозочной деятельности нефтедобывающего предприятия в соответствии с заданной тематикой представляются наиболее востребованными элементы, формирующие организационную структуру взаимодействия на данном уровне (рис. 3).



Рис. 3. Организация взаимодействия отдельных участников перевозочного процесса

В диспетчерских отделах станций идентификации поверочных данных оборудования, установленного на нефтедобывающих станциях, и у парка специализированного автомобильного транспорта технической помощи должны устанавливаться удаленные рабочие места, подключаемые к системе. За диспетчерскими подразделениями должны по-прежнему остаться функции учета и обработки соответствующей документации (в основном, информационные массивы определенного образца и путевые листы учета транспортной работы).

Особое внимание при организации транспортного процесса уделяется использованию различных методов, обеспечивающих:

- своевременность прибытия на контрольные пункты;
- сохранность качества и количества объекта перевозки;

- выполнение требований техники безопасности и требований безопасности движения, охрану окружающей среды;
- выполнение требований трудового законодательства.

3. Комплекс технических решений автоматизированной навигационной системы диспетчерского контроля

Комплекс технических решений для обеспечения эффективного функционирования системы диспетчерского контроля и учета работы транспорта нефтедобывающих предприятий включает в себя [3]:

- технические компоненты и комплексы средств автоматизации управления, в том числе бортовые абонентские терминалы, а также средства вычислительной техники, устанавливаемые в центральной диспетчерской службе (ЦДС) в диспетчерском центре (ДЦ) операторских центров;

- каналы и средства связи радиотелефонной сети и технические средства сопряжения с ними для обеспечения оперативно-диспетчерской связи, в том числе голосовой, и передачи данных;

- каналы и средства связи и технические средства сопряжения с ними для обеспечения сети оперативно-диспетчерской связи системы;

- каналы и средства связи сотовой радиотелефонной связи (GPRS, CDMA) и технические средства сопряжения с ними для обеспечения сети оперативно-диспетчерской связи системы;

- каналы связи проводной корпоративной сети передачи данных и технические средства сопряжения с ними;

- телематическое обеспечение передвижных объектов;

- каналы и средства спутниковой связи и сотовой радиотелефонной сети передачи данных и технические средства сопряжения с ними.

Вычислительные средства, устанавливаемые в подразделениях органов общего учета и контроля работы транспорта нефтедобывающих предприятий, в ЦДС центра мониторинга (ЦМ), в диспетчерских центрах (ДЦ) и операторских центрах системы, подразделяются на следующие категории:

- сервер баз данных;

- коммуникационный сервер;

- рабочая станция;

- сетевое и коммуникационное оборудование.

Сервер баз данных обеспечивает: накопление, хранение, ведение баз данных системы и доступ к ним; обмен данными ЦДС ЦМ с ДЦ операторских центров и ЛВС региональных грузоперевозчиков, с другими модулями системы.

Коммуникационный сервер обеспечивает накопление и временное хранение навигационных данных по мобильным поверочным установкам, а также коммутацию каналов связи при реализации режимов обмена данными (включая голосовую связь) между диспетчерами системы и водителями МПУ. Рабочие станции модуля ЦДС обеспечивают реализацию автоматизированных технологий информационного сопровождения и диспетчерского управления в соответствии со специализацией автоматизированных рабочих мест.

Сетевое и коммуникационное оборудование системы должно обеспечить: обмен данными между элементами вычислительных сетей системы и серверами радиооборудования, устанавливаемыми в диспетчерских центрах; обмен данными между объектами распределенной сети системы по предоставляемым каналам связи; передачу данных на сервер, определенный соответствующим регламентом – для информирования участников перевозочного процесса. Распределенная сеть системы приведена на рис. 4.

Связь в локальной компьютерной сети ЦДС в ЦМ осуществляется через коммутаторы, объединенные в стек. Минимальные требования к характеристикам коммутатора: скорость коммутации для собранного стека из двух коммутаторов – 48 Гбит/с; объединение в стек до восьми устройств; отказоустойчивое объединение в стек с возможностью «горячей» замены коммутаторов; распределенная отказоустойчивая маршрутизация (Distributed Resilient Routing); хранение таблиц маршрутизации во всех коммутаторах; между устройствами не существует отношений главный-подчиненный.

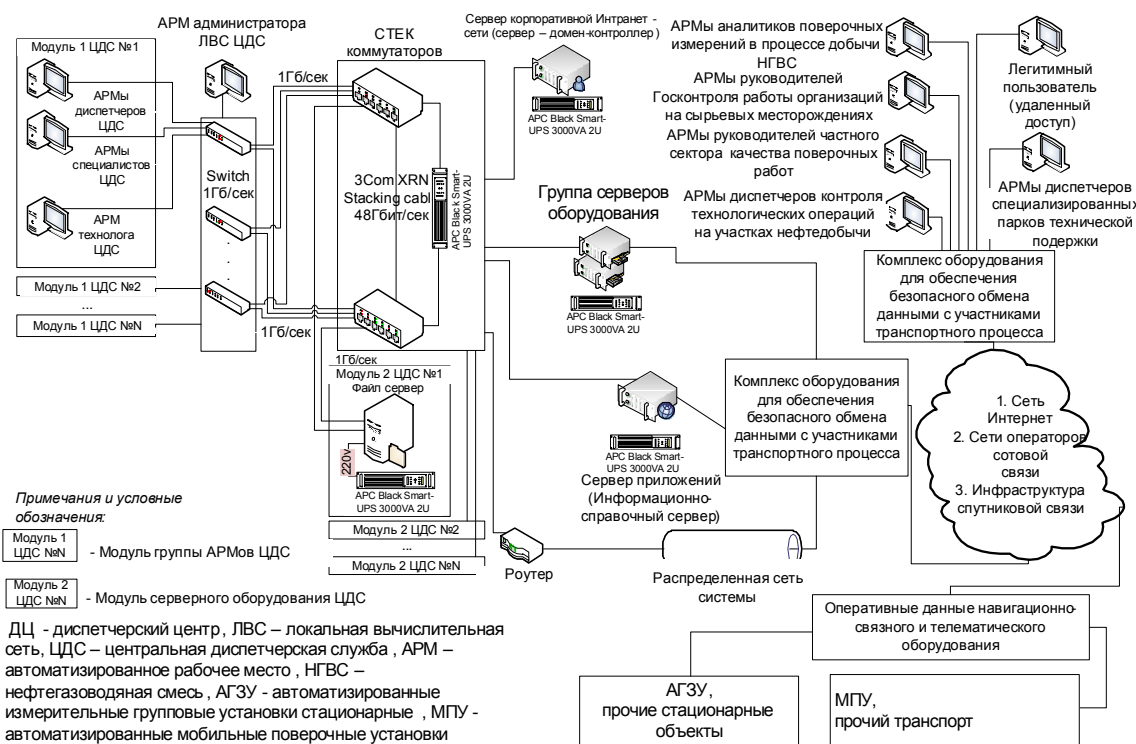


Рис. 4. Распределенная сеть системы

Сервер, выполняющий функции контроллера домена, обеспечивает управление сетью: разграничивает права доступа, обеспечивая, таким образом, защищенный доступ к ресурсам сети.

Сервер приложений выполняет обработку запросов, поступающих от внешних (не ЦДС) пользователей к данным ЦДС ЦМ.

Основная задача сервера внешних приложений – полностью автоматизированный прием и распределение информации для внешних региональных информационных систем и пользователей.

Файл-сервер обеспечивает: накопление, хранение, ведение баз данных системы; доступ к базам данных со стороны отдельных пользователей ЦДС ЦМ и групп пользователей ЦДС со своих рабочих мест с учетом установленных разграничений доступа; обмен данными с организаторами готовности ПС грузоперевозчика, с другими модулями ЦДС ЦМ и другими информационными системами.

Источником оперативных отчетов о движении МПУ нефтедобывающих предприятий, которые в дальнейшем проходят этапы приема, обработки, анализа, накопления, хранения и прочее, является мобильное навигационно-связное оборудование, которым оснащаются МПУ нефтедобывающих предприятий.

С учетом специфики перевозочного процесса МПУ нефтедобывающих предприятий отдельного внимания заслуживает спутниковая связь, используемая в автоматизированной навигационной системе диспетчерского контроля и учета работы транспорта нефтедобывающих предприятий. Основное достоинство спутниковой связи - возможность вести телефонные переговоры в любой точке мира, тогда как владельцы сотовых телефонов могут разговаривать только на территории покрытия станциями сотовой сети. Все сети спутниковой связи предоставляют возможность надежной качественной телефонии. Различия между ними состоят в наборе дополнительных услуг, предлагаемых абоненту (факс, телекс, доступ в Интернет) и области покрытия (некоторые системы не работают на территории Южного и Северного полюсов).

В России действуют системы Инмарсат, Глобалстар, Иридиум и Турайя [4].

Инмарсат (Inmarsat) – первый оператор мобильной спутниковой связи в мире, единственный, кто предлагает полный набор услуг современной спутниковой связи для морских, наземных и воздушных приложений. Глобалстар (Globalstar) – провайдер мобильных услуг спутниковой связи нового поколения, предоставляющий телефонную связь в те районы, услуги связи в которых были ранее недоступны или ограничены. Глобалстар предоставляет доступ к передаче голоса и данных из практически любого населенного района мира.

Российский наземный сегмент системы Глобалстар включает 3 станции сопряжения, расположенные под Москвой, Новосибирском и Хабаровском. Они обеспечивают покрытие на всей территории России южнее 70-й параллели с гарантированным качеством обслуживания.

Иридиум (Iridium) – беспроводная спутниковая сеть, созданная для обеспечения телефонной связи в любой точке планеты в любое время. Универсальный доступный сервис – новые возможности для бизнеса и жизни. Она доступна на всей территории Российской Федерации.

Оснащение передвижных поверочных установок средствами навигации позволит иметь актуальную информацию об их местонахождении на диспетчерском пункте, управлять их перемещением.

Развитие такого подхода позволит использовать системы мониторинга, в которой результаты измерений с использованием системы ГЛОНАСС поступают в централизованную БД, где накапливаются, хранятся и обрабатываются в автоматизированном режиме.

4. Заключение

Формирование и построение в рамках комплексного проекта: «Создание высокотехнологичного производства инновационной установки для поверки узлов учета нефтегазовой смеси на местах добычи, в нефтедобывающих предприятиях» автоматизированной системы учета и контроля автомобильного транспорта нефтедобывающего предприятия направлено на обеспечение эффективного решения целого ряда комплексных задач, важнейшими из которых являются следующие:

- оперативное планирование сопутствующих транспортному процессу операций для оптимизации затрат физических и материальных ресурсов в единицу времени;
- автоматизированное определение местонахождения подвижных объектов в составе систем комплексного обеспечения безопасности;
- охват максимального количества возможных контролируемых параметров транспорта и объекта перевозок;
- редактирование и выбор оптимальных решений изменения плана движения, в связи с непредвиденными обстоятельствами, в режиме реального времени;
- сопряжение перевозок разных типов и характеристик перевозимых объектов при сохранении параметров эффективности перевозок в целом;
- совершенствование структурной основы взаимодействия основных участников перевозочного процесса;
- оптимизация расходов на обслуживание и ремонт автомобильной и специальной техники.

На базе использования современных навигационных систем GPS и ГЛОНАСС может быть построена распределенная корпоративная автоматизированная система для оперативного доступа к результатам учета работы транспорта, независимо от удаленности транспортных объектов нефтедобывающих компаний.

Список литературы: 1. Буслаев А.П., Новиков А.В., Приходько В.М., Тагищев А. Г., Яшина М. В. Вероятностные потоки и имитационные подходы к оптимизации автодорожного движения М.:Мир, 2003. 368 с. 2. Вельможин А.В., Гудков В.А., Миротин Л.Б. Технология, организация и управление грузовыми автомобильными перевозками: Учебник для вузов. Волгоград: Волгогр. гос. техн. ун-т, 2000. 304 с. 3. Власов В.М., Ефименко Д.Б. Координатно-временное и навигационное обеспечение (КВНО) как единая информационная основа автоматизации базовых технологий на транспорте // Фундаментальное и прикладное координатно-временное и навигационное обеспечение (КВНО – 2007), Вторая Всероссийская конференция, СПб., 2007. С. 137-138. 4. Финько В.И. Автоматизированные радионавигационные системы диспетчерского управления – ключ к реформированию городского пассажирского транспорта // Автотранспортное предприятие. 2002. № 12. С.41-43.

Поступила в редколлегию 12.11.2011

Остроух Андрей Владимирович, д-р техн. наук, член-корреспондент РАЕ, профессор кафедры «Автоматизированные системы управления», ФГБОУ ВПО Московский автомобильно-дорожный государственный технический университет (МАДИ). Научные интересы: автоматизация технологических процессов и производств, разработка электронных образовательных ресурсов, дистанционное обучение. Адрес: Россия, 105066, Москва, ул. Нижняя Красносельская, 45/17, кв. 30, тел. +7 (985) 928 6908, e-mail: ostroukh@mail.ru

Николаев Андрей Борисович, лауреат премии Правительства РФ, заслуженный деятель науки РФ, д-р техн. наук, проф., декан факультета «Управление», заведующий кафедрой «Автоматизированные системы управления», ФГБОУ ВПО Московский автомобильно-дорожный государственный технический университет (МАДИ). Научные интересы: автоматизация технологических процессов и производств, разработка электронных образовательных ресурсов. Адрес: Россия, 125368, Москва, ул. Барышиха, 33, корп. 1, кв. 80, тел. +7 (499) 155 0472, e-mail: nikolaev.madi@mail.ru

Ефименко Дмитрий Борисович, канд. техн. наук, доцент кафедры «Транспортной телематики», ФГБОУ ВПО Московский автомобильно-дорожный государственный технический университет (МАДИ). Научные интересы: транспортные и транспортно-технологические системы страны, ее регионов и городов, организация производства на транспорте. Адрес: Россия, 125318, Москва, Ленинградский проспект, 64, тел. +7 (499) 155 0838, e-mail: ed2002@mail.ru

Жанказиев Султан Владимирович, канд. техн. наук, профессор кафедры «Транспортной телематики», ФГБОУ ВПО Московский автомобильно-дорожный государственный технический университет (МАДИ). Научные интересы: транспортные и транспортно-технологические системы страны, ее регионов и городов, организация производства на транспорте. Адрес: Россия, 125318, Москва, Ленинградский проспект, 64, тел. +7 (499) 155 0838, e-mail: info@tt.madi.ru

ТЕХНОЛОГИЯ СЖАТИЯ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ НА ОСНОВЕ ДВУХКОМПОНЕНТНОГО КОДИРОВАНИЯ

Излагаются основные этапы разработки информационной технологии сжатия изображений на основе обобщенного кодирования его координатно-структурной и построчно-масштабирующей составляющих, базирующейся на: формировании массивов длин координатно-структурной и построчно-масштабирующей апертурных составляющих; вычислении оснований элементов массивов аппроксимирующих величин апертур и массивов длин апертур, рассматриваемых соответственно как адаптивное позиционное число с неравными соседними элементами и позиционное число в дифференциальном пространстве; построении двухкомпонентного кода.

1. Введение

Для обеспечения комплексной безопасности, развития космической отрасли, телекоммуникаций, связи и информатизации необходимо обеспечить [1 – 3]: мониторинг стратегических объектов; мониторинг и безопасность мероприятий международного значения; мониторинг ликвидации последствий чрезвычайных ситуаций естественного и техногенного характера. Необходимо усовершенствование существующих и внедрение новых форм и способов управления кризисными ситуациями. В первую очередь это касается повышения эффективности информационного обеспечения с использованием дистанционных средств аэрокосмического базирования. При этом основная научно-прикладная проблема, требующая своего решения, заключается в сокращении времени доставки видовых изображений с необходимой разрешающей способностью. В то же время, как показал анализ, существующие возможности бортовой аппаратуры передачи данных не обеспечивают *своевременную* доставку оцифрованных изображений [4; 5]. Возникает противоречие между требуемыми характеристиками процессов доставки данных, а именно временем обработки, передачи, качеством восстанавливаемых изображений и реальными характеристиками

для существующих комплексов беспилотных авиационных систем. Повышение оперативности доведения информации возможно на основе уменьшения объемов обрабатываемых и передаваемых видеоданных с использованием технологий компрессии. При этом в условиях аэрокосмического мониторинга требуется использовать методы, эффективно кодирующие насыщенные реалистические изображения с контролируемой сложностью технической реализации. В работе [5; 6] показано, что с таких позиций эффективной является технология с применением построения апертурных составляющих изображений. В данном случае видеоданные описываются апертурами, т.е. $x_{\xi, \gamma+r} = h_{\xi}$, $r=0, \ell_{\xi}-1$, где h_{ξ} – аппроксимирующее значение для ξ -й видеопоследовательности, элементы $x_{\xi, \gamma+r}$ которой имеют ограниченные динамические диапазоны, ℓ_{ξ} – длина ξ -го участка с ограниченным динамическим диапазоном.

В то же время существующие технологии, реализующие сжатие на основе предварительного выявления апертур, базируются на раздельной обработке их составляющих. Это является одной из причин снижения уровня сжатия изображений. Поэтому необходимо использовать подход относительно обобщенной компрессии составляющих апертур. Отсюда *цель исследований* заключается в разработке информационной технологии компактного представления на основе обобщенного представления двух компонент апертурного описания изображений.

2. Обоснование концепции создания информационной технологии двухкомпонентного кодирования

На организацию процессов обработки для методов компактного представления изображений существенно влияют:

1) характер обработки видеоданных, а именно для какой последующей обработки предназначены сжатые изображения, например, для хранения в информационных системах или/и для передачи между пользователями с использованием телекоммуникационных систем;

2) существующие ограничения на вычислительные возможности и на характеристики по скорости передачи данных в каналах связи для используемых информационно-телекоммуникационных систем.

Основными ключевыми механизмами, определяющими методологию процессов сжатия и восстановления изображений, являются:

1. Виды сокращаемой избыточности. Методология устранения избыточности изображений определяет класс используемых трансформаций и методов кодирования, оказывая, таким образом, влияние на потенциальные возможности относительно степени компактного представления, времени обработки и уровня вносимых искажений. Технология компрессии допускает возможность исключения нескольких видов избыточности. Для предложенного подхода методология сокращения избыточности базируется на формировании для фрагмента изображения двух составляющих, а именно:

– неравномерная координатно-структурная составляющая, которая формирует локально-структурную архитектуру фрагмента изображений. Компонентами такой составляющей являются длины апертур, выявляемых вдоль строк изображения;

– построчно-масштабирующая составляющая, которая определяет яркостную и цветовую насыщенность архитектурной формы фрагмента изображения. Компонентами такой формы являются аппроксимирующие яркостные (цветовые) величины апертур.

2. Наличие предварительного преобразования цветовой модели.

3. Типы и параметры кодеров, используемых для сокращения избыточности. Здесь определяется количество кодовой избыточности, временные затраты на обработку, механизмы сокращения различного количества психовизуальной избыточности.

4. Наличие возможности для дополнительного снижения времени обработки и передачи данных для фиксированного объема компактно представленных видеоданных.

В процессе построения метода кодирования *предлагается* организовывать следующие подходы, а именно формировать:

– кодовое описание заданной длины;

– двухкомпонентное кодовое представление на базе совместного использования элементов координатно-структурного и построчно-масштабного представления фрагмента

изображения (рис. 1). Это обеспечит обработку целостной информации о фрагменте изображения. В этом случае в отличие от бит-ориентированного принципа добавочная группа разрядов формируется на основе взвешенного добавления компоненты апертурно-яркостного описания фрагмента изображения.

Это позволяет:

Первое. Дополнительно повысить степень сжатия за счет сокращения количества незначимых старших разрядов в кодовых комбинациях. Это достигается в результате добавления элементов массивов длин апертур, имеющих меньшие значения динамических диапазонов, в процессе формирования двухкомпонентного кода (ДК).

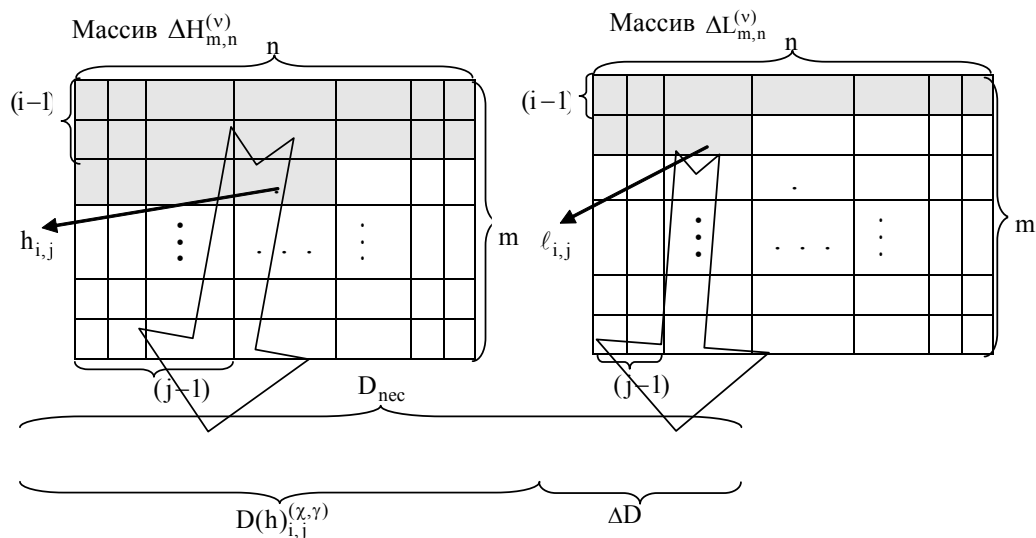


Рис. 1. Схема формирования двухкомпонентного кода

С одной стороны, это обусловлено наличием избыточного количества двоичных разрядов в кодовом слове. Действительно, формирования ДК осуществляется на базе элементов массива аппроксимирующих величин апертур. При этом кодограмма будет заполняться не полностью, т.е. $D(h)_{i,j}^{(χ,γ)} = (\lceil \ell \log_2 E(h)_{i,j}^{(χ,γ)} \rceil + 1) < D_{nec}$. Здесь $E(h)_{i,j}^{(χ,γ)}$ – значение интегрированного кодового описания для массива аппроксимирующих величин апертур, начиная с элемента на позиции $(χ; γ)$ и заканчивая элементом на позиции $(i; j)$; $D(h)_{i,j}^{(χ,γ)}$ – количество двоичных разрядов на представление величины $E(h)_{i,j}^{(χ,γ)}$; D_{nec} – заданное количество двоичных разрядов. Следовательно, в кодовом слове образуется избыточное количество ΔD двоичных разрядов, равное $\Delta D = D_{nec} - D(h)_{i,j}^{(χ,γ)}$, что приводит к снижению коэффициента сжатия. С другой стороны – тем, что для реалистических изображений значения элементов массивов длин апертур в среднем имеют меньший динамический диапазон, чем элементы массивов аппроксимирующих величин. Отсюда избыточное количество разрядов предлагается заполнять за счет дополнительного (добавочного) добавления к интегрированному значению кодового описания, полученного для последовательности аппроксимирующих величин, элементов массива длин апертур.

Второе. Повысить оперативность обработки фрагментов изображений за счет:

- 1) возможности проводить восстановление фрагмента изображения сразу при реконструкции кодового представления;
- 2) отсутствия операций обращения к отдельным разрядам кодового слова, как это делается для бит-ориентированного принципа слияния кодов;
- 3) создания дополнительного механизма для повышения степени сжатия. В случае повышения степени сжатия можно разбивать длины апертур на несколько компонент, введя для этого специальные ограничения, тем самым увеличивая возможность для дополнительного исключения количества незначимых разрядов.

Третье. Снизить вычислительную сложность, необходимую для реализации процессов обработки (т.е. снизить требования к вычислительным системам), а именно: сократить количество обращений к ВЗУ; сократить объем памяти, отводимой для хранения промежу-

точных данных. Например, при отдельном кодировании массивов длин апертур и аппроксимирующих величин требуется полностью хранить восстановленные значения одних компонент апертур до получения других ее компонент.

На основе изложенного материала можно заключить, что для обеспечения дополнительного сокращения кодовой избыточности предлагается организовывать обобщенное формирование кодового описания на основе компонент координатно-структурной и построчно-масштабирующей формы. Двухкомпонентный код строится в результате слияния кода аппроксимирующих величин апертур и взвешенных элементов длин апертур. Это позволяет: дополнительно повысить степень сжатия видеоданных; сократить время восстановления изображений.

3. Разработка технологии компрессии изображений на основе двухкомпонентного кодирования

Избыточное количество разрядов, т.е. $\Delta D > 0$, в процессе формирования ДК для элементов массивов аппроксимирующих величин обусловлено такими причинами:

– некратностью величины динамических диапазонов $w(h)_i - 1$ элементов массива аппроксимирующих значений степени двойки, т.е. $w(h)_i - [w(h)_i / 2]2 \neq 0$, в случае представления кодового описания машинным словом;

– формированием начального значения ДК для максимального значения логарифма динамического диапазона величины $\ell \log_2(w(h)_i - 1)$.

Начальное значение ДК $E(h)_{i,j}^{(i,\gamma)}$ формируется на базе элементов строки массива $\Delta H_{m,n}^{(v)}$, рассматриваемых как адаптивное одномерное позиционное число с неравными соседними элементами. В общем случае значение $E(h)_{i,j}^{(i,\gamma)}$ определяется по формуле

$$E(h)_{i,j}^{(i,\gamma)} = \sum_{\phi=\gamma}^j (h_{i,\phi} - \text{sign}(1 - \text{sign}(h_{i,\phi-1} - h_{i,\phi}))) (w(h)_i - 1)^{j-\gamma+1-\phi},$$

где (i,γ) , (i,j) – координаты соответственно начального и конечного элементов i -й строки, на базе которых формируется ДК.

Отсюда получаем количество $v(h,i)_\xi$ элементов i -й строки, для которых формируется ξ -й код, где $v(h,i) = j - \gamma + 1$. Величина динамического диапазона для элементов строки фиксирована и равна $w(h)_i - 1$. Количество запрещенных последовательностей, обусловленное равенством соседних элементов, равно $\Delta V(h'_{i,j} = h_{i,j-1}) = (h_{i,\max} - h_{i,\min})^{n-j}$, где $(h_{i,\max} - h_{i,\min})^{n-j}$ – количество последовательностей, элементы которых удовлетворяют ограничениям: $h_\xi \neq h_{\xi+1}$, $\xi = \overline{1, n}$; $h_{i,\min} \leq h_1, \dots, h_j, \dots, h_n \leq h_{i,\max}$.

С учетом этого количество элементов $v(h)_\xi$ определяется из условия

$$\ell \log_2 E(h)_{i,j}^{(i,\gamma)} \leq D(h)_{i,j}^{(i,\gamma)} = v(h,i)_\xi \ell \log_2 (w(h)_i - 1) \leq D_{\text{нec}},$$

что позволяет исключить возможность переполнения кодового слова, имеющего заранее заданную длину $D_{\text{нec}}$. Отсюда величина $v(h,i)_\xi$ равна $v(h,i)_\xi = [D_{\text{нec}} / \ell \log_2 (w(h)_i - 1)]$.

Строчная обработка массивов аппроксимирующих величин апертур позволяет сократить вычислительные затраты для определения количества $v(h,i)_\xi$ элементов и проводить процесс кодирования за один проход.

В случае наличия избыточных разрядов, т.е. $\Delta D \neq 0$, формирование второй компоненты ДК проводится на базе элементов текущей α -й строки массива $\Delta L_{m,n}^{(v)}$ длин апертур. Для этого требуется выполнить следующие действия:

1. Определить избыточное количество разрядов ΔD в кодограмме. Для этого используется формула: $\Delta D = D_{\text{нec}} - v(h,i)_\xi [\ell \log_2 (w(h)_i - 1)] - 1$.

После этого можно вычислить количество элементов $v(\ell)_\xi$ для элементов массива $\Delta L_{m,n}^{(v)}$, на базе которых формируется *добавочный* код $\Delta E(\ell)_{\phi,j}^{(\alpha,\gamma)}$ второй компоненты. Здесь первый и последний элементы массива $\Delta L_{m,n}^{(v)}$, входящие в код $\Delta E(\ell)_{\phi,j}^{(\alpha,\gamma)}$, имеют соответственно позиции (α,γ) и (ϕ,j) .

2. Определить количество добавляемых элементов $v(\ell)_\xi$ из условия необходимости исключить возможность переполнения длины ΔD последовательности избыточных разрядов кодового слова, т.е.

$$[\log_2 \Delta E(\ell)_{\phi,j}^{(\alpha,\gamma)}] + 1 \leq \Delta D, \quad (1)$$

где $(\alpha; \gamma), (\phi; j)$ – координаты соответственно начального и конечного элементов второй составляющей на базе массива длин апертур.

Величина добавочного кода $\Delta E(\ell)_{\phi,j}^{(\alpha,\gamma)}$ формируется на базе элементов массива длин апертур, которые рассматриваются как одномерное позиционное число в дифференциальном пространстве с различными основаниями. Величина основания ПЧДП $w(\ell)_{i,j} = \ell_{i,\max} - \ell_{i,\min} + 1$ является константой в пределах одной строки. Отсюда выполняется неравенство

$$\Delta E(\ell)_{\phi,j}^{(\alpha,\gamma)} \leq w(\ell)_\alpha^{n-\gamma+1} \prod_{\chi=\alpha+1}^{\phi-1} w(\ell)_\chi^n w(\ell)_\phi^j. \quad (2)$$

В частном случае, если рассматриваются элементы, принадлежащие одной строке, т.е. $\alpha = \phi$, то $\Delta E(\ell)_{\alpha,j}^{(\alpha,\gamma)} \leq w(\ell)_\alpha^{j-\gamma+1}$.

На основе соотношения (2) для выполнения условия (1) требуется обеспечить выполнение следующего неравенства: $[\log_2 (w(\ell)_\alpha^{n-\gamma+1} \prod_{\chi=\alpha+1}^{\phi-1} w(\ell)_\chi^n w(\ell)_\phi^j)] + 1 \leq \Delta D$. При этом количество элементов для второй составляющей ДК определяется по формуле $v(\ell)_\xi = (n - \gamma + 1) + n(\phi - \alpha - 1) + j$.

В случае известного значения ΔD и выбора длин апертур в пределах α -й строки получим упрощенное выражение для вычисления количества $v(\ell)_\xi$ добавочных элементов $v(\ell)_\xi = v(\ell, \alpha)_\xi = [\Delta D / \log_2 w(\ell)_\alpha]$. Понятно, что поскольку значения оснований элементов ПЧДП в разных строках и величина ΔD являются различными, то количество элементов $v(\ell)_\xi$ второй составляющей будет переменным. В общем случае, если вторая составляющая ДК формируется на базе элементов нескольких строк массива длин апертур, то

величина $v(\ell)_\xi$ будет равна $v(\ell)_\xi = \sum_{i=\alpha}^{\alpha+\beta} v(\ell, i)_\xi$. Здесь $v(\ell, \alpha)_\xi$ – количество элементов в α -й строке массива $\Delta L_{m,n}^{(v)}$, участвующих в построении второй составляющей; $\alpha, \alpha + \beta$ – индексы соответственно начальной и конечной строки массива $\Delta L_{m,n}^{(v)}$, элементы которых образуют вторую составляющую ДК.

Это приводит к необходимости организовывать рекуррентный отбор элементов первой и второй составляющих для формирования ДК. Схема отбора элементов строк массивов $\Delta H_{m,n}^{(v)}$ и $\Delta L_{m,n}^{(v)}$ для построения двухкомпонентного кода приведена на рис. 2, где рассматривается вариант построения ДК в случае неравномерного вклада по количеству компонент из различных массивов в двухкомпонентный код. Элементы строки массива длин апертур будут участвовать в образовании ДК совместно для различных строк массива аппроксимирующих величин.

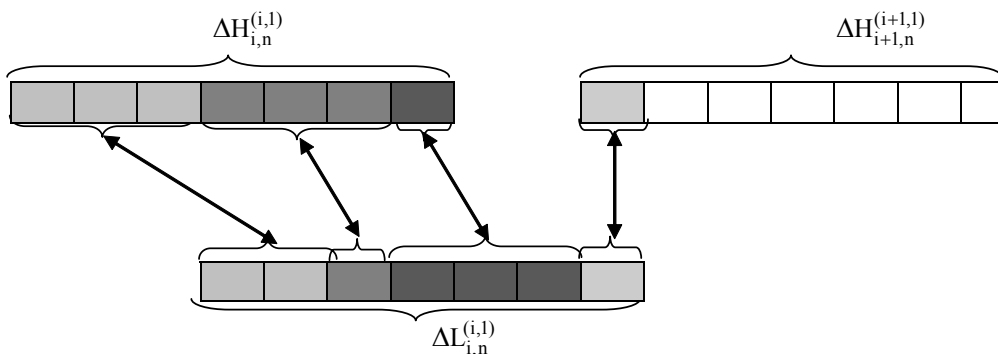


Рис. 2. Схема формирования двухкомпонентного кода для фрагмента изображения

Рассмотрим основные этапы технологии построения двухкомпонентного кода [6].

Первый этап заключается в формировании координатно-структурной и построочно-масштабных составляющих фрагмента изображения. Для этого осуществляется выявление апертур и построение массивов $\Delta H_{m,n}^{(v)}$ аппроксимирующих величин и $\Delta L_{m,n}^{(v)}$ длин апертур. Образование массивов $\Delta H_{m,n}^{(v)}$ и $\Delta L_{m,n}^{(v)}$ проводится в направлении строк, что позволяет выявить дополнительные структурные закономерности и обеспечить потенциальные возможности для устранения избыточности.

Второй этап. Определяются основания элементов массивов $\Delta H_{m,n}^{(v)}$ и $\Delta L_{m,n}^{(v)}$, рассматриваемых соответственно как адаптивное позиционное число с неравными соседними элементами и позиционное число в дифференциальном пространстве. Выполняются следующие действия:

1) для формирования системы оснований $W(h)$, $W(h) = \{w'(h)_i\}$, $i = \overline{1, m}$ элементов АПЧ с неравными соседними элементами

$$w'(h)_i = h_{i, \max} - h_{i, \min} + 1 - \text{sign}(j-1), \quad h_{i, \max} = \max_{1 \leq j \leq n} \{h_{i, j}\} + 1; \quad h_{i, \min} = \min_{1 \leq j \leq n} \{h_{i, j}\};$$

2) для системы оснований $W(\ell)$, $W(\ell) = \{w(\ell)_i\}$, $i = \overline{1, m}$ элементов ПЧДП

$$w(\ell)_{ij} = \ell_{i, \max} - \ell_{i, \min} + 1 = w(\ell)_i, \quad \ell_{i, \max} = \max_{1 \leq j \leq n} \{\ell_{i, j}\} + 1; \quad \ell_{i, \min} = \min_{1 \leq j \leq n} \{\ell_{i, j}\}.$$

Третий этап. Организуется оценка количества элементов $v(h, i)_\xi$ и $v(\ell)_\xi$ двухкомпонентных составляющих для построения обобщенного кода. Длина $D_{\text{нec}}$ кодового слова для построения текущего обобщенного кода считается заданной. По условию формирования ДК выбор первой составляющей на основе построения кода проводится для элементов одной строки массива $\Delta H_{m,n}^{(v)}$. Отсюда $v(h, i)_\xi = [D_{\text{нec}} / \log_2(w(h)_i - 1)]$. Вторая составляющая формируется на основе кодового описания элементов массива $\Delta L_{m,n}^{(v)}$, расположенных в общем случае на разных строках. Поэтому величина $v(\ell)_\xi$ определяется в несколько этапов. Суммарное количество элементов второй составляющей равно

$$v(\ell)_\xi = \sum_{i=\alpha}^{\alpha+\beta} v(\ell, i)_\xi,$$

где $v(\ell, i)_\xi = \min\{n_i; [\Delta D_{i-1} / (\log_2 w(\ell)_i + 1)]\}$; n_i – количество свободных элементов в i -й строке массива длин апертур.

В итоге получаем количество $v(h, i)_\xi$ элементов массива аппроксимирующих величин апертур и количество $v(\ell)_\xi$ элементов массивов длин апертур, участвующих в образовании двухкомпонентного кода.

Четвертый этап. Осуществляется построение ДК. Первая кодовая составляющая $E(h)_{i, \gamma+v(h, i)_\xi-1}^{(i, \gamma)}$, формируемая на основе $v(h, i)_\xi$ элементов строки массива аппроксимирующих величин, будет равна

$$E(h)_{i, \gamma+v(h, i)_\xi-1}^{(i, \gamma)} = \sum_{j=\gamma}^{\gamma+v(h, i)_\xi-1} (h_{i, j} - \text{sign}(1 - \text{sign}(h_{i, j-1} - h_{i, j}))) (w(h)_i - 1)^{v(h, i)_\xi + \gamma - 1 - j}. \quad (3)$$

Структура ДК на основе первой компоненты задается таким выражением:

$$E(h; \ell)_\xi = E(h)_{i, \gamma+v(h, i)_\xi-1}^{(i, \gamma)} \prod_{\phi=\alpha}^{\alpha+\beta} w(\ell)_\phi^{v(\ell, \phi)_\xi} + \Delta E(\ell)_{\alpha+\beta, \tau}^{(\alpha, \gamma)}, \quad \text{где } V(\ell)_{\alpha+\beta, \tau}^{(\alpha, \gamma)} = \prod_{\phi=\alpha}^{\alpha+\beta} w(\ell)_\phi^{v(\ell, \phi)_\xi}$$

– весовой коэффициент первой компоненты $E(h)_{i, \gamma+v(h, i)_\xi-1}^{(i, \gamma)}$ двухкомпонентного кода.

Схемотехническая структура кодирующего устройства для получения значения кода первой составляющей приведена на рис. 3.

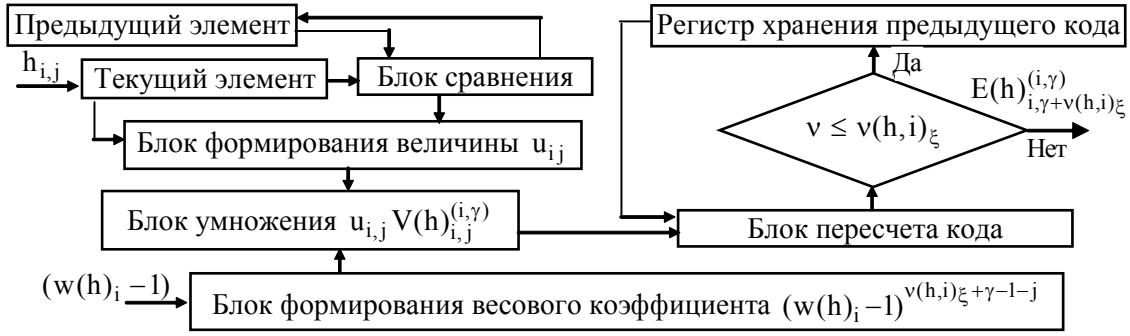


Рис. 3. Схематехническая реализация кодера

В блоке сравнения определяется максимальное значение из двух: предыдущего и текущего элементов массива аппроксимирующих величин апертур, после чего текущий элемент записывается в регистр предыдущего элемента для проведения сравнения с очередным элементом. Результат сравнения поступает на блок формирования величины $u_{i,j}$. Блок формирования $u_{i,j}$ (рис. 4) реализует $u_{i,j} = h_{i,j} - \text{sign}(1 - \text{sign}(h_{i,j-1} - h_{i,j}))$.

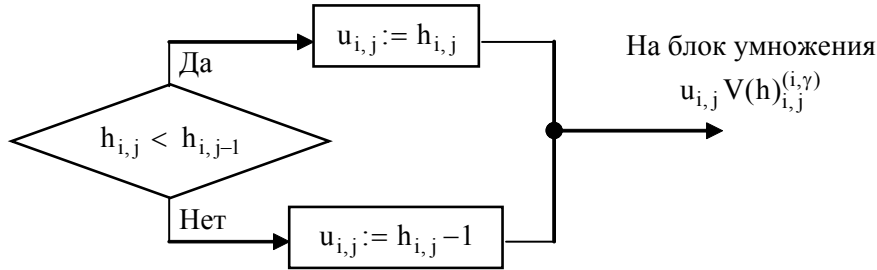


Рис. 4. Схематехническая реализация формирования величины $u_{i,j}$

В блоке умножения осуществляется вычисление произведения $u_{i,j} V(h)_{i,j}^{(i,\gamma)}$. При этом величина $V(h)_{i,j}^{(i,\gamma)}$ поступает с блока формирования весового коэффициента. Блок пересчета кода выполняет функцию рекуррентного вычисления соотношения, заданного формулой (3). Для этого на вход данного блока поступают значение произведения $u_{i,j} V(h)_{i,j}^{(i,\gamma)}$ (с блока умножения) и значение предыдущего кода-номера. В блоке сравнения осуществляется проверка условия на обработку последнего допустимого элемента массива аппроксимирующих величин апертур при формировании текущего кода. Если обработан последний элемент массива, то текущее значение кода является начальным значением двухкомпонентного кода, и происходит переход на обработку элементов массива длин апертур.

Пятый этап. Вычисление второй кодовой составляющей $\Delta E(\ell)_{\alpha+\beta,\tau}^{(\alpha,\gamma)}$ при условии рассмотрения массива длин апертур как позиционного числа в дифференциальном пространстве с учетом распределения элементов по строкам осуществляется следующим образом:

$$\Delta E(\ell)_{\alpha+\beta,\tau}^{(\alpha,\gamma)} = \sum_{j=\gamma}^n (\ell_{\alpha,j} - \ell_{\alpha,\min}) w(\ell)_\alpha^{n-j} \prod_{\chi=\alpha+1}^{\alpha+\beta-1} w(\ell)_\chi^{v(\ell,\chi)_\xi} w(\ell)_{\alpha+\beta}^{v(\ell,\alpha+\beta)_\xi} + \sum_{i=\alpha+1}^{\alpha+\beta-1} \sum_{j=1}^n (\ell_{i,j} - \ell_{i,\min}) \times \\ \times w(\ell)_i^{n-j} \prod_{\chi=i+1}^{\alpha+\beta-1} w(\ell)_\chi^{v(\ell,\chi)_\xi} w(\ell)_{\alpha+\beta}^{v(\ell,\alpha+\beta)_\xi} + \sum_{j=1}^{\tau} (\ell_{\alpha+\beta,j} - \ell_{\alpha+\beta,\min}) w(\ell)_{\alpha+\beta}^{\tau-j}.$$

Значение дополнительного кода $\Delta E(\ell)_{\alpha+\beta,\tau}^{(\alpha,\gamma)}$ в случае его рекуррентного вычисления разделяется на три составляющих, а именно: $\Delta E(\ell)_{\alpha,n}^{(\alpha,\gamma)}$ – долевой код для допустимых

элементов α -й строки массива длин апертур; $\Delta E(\ell)_{\alpha+\beta-1,n}^{(\alpha+1,1)}$ – долевого код для элементов полных строк, начиная с $(\alpha + 1)$ -й строки и заканчивая $(\alpha + \beta - 1)$ -й строкой; $\Delta E(\ell)_{\alpha+\beta,\tau}^{(\alpha+\beta,1)}$ – долевого код на основе допустимых элементов $(\alpha + \beta)$ -й строки.

Значит, на основе полученных выражений осуществляется формирование двухкомпонентного кода на базе неравнозначного вклада элементов массива аппроксимирующих величин апертур и элементов массива длин апертур.

Для сокращения временных затрат на сжатие изображения предлагается совмещать во времени некоторые уровни обработки, т.е. организовывать конвейерную обработку. При этом особенностями данного метода является то, что для некоторых уровней переход на очередной уровень начнется только после завершения предыдущего. Такая ситуация возникает на этапе вычисления двухкомпонентного кода, когда его формирование начинается только после того, как получены все строки массива аппроксимирующих величин и массива длин апертур, т.е. реализуется иерархический вид обработки (это необходимо для проведения преобразования динамических диапазонов массивов, которое начинается после образования всего массива). Значит, учитывая особенности разработанного метода сжатия и необходимость снижения времени компрессии предлагаем организовывать иерархически – конвейерный порядок выполнения кодирующих действий. Для разработанного метода иерархически – конвейерная организация сжатия заключается в том, что $\Delta L_{m,n}^{(v)}$ образуются одновременно с формированием массивов $\Delta L_{m,n}^{(v)}$ (конвейерная обработка). Это облегчается тем, что данные массивы имеют одинаковые размеры. Переход на этап формирования начального значения ДК осуществляется после завершения формирования массивов $\Delta L_{m,n}^{(v)}$ и $\Delta H_{m,n}^{(v)}$ (иерархическая обработка). Начальные значения ДК вычисляются одновременно на базе нескольких совокупностей элементов массива аппроксимирующих величин апертур. По мере получения начальных значений ДК осуществляется проверка условия на возможность и вычисление добавочного значения второй кодовой компоненты на основе элементов массива длин апертур. Процесс вычисления второй составляющей ДК проводится одновременно с вычислением остальных начальных значений единого кода (конвейерная обработка).

4. Выводы

1. Разработана информационная технология сжатия изображений на основе обобщенного кодирования его координатно-структурной и построчно-масштабирующей составляющих, базирующаяся на следующих составляющих:

– формировании массивов аппроксимирующих величин и массивов длин координатно-структурной и построчно-масштабирующей апертурных составляющих фрагмента изображения;

– вычислении оснований элементов массивов аппроксимирующих величин апертур и массивов длин апертур, рассматриваемых соответственно как адаптивное позиционное число с неравными соседними элементами и позиционное число в дифференциальном пространстве;

– нахождении позиций и количества элементов для массивов аппроксимирующих величин апертур и массивов длин апертур, участвующих в построении двухкомпонентных составляющих кода;

– построении двухкомпонентного кода на основе первой кодовой составляющей, формируемой на основе элементов строки массива аппроксимирующих величин;

– вычислении второй кодовой составляющей с учетом рассмотрения массива длин апертур как позиционного числа в дифференциальном пространстве.

2. Сжатие изображений достигается за счет:

1) исключения статистической избыточности, обусловленной учетом корреляционных зависимостей между элементами локальных областей в результате построения неравномерной координатно-структурной составляющей фрагмента изображения;

2) снижения психовизуальной избыточности путем аппроксимации последовательности элементов в составе апертуры одной усредненной величиной. Количество снижаемой психовизуальной избыточности регулируется в результате выбора высоты апертуры;

3) сокращения структурной избыточности, обусловленной: выявлением длин когерентных областей; выявлением структурно-комбинаторных закономерностей в координатно-структурной и построчно-масштабирующих формах фрагмента изображения.

Список литературы: 1. *Лабутина И.А.* Дешифрование аэрокосмических снимков: Учебное пособие. М: Аспект-Пресс, 2004. 184 с. 2. *Кашкин В.Б.* Цифровая обработка аэрокосмических изображений: Конспект лекций. Красноярск : ИПК СФУ, 2008. 121 с. 3. *Баранник В.В.* Методологический анализ системы аэрокосмического видеомониторинга чрезвычайных ситуаций / В.В. Баранник, А.В. Яковенко, А.Ю. Школьник // Сучасна спеціальна техніка, К.: №4(27). 2011. С. 12 – 22. 4. *Сэломон Д.* Сжатие данных, изображений и звука / Д. Сэломон. М: Техносфера, 2004. 368 с. 5. *Баранник В.В.* Структурно-комбинаторное представление данных в АСУ / В.В. Баранник, Ю.В. Стасев, Н.А. Королева. Х.: ХУПС, 2009. 252 с. 6. *Баранник В.В.* Информационная модель построчно-масштабирующих составляющих фрагмента изображения / В.В. Баранник, А.Ю. Школьник, Н.А. Королева // Системи обробки інформації. Х.: ХУПС. 2011. Вип. 4. С. 55 – 59.

Поступила в редколлегию 14.10.2011

Баранник Владимир Викторович, д-р техн. наук, профессор, ведущий научный сотрудник научного центра Харьковского университета Воздушных Сил им. Ивана Кожедуба. Научные интересы: системы, технологии преобразования, кодирования, защиты и передачи информации, семантической обработки изображений. Адрес: Украина, 61023, Харьков, ул. Сумская, 77/79, тел. 8 050-3038971.

Додух Александр Николаевич, инженер Харьковского университета Воздушных Сил имени Ивана Кожедуба. Научные интересы: системы, технологии преобразования, кодирования, защиты и передачи информации, семантической обработки изображений. Адрес: Украина, 61023, Харьков, ул. Сумская, 77/79.

**СРАВНИТЕЛЬНЫЙ МУЛЬТИФРАКТАЛЬНЫЙ АНАЛИЗ
ВРЕМЕННЫХ РЯДОВ МЕТОДАМИ ДЕТРЕНДИРОВАННОГО
ФЛУКТУАЦИОННОГО АНАЛИЗА И МАКСИМУМОВ МОДУЛЕЙ
ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ**

Описываются результаты исследования выборочных мультифрактальных характеристик, полученных методами детрендированного флуктуационного анализа и максимумов модулей вейвлет-преобразования для реализаций самоподобных и мультифрактальных процессов небольшой длины. Предлагается статистический критерий для принятия гипотезы о наличии монофрактальных свойств временного ряда. Проводится фрактальный анализ показаний крутильных весов, который выявил наличие мультифрактальных свойств.

Введение

В настоящее время стало общепризнанным, что многие информационные, биологические, физические, технологические процессы обладают сложной фрактальной структурой. Фрактальный анализ используется для моделирования, анализа и контроля сложных систем в различных областях науки и техники. В частности, в геологии – для прогнозирования сейсмической активности и цунами, определения возраста геологических пород; в биологии и медицине – для диагностики заболеваний и физиологического состояния по записям ЭКГ и ЭЭГ, при исследовании мутаций и изменений на генетическом уровне; в экономике – для прогнозирования кризисных ситуаций и оценивания риска по финансовым рядам; в физике – для исследования турбулентности и термодинамических процессов. Этот список далеко не полон [1-4].

Процессы, обладающие фрактальными свойствами, можно разделить на две группы: самоподобные (монофрактальные) и мультифрактальные. Монофрактальные процессы являются однородными в том смысле, что имеют один показатель скейлинга: их скейлинговые характеристики остаются неизменными на любом диапазоне масштабов. Мультифрактальные процессы допускают разложение на участки с различными локальными масштабными свойствами и характеризуются спектром скейлинговых показателей.

Двумя самыми востребованными инструментами исследования мультифрактальной структуры временных рядов являются метод мультифрактального детрендированного флуктуационного анализа (МФДФА) [5,6], ориентированный на обработку рядов с трендовыми составляющими, и метод максимумов модулей непрерывного вейвлет-преобразования (ММВП) [6-8], базирующийся на вейвлет-анализе, который называют «математическим микроскопом» из-за способности сохранять хорошее разрешение на разных масштабах. Оба метода являются мощным инструментом статистической обработки нестационарных процессов.

В настоящее время проведен ряд исследований, направленных на изучение статистических свойств оценок фрактальных характеристик, полученных данными методами [5,8-10]. Однако в большинстве работ рассмотрены модельные реализации больших длин – более десятка тысяч значений. В этом случае выборочные характеристики достаточно близки к теоретическим. В то же время, временные ряды, полученные на практике, имеют значительно меньший диапазон значений.

Целью представленной работы является численное исследование применения методов МФДФА и ММВП на модельных реализациях самоподобных и мультифрактальных процессов небольшой длины и использование полученных результатов для анализа экспериментальных данных.

1. Характеристики самоподобных и мультифрактальных случайных процессов [4,6,11,12]

Самоподобие случайных процессов заключается в сохранении статистических характеристик при изменении масштаба времени. Стохастический процесс $X(t)$ является самоподобным с параметром H , если процесс $a^{-H}X(at)$ описывается теми же конечномерными законами распределений (Law), что и $X(t)$:

$$\text{Law}\{a^{-H}X(at)\} = \text{Law}\{X(t)\}, \quad \forall a > 0, t > 0. \quad (1)$$

Параметр H , $0 < H < 1$, называемый показателем Херста, представляет собой степень самоподобия. Наряду с этим свойством, показатель H характеризует меру долгосрочной зависимости стохастического процесса. Можно показать, положив в (1) $a = 1/t$, что для самоподобного процесса выполняется следующее равенство:

$$\text{Law}\{X(t)\} = \text{Law}\{(1/t)^{-H}X(1)\} = \text{Law}\{t^H X(1)\}. \quad (2)$$

Учитывая (2), моменты q -го порядка самоподобного случайного процесса можно выразить как

$$M\left[|X(t)|^q\right] = M\left[|t^H X(1)|^q\right] = t^{qH} M\left[|X(1)|^q\right] = C(q) \cdot t^{qH}, \quad (3)$$

где величина $C(q) = M\left[|X(1)|^q\right]$.

Для мультифрактальных процессов рассматривается более общее соотношение

$$\text{Law}\{X(at)\} = \text{Law}\{M(a) \cdot X(t)\},$$

где $M(a)$ – независимая от $X(t)$ случайная функция. При самоподобном процессе $M(a) = a^H$. Мультифрактальные процессы проявляют более гибкие скейлинговые закономерности для моментных характеристик:

$$M\left[|X(t)|^q\right] = c(q) \cdot t^{qh(q)}, \quad (4)$$

здесь $c(q)$ – некоторая детерминированная функция; $h(q)$ – обобщенный показатель Херста, являющийся в общем случае нелинейной функций, для которой значение $h(q)$ при $q = 2$ совпадает со значением степени самоподобия H . Для монофрактальных процессов обобщенный показатель Херста не зависит от параметра q : $h(q) = H$.

В общем случае мультифрактальное множество характеризуется скейлинговой экспонентой $\tau(q)$, где параметр $q \in \mathbb{R}$. Функция $\tau(q)$ показывает, насколько неоднородным является исследуемое множество точек. Обобщенный показатель Херста связан с функцией $\tau(q)$ соотношением: $\tau(q) = qh(q) - 1$. Для самоподобного процесса функция $\tau(q)$ линейна.

Еще одной важной характеристикой мультифрактальных свойств является функция мультифрактального спектра $f(\alpha)$. Величина α_i представляет собой показатель сингулярности в точке x_i . Для самоподобного процесса все α_i одинаковы и равны параметру самоподобия H . Функция мультифрактального спектра $f(\alpha)$ характеризует распределение вероятностей различных значений α_i . Формально переход от переменных $\{q, \tau(q)\}$ к переменным $\{\alpha, f(\alpha)\}$ может быть осуществлен при помощи преобразования Лежандра:

$$\left\{ \alpha = \frac{d\tau}{dq}; f(\alpha) = q \frac{d\tau}{dq} - \tau \right\}.$$

2. Метод мультифрактального детрендрованного флуктуационного анализа [5,6,13]

Метод детрендрованного флуктуационного анализа используется при оценивании параметра H для самоподобных временных рядов. В этом случае для исходного временного ряда $x(t)$ строится кумулятивный ряд $y(t) = \sum_{i=1}^t x(i)$, который разбивается на N сегментов длиной s . Для каждого сегмента $y(t)$ вычисляется флуктуационная функция

$$F^2(s) = \frac{1}{s} \sum_{t=1}^s (y(t) - Y_m(t))^2, \quad (5)$$

где $Y_m(t)$ – локальный m -полиномиальный тренд в пределах данного сегмента (рис.1). Функция $F(s)$ усредняется по всему ряду $y(t)$. Такие вычисления повторяются для различных размеров сегментов, чтобы получить зависимость $F(s)$ в широком диапазоне значений параметра s . Для процессов с фрактальными свойствами с ростом s функция $F(s)$ также возрастает, и линейная зависимость $\log F(s)$ от $\log s$ свидетельствует о наличии свойства масштабной инвариантности: $F(s) \propto s^H$.

При исследовании свойств мультифрактальных процессов применяется мультифрактальный флуктуационный анализ. При проведении МФДФА исследуется зависимость флуктуационной функции $F_q(s)$ от параметра q :

$$F_q(s) = \left\{ \frac{1}{N} \sum_{i=1}^N [F^2(s)]^{\frac{q}{2}} \right\}^{\frac{1}{q}}, \quad (6)$$

полученной возведением выражения (5) в степень q и последующим усреднением по всем сегментам.

Изменяя временную шкалу s при фиксированном показателе q , находим зависимость $F_q(s)$, представляя её в двойных логарифмических координатах. Если исследуемый ряд сводится к мультифрактальному множеству, проявляющему долгосрочные зависимости, то флуктуационная функция $F_q(s)$ представляется степенной зависимостью $F_q(s) \propto s^{h(q)}$ с функцией обобщенного показателя Херста $h(q)$. Из определений (5) и (6) следует, что при $q = 2$ этот показатель сводится к обычному значению H . Для временных рядов, которые отвечают монофрактальному множеству, флуктуационная функция $F_q(s)$ одинакова для всех сегментов, и обобщенный показатель Херста $h(q) = H$ не зависит от параметра q .

3. Метод максимумов модулей вейвлет-преобразования [6-8, 12,14]

Метод максимумов модулей вейвлет-преобразования базируется на математическом аппарате вейвлет-анализа. Непрерывное вейвлет-преобразование функции $X(t)$ имеет вид

$$W(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} X(t) \psi_{ab}(t) dt,$$

где $\psi_{ab}(t)$ – вейвлет-функция с параметрами масштаба a и сдвига b . Функция $W(a, b)$ называется вейвлет-спектром и может быть представлена как поверхность вейвлет-коэффициентов в трехмерном пространстве. Наиболее важная информация содержится в линиях локальных экстремумов поверхности $W(a, x)$, поиск которых проводится на каждом масштабе a (рис.2).

Метод ММВП позволяет численно получить статистическую сумму:

$$Z(q, a) = \sum_{l \in L(a)} \left(\sup_{a' \leq a} |W(a', x_1(a'))| \right)^q,$$

где $L(a)$ – множество всех линий l максимумов модулей вейвлет-коэффициентов на масштабе a ; $x_1(a)$ – расположение максимума на этом масштабе. Для вычисления $Z(q, a)$ выбирается максимальное значение модуля вейвлет-коэффициентов вдоль каждой линии на масштабах, меньших заданного значения масштаба a . В этом случае выполняется зависимость $Z(q, a) \propto a^{\tau(q)}$, где $\tau(q)$ – скейлинговая экспонента, которую определяют для каждого значения q путем вычисления тангенса угла наклона $\ln Z(q, a)$ от $\ln a$.

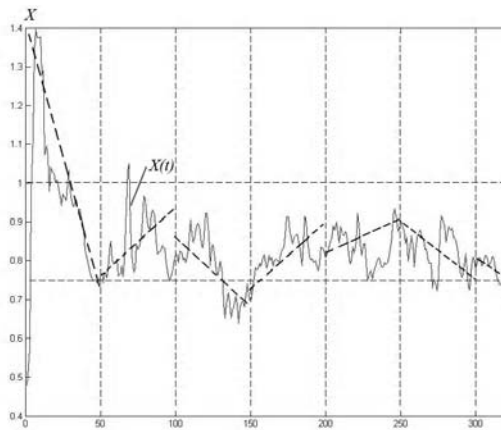


Рис. 1. Построение флуктационной функции

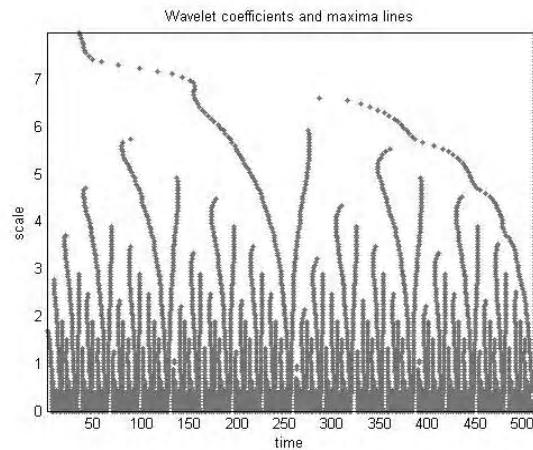


Рис. 2. Локальные максимумы вейвлет-преобразования

4. Исследование характеристик модельных фрактальных реализаций

В работе представлены результаты численного эксперимента, в ходе которого моделировались реализации разных типов фрактальных стохастических процессов: фрактального броуновского движения (монофрактальный процесс) и мультифрактальных детерминированных и стохастических каскадных процессов. Длина реализаций была выбрана равной 256, 512, 1024 и 2048 значений. Для каждого сгенерированного временного ряда методами МФДФА и ММВП рассчитывались мультифрактальные характеристики, которые потом усреднялись по множеству реализаций. Значения параметра q изменялись в диапазоне $-5 \leq q \leq 5$.

Фрактальное броуновское движение. Одной из наиболее известных и простых моделей самоподобного процесса является фрактальное броуновское движение (ФБД) [1,2]. Гауссовский процесс $X(t)$ называется фрактальным броуновским движением с параметром H , $0 < H < 1$, если приращения случайного процесса $\Delta X(\tau) = X(t + \tau) - X(t)$ имеют распределение вида

$$P(\Delta X < x) = \frac{1}{\sqrt{2\pi\sigma_0\tau^H}} \cdot \int_{-\infty}^x \text{Exp}\left[-\frac{z^2}{2\sigma_0^2\tau^{2H}}\right] dz,$$

где σ_0 – коэффициент диффузии. ФБД с параметром $H=0.5$ совпадает с классическим броуновским движением. На рис. 3 показаны реализации ФБД и соответствующие реализации приращений при значениях параметра $H=0.6$ (слева) и $H=0.8$ (справа) длиной 1024 значения.

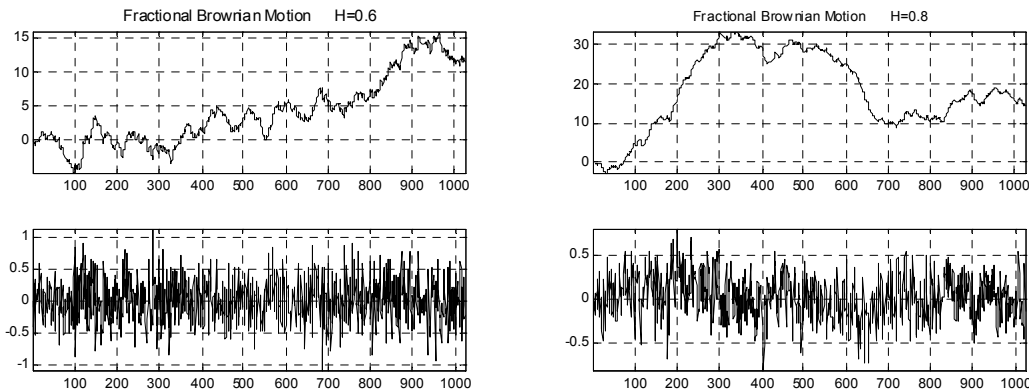


Рис. 3. Реализации ФБД (вверху) и соответствующих приращений (внизу)

ФБД является монофрактальным процессом, его скейлинговые характеристики полностью определяются значением показателя Херста H [4,5]:

$$h(q) = H; \quad \tau(q) = q^{H-1}; \quad \begin{cases} \alpha = H \\ f(\alpha) = 1 \end{cases}.$$

На рис. 4 представлены выборочные значения функции обобщенного показателя Херста, скейлинговой экспоненты и мультифрактального спектра, полученные по реализациям ФБД разной длины с параметром $H = 0.8$. В верхнем ряду рис.4 расположены характеристики, построенные методом МФДФА, в нижнем – методом ММВП. Пунктирной линией на графиках выборочных $h(q)$ и $\tau(q)$ показаны их теоретические значения. Для наглядности значения функции $\tau(q)$ приведены в увеличенном масштабе. Теоретическое значение мультифрактального спектра $f(\alpha)$ представляет собой точку $(0.8,1)$.

Очевидно, что оценки мультифрактальных характеристик, полученные методом МФДФА, значительно ближе к своим теоретическим значениям, чем оценки, построенные с применением ММВП. К тому же, при использовании метода ММВП построение вейвлет-спектра, линий локальных максимумов, выбор подходящего диапазона изменения масштаба вейвлет-преобразования представляет собой значительно более сложную задачу, чем определение степени полинома и диапазона длин сегментов для флуктуационного анализа.

С увеличением длины реализации выборочные характеристики стремятся к своим аналитическим значениям. Однако при небольшом числе значений во временном ряде, полученные оценки характеристики демонстрируют ложные мультифрактальные свойства. В работах [8,9,15] отмечается, что одним из недостатков оценивания мультифрактального спектра, особенно методом ММВП, является то, что оценивается верхняя огибающая истинного спектра. Это может приводить к неправильной интерпретации численных результатов, поскольку при истинном дискретном спектре $f(\alpha)$ полученная огибающая будет включать множество ложных точек. Поэтому в случае рядов небольшой длины возникает необходимость проведения дополнительных исследований для подтверждения наличия монофрактальных свойств.

Детерминированный биномиальный каскад [1,4,6]. Простейшей моделью мультифрактального процесса с заданными свойствами является биномиальный каскад. При его построении первоначальный единичный отрезок делится на два равных интервала, которым приписываются весовые коэффициенты p_1 и $p_2 = 1 - p_1$ соответственно. Затем с

каждым из интервалов прodelывается аналогичная процедура. В результате на втором шаге имеется четыре интервала с весами $p_1^2, p_1 p_2, p_2 p_1$ и p_2^2 . При числе шагов $n \rightarrow \infty$ и $p_1 \neq p_2$ мы приходим к неоднородному фрактальному множеству. Характеристики $h(q)$, $\tau(q)$ и $f(\alpha)$ для биномиального мультифрактального процесса зависят только от коэффициентов p_1 и p_2 и определяются аналитически следующим образом:

$$\left. \begin{aligned}
 h(q) &= \left(\frac{1}{q} - \frac{\ln(p_1^q + p_2^q)}{q \ln 2} \right); \\
 \tau(q) &= \frac{-\ln(p_1^q + p_2^q)}{\ln 2}
 \end{aligned} \right\} \begin{aligned}
 \alpha &= - \frac{p_1^q \ln(p_1) + p_2^q \ln(p_2)}{\ln 2 (p_1^q + p_2^q)} ; \\
 f(\alpha) &= - \frac{q}{\ln 2} \frac{p_1^q \ln(p_1) + p_2^q \ln(p_2)}{(p_1^q + p_2^q)} + \frac{\ln(p_1^q + p_2^q)}{\ln 2} .
 \end{aligned}$$

На рис.5 слева представлены теоретические скейлинговые экспоненты $\tau(q)$ для значений $p_1 = \{0.6, 0.7, 0.8, 0.9\}$, а справа показаны реализации биномиального каскада при значениях $p_1 = 0.6$ (вверху) и $p_1 = 0.8$ (внизу). Число итераций $n = 10$, т.е. длина реализации равна 2^{10} значений. Очевидно, что с увеличением первоначального весового коэффициента p_1 увеличивается неоднородность временного ряда.

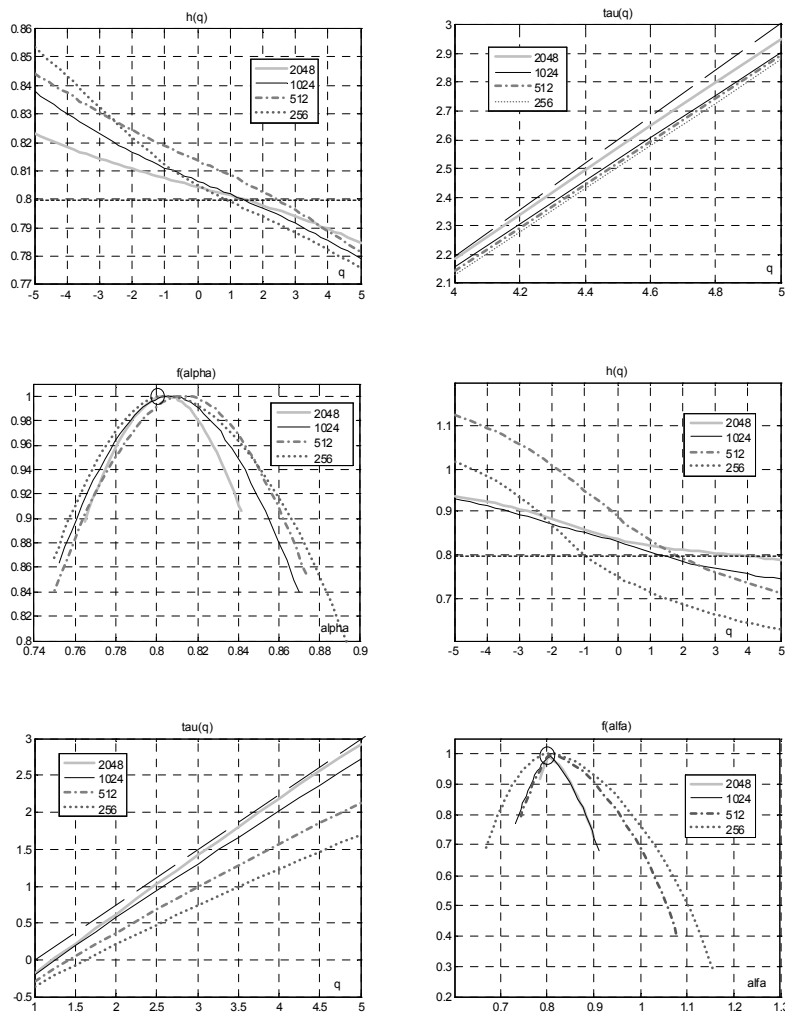


Рис. 4. Выборочные функции $h(q)$, $\tau(q)$, $f(\alpha)$ для реализаций ФБД, полученные методом МФДФА (вверху) и методом ММВП (внизу)

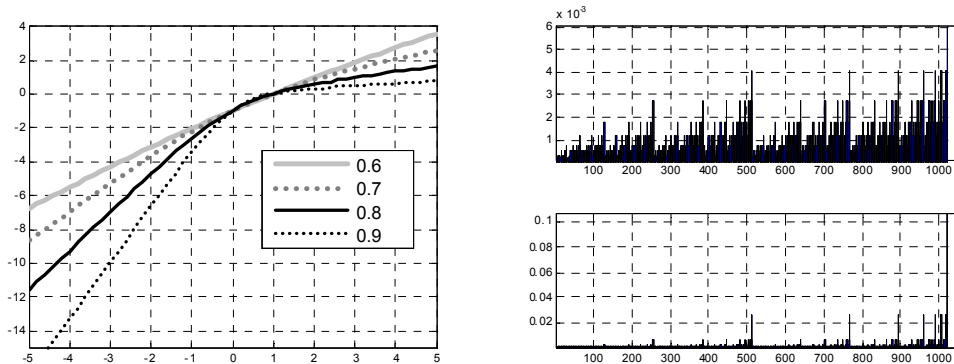


Рис. 5. Скейлинговые экспоненты $\tau(q)$ для разных p_1 и реализации каскада при $p_1 = 0.6$ и $p_1 = 0.8$

На рис. 6 приведены выборочные мультифрактальные характеристики $\tau(q)$ и $f(\alpha)$, полученные методами МФДФА и ММВП по реализациям биномиального каскада разной длины с коэффициентом $p_1 = 0.7$. Характеристики, построенные методом МФДФА, расположены в верхнем ряду, методом ММВП – в нижнем. Пунктирной линией на графиках показаны теоретические значения функций. Для наглядности значения функции $\tau(q)$ приведены только для $q > 0$.

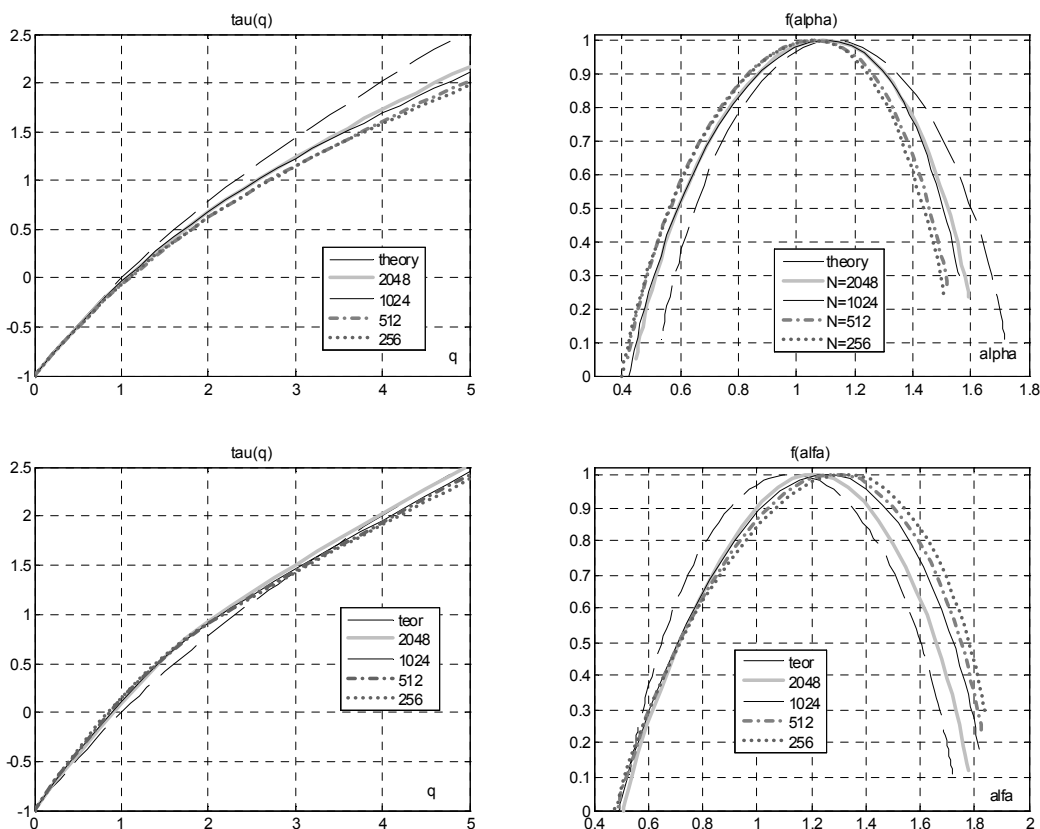


Рис. 6. Выборочные функции $\tau(q)$ и $f(\alpha)$ для реализаций биномиального каскада, полученные методом МФДФА (вверху) и методом ММВП (внизу)

При оценивании рядов с мультифрактальными свойствами метод ММВП позволяет получить оценки, сопоставимые по своим статистическим характеристикам с оценками метода МФДФА. Однако стоит отметить, что характеристики, полученные на основе метода ММВП, очень чувствительны к изменению масштаба диапазона вейвлет-преобразования, в то время как метод флуктуационного анализа позволяет получать устойчивые

оценки. При небольшом числе значений во временном ряде его выборочные фрактальные характеристики демонстрируют смещение диапазона значений. В обоих случаях с увеличением длины реализации выборочные характеристики стремятся к своим аналитическим значениям.

Стохастический биномиальный каскад. Реализации детерминированного каскада полностью определяются величиной p_1 , что неприемлемо для моделирования случайных процессов. При построении стохастических каскадов весовыми коэффициентами являются независимые значения некоторой заданной случайной величины W [2,4,5]. Она выбирается таким образом, чтобы математическое ожидание суммы весовых коэффициентов на каждой итерации равнялось единице. Если выбрать случайную величину, определенную на интервале $[0,1]$, то сумма коэффициентов на каждой итерации будет точно равной единице. В этом случае первым двум интервалам будут приписаны веса w_1 и $1-w_1$ соответственно. На втором шаге добавляются два новых независимых случайных значения w_2 и w_3 и образуются четыре интервала с весами w_1w_2 , $w_1(1-w_2)$, $(1-w_1)w_3$ и $(1-w_1)(1-w_3)$. При числе шагов $n \rightarrow \infty$ мы приходим к предельной мере, являющейся неоднородным фрактальным множеством.

В данной работе предложено в качестве случайной величины, порождающей весовые коэффициенты, использовать случайную величину, имеющую бета-распределение с плотностью вероятностей $p(x) = \frac{1}{B(a,b)}(1-x)^{b-1}$, где $B(a,b) = \int_0^1 x^{a-1}(1-x)^{b-1} dx$ – бета-функция, определенную на интервале $[0,1]$. Для бета-распределения с одинаковыми значениями параметров $a=b$ можно аналитически определить скейлинговую экспоненту $\tau(q)$ на интервале значений $q > -1$ [4]:

$$\tau(q) = -\log_2 \frac{\text{Beta}(\alpha+q, \alpha)}{\text{Beta}(\alpha, \alpha)} - 1.$$

На рис. 7 слева представлены графики теоретических скейлинговых экспонент $\tau(q)$ для значений параметра $a = \{0.5, 1, 1.5, 3\}$. Очевидно, что с увеличением значения параметра a происходит ослабление мультифрактальных свойств процесса. На рис. 7 справа показаны реализации биномиальных стохастических каскадов для значений $a=1$ (вверху) и $a=3$ (внизу).

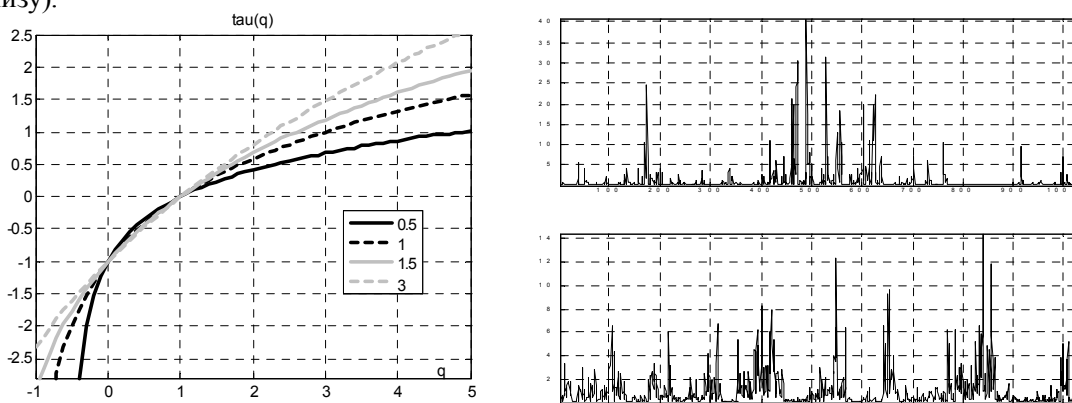


Рис. 7. Функция $\tau(q)$ для разных значений параметра a бета-распределения; реализации биномиального каскада для значений $a=1$ и $a=3$

В работе был выполнен мультифрактальный анализ реализаций мультипликативных стохастических бета-каскадов с различной степенью неоднородности, т.е. мультифрактальных свойств. Для сравнения с аналитическими характеристиками вычисления проводились в диапазоне положительных значений параметра q . На рис. 8 представлены мультифрактальные характеристики, полученные обоими методами по реализациям каскада, веса которого имеют равномерное распределение ($a=1$) разной длины.

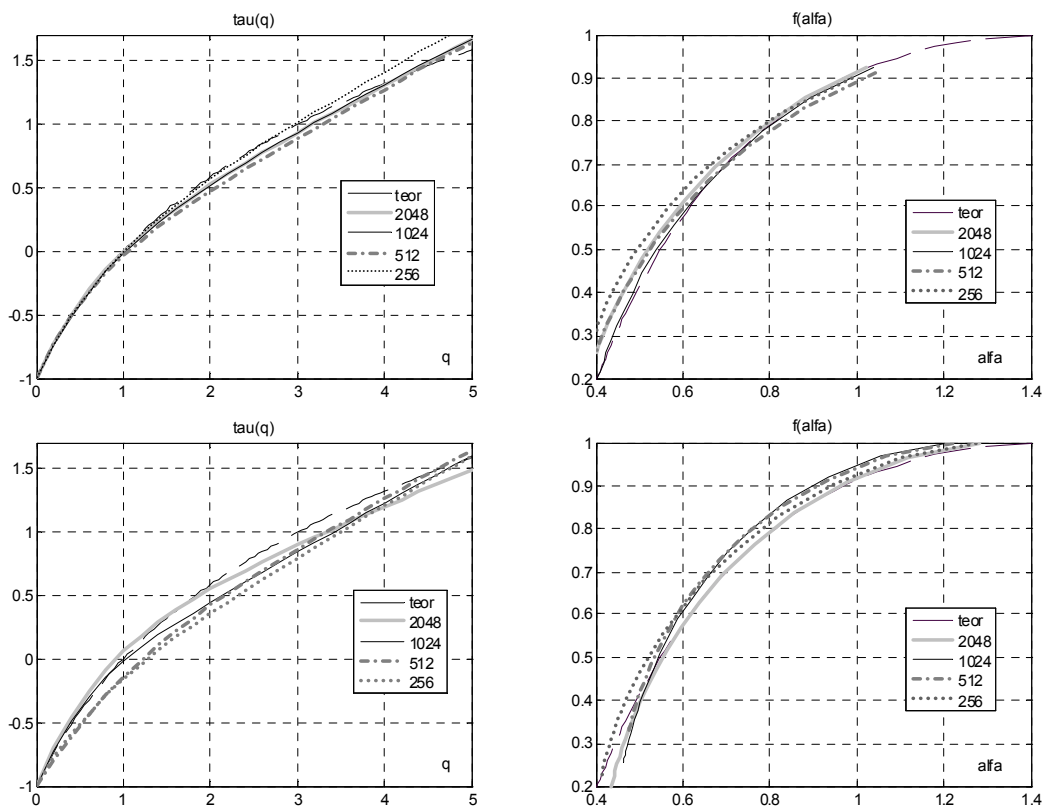


Рис. 8. Выборочные функции $\tau(q)$ и $f(\alpha)$ для реализаций стохастического каскада, полученные методом МФДФА (вверху) и методом ММВП (внизу)

Результаты численных исследований показывают, что характеристики, полученные методом ММВП, менее устойчивые и имеют больший разброс значений. Однако изучение вейвлет-спектра и линий локальных максимумов позволяет сделать дополнительные выводы о локальных особенностях структуры ряда и улучшить выборочные характеристики. Таким образом, при исследовании неизвестного временного ряда метод МФДФА должен быть использован в первую очередь, а затем мультифрактальная структура может быть исследована более детально методом ММВП.

5. Проверка гипотезы о монофрактальных свойствах временного ряда

Приведенные выше результаты мультифрактального анализа модельных временных рядов демонстрируют необходимость разработки инструмента, позволяющего различать моно- и мультифрактальные реализации. В данной работе предлагается метод, позволяющий принимать или отвергать гипотезу о наличии монофрактальных свойств ряда на основе исследования выборочных значений обобщенного показателя Херста, полученных методом МФДФА.

Для самоподобных процессов вопрос о законе распределения оценок H рассмотрен в ряде работ, где было показано, численно или аналитически, что оценки являются нормальными случайными величинами [1,12,16]. В представленной работе исследованы законы распределения оценок $h(q)$ при разных значениях параметра q . Проведенный анализ выборочных законов распределения $h(q)$ показал, что при $q > 0$ оценки $h(q)$ имеют нормальное распределение, параметры которого зависят от значения q . Для $q < 0$ выборочные значения обобщенного показателя Херста в общем случае не являются нормально распределенными. На рис.9 представлены значения функции $h(q)$ ($0 \leq q \leq 5$) и гистограммы значений оценок $h(q)$, полученные по реализациям длиной 1024 значения для $q = 1$ и $q = 5$. Вверху приведены результаты для реализаций ФБД с параметром Херста $H = 0.8$, внизу – для реализаций биномиального стохастического бета-каскада при значении $a = 1$.

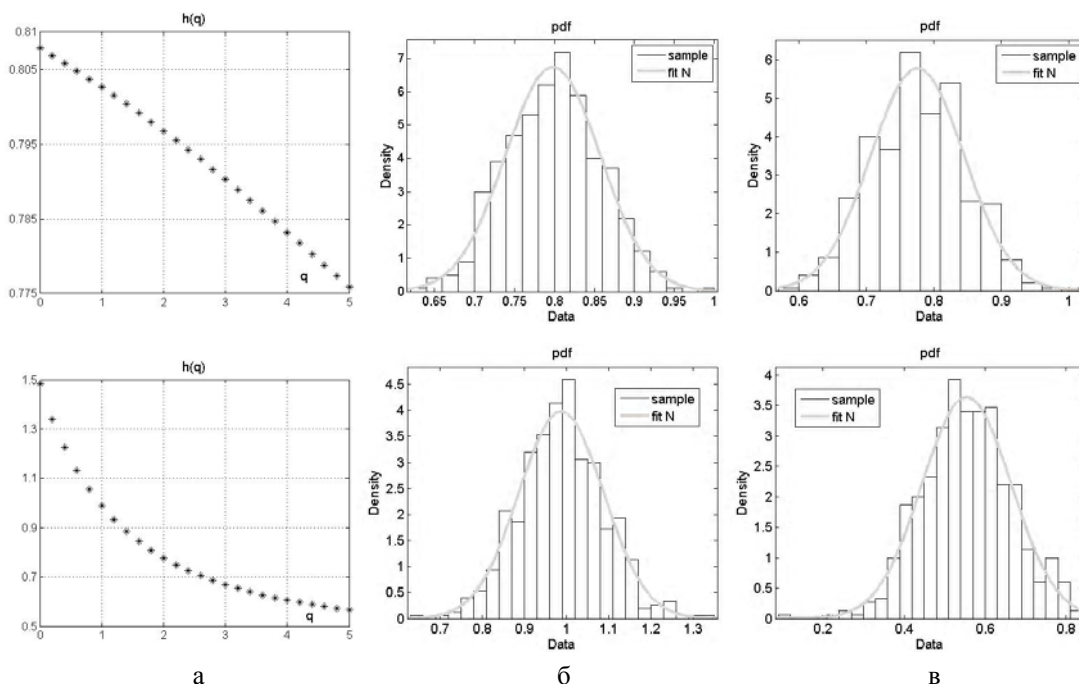


Рис. 9. Выборочные функции $h(q)$ (а); выборочные плотности распределений $h(q)$ при $q = 1$ (б) и $q = 5$ (в) для реализаций ФБД и реализаций стохастического каскада

Очевидно, что мультифрактальные процессы имеют гораздо больший перепад значений $\Delta h = h(q_1) - h(q_2)$. Случайная величина Δh при $q > 0$ имеет нормальное распределение $N(m_h, s_h)$, параметры которого зависят от длины реализации и выбранных значений q . Предложенный в работе критерий рассматривает $\Delta h = h(0.1) - h(5)$. Путем численного моделирования монофрактальных процессов с различной степенью самоподобия были получены выборочные значения m_h и S_h для рядов длины N , представленные в таблице. Там же приведены аналогичные характеристики, рассчитанные по реализациям каскадных процессов с разной степенью неоднородности, задаваемой параметром бета-распределения a .

Среднее значение и среднее квадратическое отклонение случайной величины Δh

	N=256		N=512		N=1024		N=2048	
	m_h	s_h	m_h	s_h	m_h	s_h	m_h	s_h
ФБД	0.075	0.052	0.05	0.038	0.035	0.026	0.03	0.022
Каскад								
a=1	0.87	0.23	0.86	0.18	0.85	0.16	0.82	0.12
a=5	0.32	0.13	0.29	0.095	0.28	0.08	0.27	0.07
a=10	0.23	0.1	0.2	0.07	0.18	0.06	0.17	0.05

Из таблицы видно, что даже при очень слабых мультифрактальных свойствах процесса, при $a=10$, можно различать моно- и мультифрактальные реализации. Таким образом, значение случайной величины Δh можно использовать в качестве статистического критерия для принятия или отбрасывания гипотезы о наличии монофрактальных свойств временных рядов. В данном случае нулевой гипотезой является предположение о монофрактальности ряда с некоторым заданным уровнем значимости α . После получения оценки функции $h(q)$ методом МДФА вычисляется наблюдаемая величина $\Delta h = h(0.1) - h(5)$. Гипотеза принимается с уровнем значимости α , если полученное значение попадает в область допустимых значений $m_h(N) \pm t_\alpha s_h(N)$, где N – длина реализации, m_h и S_h – соответствующие значения, рассчитанные для монофрактального процесса, t_α – значение функции Лапласа.

6. Исследование рядов показаний крутильных весов

Хорошо известно, что многие временные ряды, отвечающие природным явлениям, имеют самоподобную структуру. К таким явлениям относятся разливы рек, количество выпавших осадков, изменение температуры, увеличение диаметра колец деревьев [1-3]. В последние годы появился ряд работ, посвященных фрактальному анализу данных, связанных с солнечной активностью [17]. Одним из приборов, реагирующих на солнечную активность, являются сверхлегкие крутильные весы (КВ). С их помощью Ш. Кулон установил закон взаимодействия электрических зарядов, а Г. Кавендиш измерил величину мировой гравитационной постоянной. Крутильные весы относятся к числу исключительно чувствительных приборов и до настоящего времени применяются в ответственных экспериментах при изучении так называемых «сверхслабых взаимодействий» в биологии и связанных с нею дисциплинах, а также в физике для уточнения значения гравитационной постоянной.

В работах украинского радиоастронома А.Ф. Пугача [18] впервые были проведены длительные однородные наблюдения за поведением сверхлегких КВ, снабженных автоматической системой регистрации данных. Показания КВ регистрировались каждую минуту. Анализ наблюдаемых данных указывает на то, что суточная переменность показаний КВ не связана с изменениями температуры, давления, условиями освещенности, гравитационным воздействием Солнца. Однако КВ однозначно реагируют на солнечные и лунные затмения, на восход и заход Солнца, на покрытие планет Луной. На рис.10. представлены показания КВ за период с 13 по 23 ноября 2010 г.

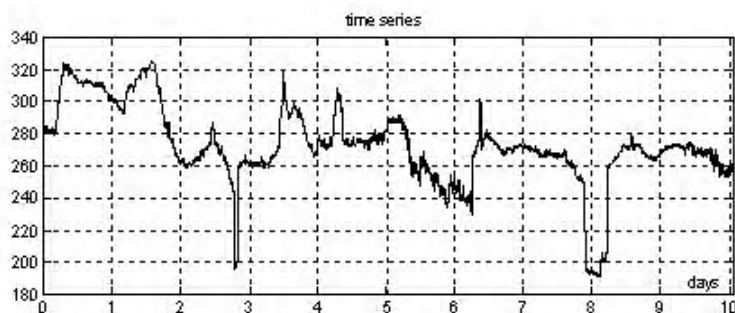


Рис. 10. Временной ряд показаний КВ

Исследование фрактальной структуры данных КВ имеет несомненный научный интерес. Особую важность представляет анализ наблюдений, полученных в течение одно-двух суток, что позволит отслеживать изменения фрактальной структуры в зависимости от изменения солнечной активности. В данной работе проведен анализ временных рядов, полученных в разные периоды года, который показал наличие явно выраженных мультифрактальных свойств. На рис.11 представлены мультифрактальные спектры, полученные методом МФДФА (слева) и ММВП (справа), построенные по рядам двухсуточных наблюдений (2880 значений), типичные для февраля, мая и октября.

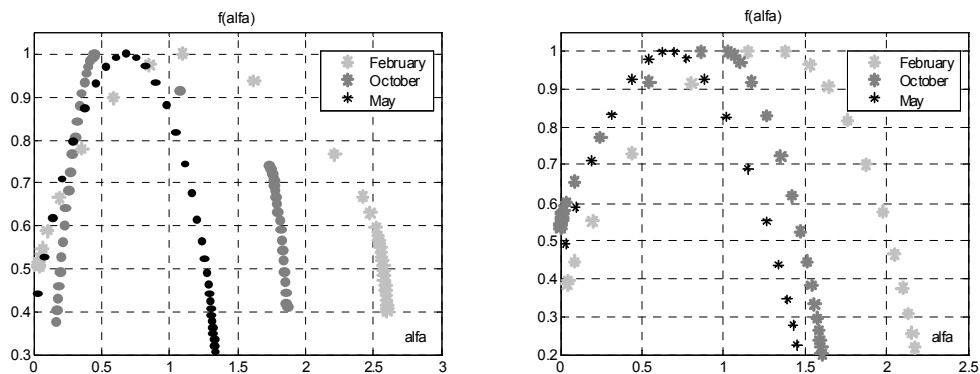


Рис. 11. Выборочные функции $f(\alpha)$ для показаний КВ, полученные методом МФДФА (слева) и ММВП (справа)

Результаты исследований показали, что мультифрактальная структура рядов показаний КВ изменяется в зависимости от времени года. Зависимость этой структуры от солнечной активности требует дальнейших исследований. Необходимо отметить, что в данном случае метод ММВП показывал более устойчивые результаты, чем метод флуктуационного анализа, что, очевидно, связано с наличием больших выбросов в наблюдениях.

Выводы

В работе представлены результаты исследований выборочных мультифрактальных характеристик, полученных методами мультифрактального флуктуационного анализа и максимумов модулей вейвлет-преобразования для реализаций самоподобных и мультифрактальных процессов небольшой длины. Результаты показывают, что оценки характеристик, полученные методом максимумов модулей вейвлет-преобразования, менее устойчивые и имеют больший разброс значений. Показано, что для реализаций с теоретическим монофрактальным спектром в выборочном спектре появляются дополнительные ложные значения, что может приводить к ошибочной интерпретации результатов. Впервые предложен статистический критерий для принятия гипотезы о наличии монофрактальных свойств временного ряда. Впервые проведен фрактальный анализ показаний сверхлегких крутильных весов, который выявил мультифрактальную структуру рядов, изменяющуюся в зависимости от времени года. Практическая значимость полученных результатов заключается в том, что рассмотренные методы могут быть использованы для анализа фрактальной структуры реализаций небольшой длины в технических, информационных, биологических и других системах.

Автор выражает большую благодарность А.Ф.Пугачу за предоставленный цифровой материал показаний крутильных весов и обсуждение результатов работы.

Список литературы: 1. *Федер Е.* Фракталы / Е. Федер. М.: Мир, 1991. 254 с. 2. *Мандельброт Б.* Фрактальная геометрия природы / Б. Мандельброт. Москва–Ижевск : Институт компьютерных исследований, 2002. 656 с. 3. *Шредер М.* Фракталы, хаос и степенные ряды. Миниатюры из бесконечного рая / М.Шредер. Ижевск: НИЦ «Регулярная и хаотическая динамика», 2005. 528 с. 4. *Riedi R.H.* Multifractal processes, in Doukhan P., Oppenheim G., Taqqu M.S. (Eds.), Long Range Dependence: Theory and Applications, p. 625–715, Birkhuser. 2002. 5. *Kantelhardt J.W.* Multifractal detrended fluctuation analysis of non-stationary time series / J.W. Kantelhardt, S.A. Zschiegner, A. Bunde, S. Havlin, E. Koscielny-Bunde, H.E. Stanley // Physica A. 2002. № 316. P. 87-114. 6. *Kantelhardt J.W.* Fractal and Multifractal Time Series. 2008 [Электронный ресурс]: <http://arxiv.org/abs/0804.0747> 7. *Muzy J.F.* Multifractal formalism for fractal signals: the structure-function approach versus the wavelet-transform modulus-maxima method / Muzy J.F., Vachy E., Arneodo A. // Phys. Rev. E. 1993. V.47. P.875-884. 8. *Павлов А.Н.* Мультифрактальный анализ сигналов / А.Н. Павлов, В.С. Анищенко // Известия Саратовского университета. Серия «Физика». 2007. Т. 7. Вып. 1. С.3-25. 9. *Oswiecimka P.* Wavelet versus detrended fluctuation analysis of multifractal structures / P. Oswiecimka, J. Kwapien, S.Drozd // Physical Review E: Statistical, Nonlinear, and Soft Matter Physics. - Vol. 74. 2006. P. 161-203. 10. *Олемской А.И.* Мультифрактальный анализ временных рядов / А.И. Олемской, В.Н. Борисюк, И.А. Шуда // Вісник СумДУ. Серія «Фізика, математика, механіка». 2008. №2. С.70-81. 11. *Божокин С.В.* Фракталы и мультифракталы / С.В. Божокин, Д. А. Паршин. Ижевск: НИЦ «Регулярная и хаотическая динамика». 2001. 128 с. 12. *Abry P., Flandrin P., Taqqu M.S., Veitch D.* Self-similarity and long-range dependence through the wavelet lens. in Doukhan P., Oppenheim G., Taqqu M.S. (Eds.), Long Range Dependence: Theory and Applications. Birkhduser: 2002. P. 125–160. 13. *Kantelhardt J.W.* Detecting long-range correlations with detrended fluctuation analysis / J.W. Kantelhardt, E. Koscielny-Bunde, H.N.A. Rego, S. Havlin, A. Bunde // Physica A. 2001. № 295. P. 441-454. 14. *Малла С.* Вэйвлеты в обработке сигналов / С. Малла. М.: Мир, 2005. 671 с. 15. *Veneziano D.* Multifractal analysis: pitfalls of standard procedures and alternatives / D. Veneziano, G.E. Moglen, R.L. Bras // Phys. Rev. E. 1995. V.52. P.1387-1398. 16. *Петерс Э.* Фрактальный анализ финансовых рынков: Применение теории хаоса в инвестициях и экономике / Э.Петерс. М: Интернет-трейдинг, 2004. 304 с. 17. *Abramenko V. I.* Multifractal Nature of Solar Phenomena / Solar Physics Research Trends, Edited by P. Wang. -Nova Publisher: 2008. P. 95-136. 18. *Пугач А. Ф.* Наблюдения астрономических феноменов с помощью крутильных весов / А.Ф. Пугач // Physics of consciousness and life, cosmology and astrophysics. 2009. №2. С.30-51.

Поступила в редколлегию 13.10.2011

Кириченко Людмила Олеговна, канд. техн. наук, доцент кафедры прикладной математики ХНУРЭ. Научные интересы: самоподобные и мультифрактальные случайные процессы, фрактальный и мультифрактальный анализ временных рядов. Адрес: Украина, 61166, Харьков, просп. Ленина, 14, тел. 70-21-640. E-mail: ludmila@kture.kharkov.ua.

АНАЛИЗ МЕТОДОВ РАСПОЗНАВАНИЯ И ПРЕОБРАЗОВАНИЯ АУДИОИНФОРМАЦИИ В НОТЫ

Предлагаются методы решения задачи распознавания и преобразования аудиозаписи в ноты с помощью создания универсального программного решения. В качестве метода решения задачи распознавания аудиоинформации рассматривается возможность применения численных методов (преобразование Фурье) и методов искусственного интеллекта.

1. Введение

Идентификацией объекта является определение его характеристик на основе опытного исследования, она самая трудоемкая и самая ответственная операция при анализе объектов. К настоящему времени теория идентификации превратилась в обширное по содержанию и богатое по методам учение, тем не менее, еще многие актуальные проблемы ждут в ней своего решения. Одной из самых актуальных задач в настоящее время является задача расширения класса объектов, поддающихся эффективной идентификации [1].

Проблема поиска музыки становится актуальной в современном мире. Большое количество музыки в различных форматах существует в электронном виде, и одной из трудноразрешимых проблем является преобразование этих файлов в нотный ряд.

Цель данного исследования – разработка методов распознавания аудиоинформации и дальнейшего ее преобразования в ноты. Для достижения поставленной цели необходимо разработать программное обеспечение, позволяющее решать следующие задачи[3]:

- а) поиск аудиоинформации по фрагменту аудио;
- б) поиск аудиоинформации по описанию, её каталогизация и категоризация аудио в поисковых целях;
- в) поиск аудиоинформации по «напетой» мелодии;
- г) распознавание аудиоинформации и получение нот.

Для реализации первого пункта на данный момент уже существуют определённые программные средства. Их работа основана на том, что в базе данных хранится большое количество аудиозаписей. Однако каждое из этих программных приложений имеет свои, весьма существенные недостатки. Предлагается использовать новый метод, который одновременно предоставит возможность решения всех описанных выше проблем: аудиозапись распознаётся и переводится в ноты (технически это близко к хранению аудио в миди-формате). Это предоставит возможность сравнивать аудиозапись не только с её фрагментом для поиска, но также и различных вариаций, вплоть до напетого кусочка в микрофон. Таким образом, решится глобальная проблема многих музыкантов: автоматическое получение нот из записей.

Практическая значимость решения задачи распознавания аудиоинформации и преобразования в нотный ряд заключается в:

- автоматизации ручного процесса снятия нот, т.е. перевода из аудио в ноты;
- создании потенциальной базы для разработки “музыкального поисковика”.

2. Постановка задачи

Для решения задачи распознавания и преобразования аудиоинформации в ноты необходимо создать универсальный программный продукт, базирующийся на численных методах (преобразование Фурье) и использующий методы искусственного интеллекта (многослойный перцептрон) [2].

Такой программный продукт должен выполнить следующие действия:

1. Считать звуковой файл.
2. На основе совокупности методов проанализировать его.
3. Получить на выходе нотное представление записи.
4. Дальнейшие действия зависят от конкретного применения.

При более подробном рассмотрении процесса распознавания и преобразования аудиоинформации можно выделить следующие этапы исследования:

1. Открытие звукового файла и считывание информации (частота дискретизации, длительность).
2. Спектральный анализ некоего «минимального» фрагмента звука при помощи преобразования Фурье.
3. По наиболее сильным гармоникам определяем, какие ноты присутствуют в аккорде (метод обертонового ряда)[3].
4. По гармоникам, присутствующим в спектре каждой ноты, можем определить, какие тембры присутствуют.
5. Распознавание в динамике – на каждом интервале. Если картина не изменилась для какой-то ноты, мы соответственно предполагаем, что нота длится дольше, и не разрываем её.
6. Оформление результатов.

3. Анализ существующих методов и программных продуктов, реализующих проблему распознавания и преобразования аудиоинформации

На данный момент существует несколько программных решений подобной направленности. Вот некоторые из них:

Amazing Midi – программа для распознавания аудио и преобразования в midi (рис. 1).

Вначале пользователю предлагается выбрать аудиофайл с образцом тембра музыкального инструмента, который звучит в записи, входной и выходной файлы. Далее находится поле, в котором будут отображаться распознанные ноты.

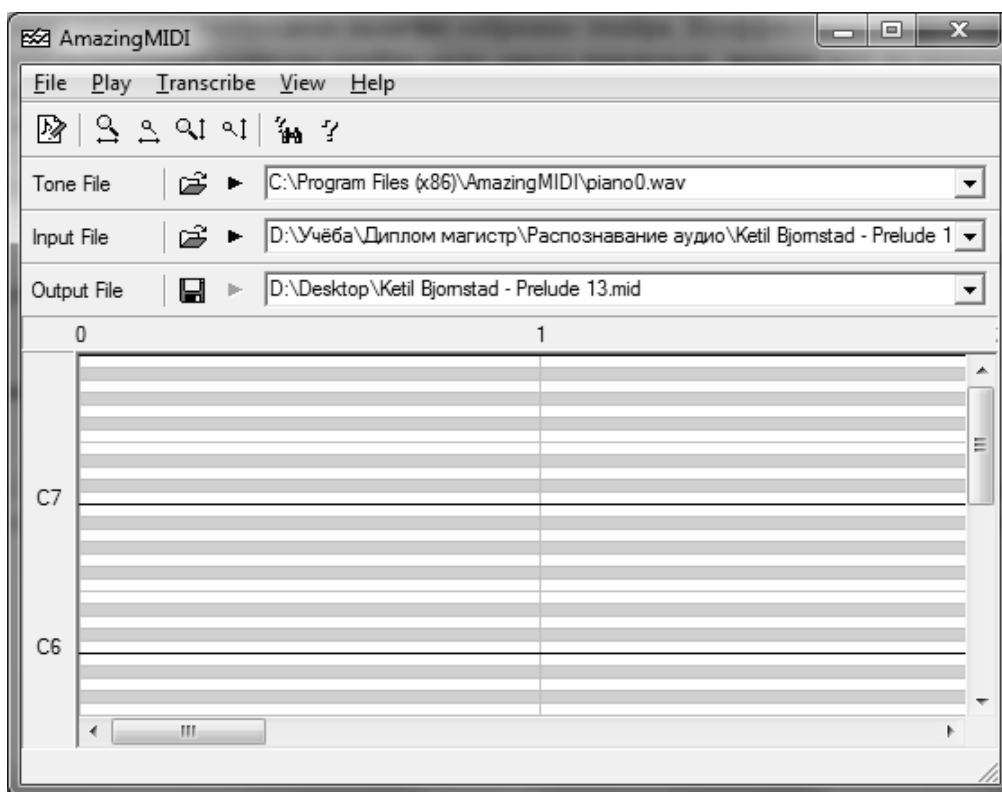


Рис. 1. Главное окно программы AmazingMidi

Недостатки программы: только один инструмент; не позволяет распознавать голос. Для распознавания необходимо наличие «образца» тембра. Неэффективна, так как результирующее качество крайне низко: много призвуков, лишних нот, на выходе, по сути, выдает «кашу» из нот, представляющую нечто близкое к исходной мелодии.

Digital Ear – в целом, решение, близкое к предыдущему (рис. 2). Возможно наличие нескольких инструментов в составе.

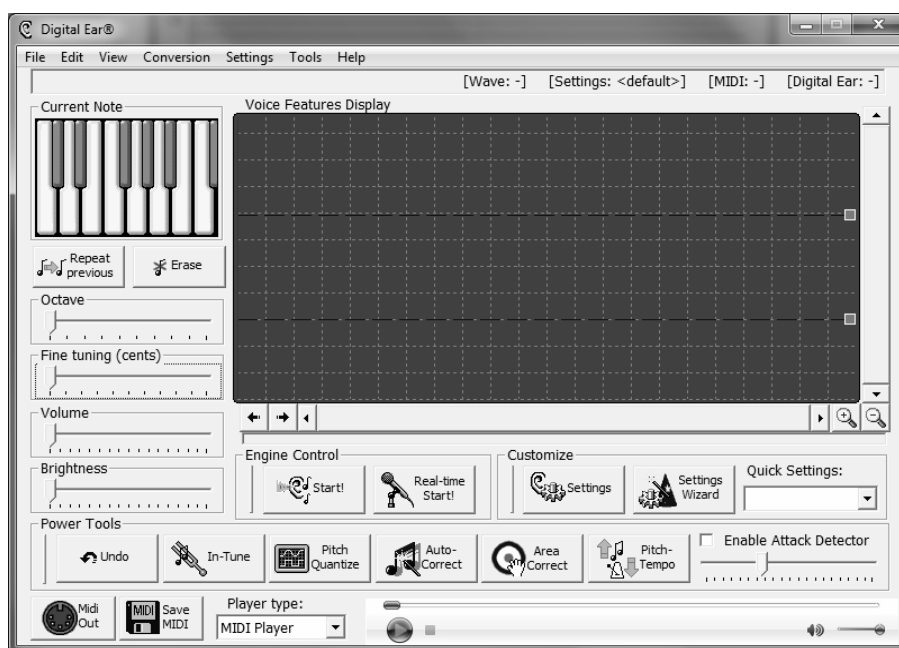


Рис. 2. Главное окно программы «Digital Ear»

В меню «File» пользователь может открыть распознаваемый файл. Далее программа предлагает пользователю выполнить анализ аудиозаписи (попытаться определить диапазон нот, отклонение от стандартного строя), некоторые из них потом можно сменить. После этого можно приступить к распознаванию. В поле «Voice Features Display» будут отображаться ноты.

Если программа «Amazing Midi» захватывала огромное количество призывков, делая результат распознавания бесполезным, то эта программа, наоборот, распознаёт только несколько ключевых нот на эпизоде, что опять-таки делает её неэффективной.

D'Accord iChords 2.0 – программное решение для распознавания аккордов.

Данная программа устанавливается как подключаемый модуль к проигрывателю на компьютере и позволяет распознавать аккорды, входящие в состав композиции. Этот вариант малоинтересен, так как он распознаёт только гармоническую, но не мелодическую, составляющую.

IntelliScore Ensemble WAV to MIDI Converter.

На рис. 3 – главное окно этой программы. Это самое эффективное программное решение – по сути, единственное из существующих, которое позволяет получить хоть какой-нибудь полезный и работоспособный результат. При запуске программы открывается мастер создания проекта, в котором предлагается выбрать звуковой файл и выполнить ряд настроек.

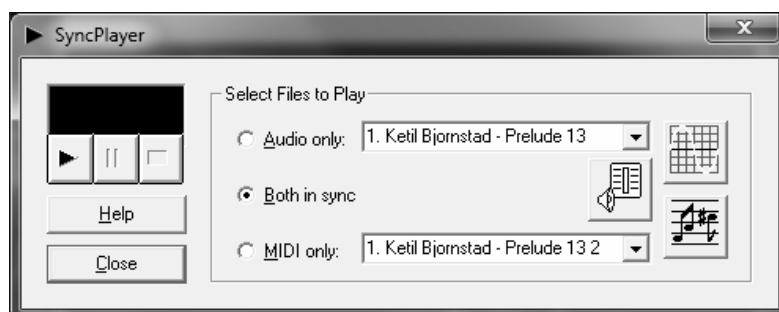


Рис. 3. Главное окно программы IntelliScore Ensemble WAV to MIDI Converter

Недостатки: значительная неточность распознавания вокальной музыки, необходимость вручную обозначать темп, нажимая клавишу на сильные доли, неровность ритма в конеч-

ном результате, платность программы (демо-версия распознаёт первые 15 секунд композиции, что достаточно для оценки качества).

Кроме того, модуль также имеется в секвенсоре *Cakewalk Pro Audio*. Позволяет распознавать одноголосные мелодии.

4. Решение задачи распознавания и преобразования аудиоинформации в ноты

Предложенный метод базируется на следующих действиях:

1. *Преобразование Фурье*, применяемое в целях получения энергии спектральных составляющих сигнала на каждом участке. Поскольку аудиозапись имеет дискретное представление музыкальных данных, используем дискретный вариант записи закона. Энергия i -го сигнала составляется из двух компонент:

$$G_j : a_j = \frac{2\Delta t}{T} \sum_{i=0}^{f_d} S(i\Delta t) \cos(j \frac{2\pi}{t} i\Delta t) ;$$
$$b_j = \frac{2\Delta t}{T} \sum_{i=0}^{f_d} S(i\Delta t) \sin(j \frac{2\pi}{t} i\Delta t),$$

где G_j – энергия сигнала; f_d – частота дискретизации; Δt – промежуток времени между отсчётами; T – период измерения.

Сама энергия тогда составит: $|G_j| = a_j^2 + b_j^2$.

2. *Выделение нот*. Для каждой ноты характерно наличие в спектре нескольких частот. Одна, самая главная, частота имеет наибольшую энергию. Например, как известно, для ноты «ля» первой октавы это – 440 Гц. Остальные частоты определяются в равномерно темперированном строе следующим образом: $f(i) = f_0 \cdot 2^{i/12}$, где f_0 – частота ноты, от которой считаем, i – количество полутонов, которые отстаем от этой частоты.

Например, можно вычислить частоту звука на тон (2 полутона) ниже от «ля» – ноты «соль»: $i = -2$, $f(-2) = 440 \text{ Гц} \cdot 2^{-2/12} \approx 391,995 \text{ Гц}$

Кроме того, в спектре каждая нота имеет более слабые, обертоновые, спектральные составляющие.

3. *Анализ присутствующих музыкальных инструментов*. Самый сложный этап решения проблемы. Здесь предлагается прибегнуть к методам искусственного интеллекта, а именно – к обычному многослойному перцептрон.

Каждый инструмент имеет характерный тембр, который определяется количеством и энергией гармоник, входящих в обертоновые частоты данной ноты. Построив многослойный перцептрон, можно обучить его, подавая на вход образцы тембров разных инструментов разных нот. Параметрами, подаваемыми на вход, в простейшем случае будут n первых спектральных составляющих ноты (так как обертоновый ряд в теории бесконечный). На выходе перцептрон будет выдавать ноту.

4. *Корректировка*. В процессе распознавания неизбежно столкнёмся со следующими проблемами: как распознать корректно ноту, если накладываются несколько нот разных инструментов разной динамики. Возможно, перцептрон будет выдавать некорректные значения. Для этого предлагается использовать ещё один, корректирующий, перцептрон. Обучение его будет заключаться в анализе различных сочетаний нот и аккордов разных тембров.

Проблемой здесь также является определение необходимой сложности используемых нейронных сетей.

Выводы

Результаты анализа задачи распознавания и преобразования аудиоинформации в ноты показали, что на данный момент она актуальна в связи с отсутствием программных продуктов, осуществляющих качественное ее решение. Дальнейшее исследование включает практическую реализацию программного продукта, базирующегося на численных методах (преобразовании Фурье) и методах искусственного интеллекта (многослойный перцептрон), который позволит максимально качественно распознавать аудиоинформацию и преобразовывать ее в необходимый формат.

Список литературы: 1. Бодянский Е.В., Руденко О.Г. Искусственные нейронные сети. Харьков: ТЕЛЕ-ТЕХ, 2004. 369 с. 2. Уоссермен Ф. Нейрокомпьютерная техника. 1992. С. 144-162. 3. Белобородов А.Ю. Распознавание аудиообразов с применением обертонового ряда. 2010. С.72-77.

Поступила в редколлегию 20.11.2011

Булавин Дмитрий Алексеевич, канд. техн. наук, доцент кафедры теоретической и прикладной системотехники факультета компьютерных наук Харьковского национального университета им. В.Н. Каразина. Научные интересы: искусственный интеллект, теория принятия решений. Увлечения: футбол, теннис. Адрес: Украина, 61166, Харьков, ул. Ленина, 3, кв. 20, тел. 702-30-79, e-mail: dimetroid@yandex.ru.

Харитонов Илья Александрович, студент факультета компьютерных наук Харьковского национального университета им. В.Н. Каразина. Научные интересы: искусственный интеллект, компьютерные сети. Увлечения: музыка. Адрес: Украина, 61000, Харьков, ул. Вальтера, 14, e-mail: iliyaisd@gmail.com

УДК 621.396.6

О.Ю. МАЛИЙ, О.Ю. ФАРАФОНОВ, О.А. ДЕМ'ЯНЕНКО

МЕТОД ОПИСУ ІНЖЕНЕРНИХ ЗАДАЧ ЗА ДОПОМОГОЮ УНІВЕРСАЛЬНОЇ МОВИ ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРІВ

Пропонується метод, що дозволяє проводити опис інженерної задачі у вигляді програми на УМП для подальшої трансляції з метою автоматизованої генерації тексту програми обраного мікроконтролера. Метод дозволяє не прив'язуватися до окремих фірм виробників периферійних пристроїв та мікроконтролерів, що в свою чергу дозволяє в підсумку отримати оптимальне схемотехнічне рішення інженерної задачі з мінімальними витратами для виробника.

1. Вступ

Для створення кінцевого пристрою і написання програми керуючого мікроконтролера першим етапом є осмислення інженерного завдання і опис структури пристрою. Для цього інженер, який є користувачем системи автоматизації програмування, повинен першочергово осмислити, який кінцевий результат він хоче побачити при проектуванні пристрою. Процес створення радіоелектронної системи на основі мікроконтролера, з точки зору осмислення принципів її роботи та опису для системи автоматизації проектування, можна розбити на три основних етапи:

1. Визначення вхідних і вихідних пристроїв, які будуть обмінюватися інформацією з кінцевим користувачем системи і навколишнім середовищем.
2. Визначення структури складових частин мікроконтролерної системи.
3. Визначення принципів взаємодії складових частин та загального алгоритму роботи системи.

2. Класифікація пристроїв, що підключаються до мікроконтролерів

Для визначення вхідних і вихідних пристроїв в першу чергу необхідно класифікувати периферійні пристрої, що підключаються до мікроконтролерів та можуть бути виконані у вигляді як зовнішніх пристроїв, так і вбудованої в мікроконтролер периферії. Класифікація пристроїв за принципами організації зв'язку з мікроконтролером, а також по взаємодії з користувачем і навколишнім середовищем наведена на рис. 1.

Як видно з рис. 1, пристрої, що підключаються до мікроконтролерів, можна розділити на вхідні, вихідні та пристрої двостороннього зв'язку.

Вхідні пристрої здатні лише передавати будь-яку інформацію мікроконтролеру для подальшої обробки. До них можна віднести всі види датчиків, пристрої введення, такі як кнопки, клавіатура, а також компаратори, АЦП, ЦАП (у разі підключення їх виходів до мікроконтролера).

До вихідних відносяться пристрої, здатні тільки виводити дані. До них відносяться різні типи індикаторів (світлодіоди, рідкокристалічні індикатори, семисегментні індикатори, дисплеї), виконавчі пристрої (електромеханічні, електричні), АЦП, ЦАП, реєстри (у разі підключення їх входів до мікроконтролера).

До пристроїв двостороннього зв'язку можна віднести різні види зовнішньої та вбудованої пам'яті й інші пристрої, які дозволяють проводити обмін даними з мікроконтролером за стандартними шинами обміну [1].

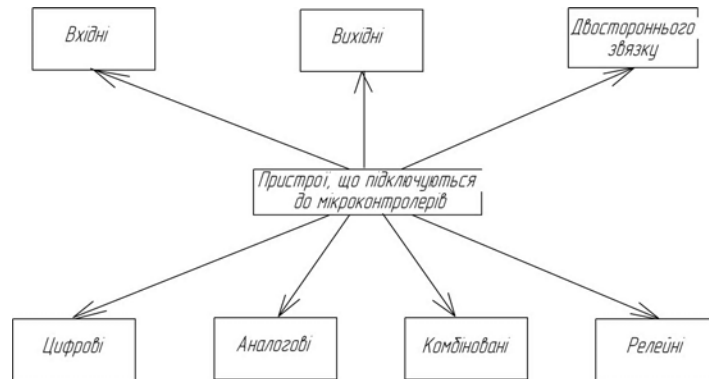


Рис. 1. Класифікація пристроїв, що підключаються до мікроконтролера

3. Опис периферійних пристроїв за допомогою універсальної мови програмування мікроконтролерів

За допомогою запропонованих в [2-6] методів опису структури мікроконтролерів і периферійних пристроїв, що входять до його складу, можна описувати будь-який тип периферійних пристроїв, які підключаються до різних мікроконтролерів. В кінцевому програмному комплексі для цих цілей використовується файл, що містить докладний опис принципів роботи периферійних пристроїв device.dat. Тому користувач/інженер може в програмному комплексі вибирати зі списку необхідний йому тип пристрою, а в разі необхідності конкретний пристрій для подальшого його підключення до мікроконтролера. Завдяки відкритій архітектурі файлу опису пристроїв, а також наявності вбудованих інструментів створення і редагування опису пристроїв, існує можливість додавання як нових типів пристроїв, так і нових конкретних пристроїв, що випускаються різними виробниками.

Для вирішення завдання реалізації першого етапу процесу створення радіоелектронної системи на основі мікроконтролера, з точки зору осмислення принципів її роботи та опису для системи автоматизації проектування, тобто визначення вхідних і вихідних пристроїв, які будуть обмінюватися інформацією з кінцевим користувачем системи і навколишнім середовищем, у програмному комплексі передбачена можливість вибору і додавання пристроїв у проєкт. З точки зору програмної реалізації дана процедура відбувається шляхом додавання в проєкт нового об'єкта, що має певну архітектуру, властивості та функції для управління ним і реалізації взаємодії з мікроконтролером.

Наприклад, термопара, яка за запропонованою класифікацією відноситься до вхідного аналогового пристрою, описується так:

Thermocouple:

Type, commands, tmin, Udt, tmax, gisteresise, dt, gisk, pins.

Після додавання в проєкт цей об'єкт має такі функції для процедури управління ним і реалізації взаємодії з мікроконтролером: overload (true, false) – вихід за верхню межу t° ; notload (true, false) – вихід за нижню межу t° ; work (true, false) – режим нормальної роботи readtemp – функція читання t° .

На другому етапі користувач / інженер вказує структуру взаємодії запропонованих пристроїв з мікроконтролером. З математичної точки зору це відбувається як присвоєння доступними для даного об'єкта параметрами і функціями певних значень, які необхідні для подальшого вибору мікроконтролера і створення блоків ініціалізації пристроїв у коді програми мікроконтролера, а також включення переривань.

Як приклад розглянемо підключення конкретного типу термопари (не задаючись конкретною моделлю даного пристрою для більшої гнучкості використання моделей в кінцевій мікроконтролерній системі) до мікроконтролера для введення з навколишнього середовища значення температури, яка змінюється в діапазоні від 0 до 100°C. При цьому будемо вважати, що коефіцієнт термоЕРС даного типу термопари дорівнює 50мкВ/°С, при мінімальній вимірюваній температурі (тобто при 0) вихідна напруга має значення 100мВ, похибку вимірювання приймемо 0,1 °С. Для спрощення завдання будемо вважати, що ефект гістерезиса для даної термопари відсутній. У цьому випадку об'єкт thermocouple буде мати такі значення параметрів:

```
Type = analog
commands = no
tmin = 0
Udt = 50(мкВ/°С)
tmax = 100
gisteresise = none
u0 = 100(мВ)
dt = 0.1
gisk = 0
pins = 2
```

На третьому етапі процесу створення радіоелектронної системи на основі мікроконтролера, з точки зору осмислення принципів її роботи та опису, для системи автоматизації проектування користувач / інженер визначає принципи взаємодії складових частин та загальний алгоритм роботи системи.

4. Приклади реалізації введення інженерних задач в систему автоматизації програмування

Пропонується для опису функціонування розроблюваної мікроконтролерної системи використовувати описану в [2-6] універсальну алгоритмічну мову програмування мікроконтролерів шляхом вказання логічних взаємозв'язків між об'єктами, прив'язаними до конкретних пристроїв підключеним користувачем на попередніх етапах планування завдання. Опис проводиться за допомогою встановлення чіткого взаємозв'язку вхідних і вихідних пристроїв з використанням, як посередника, мікроконтролера. Взаємозв'язок описується шляхом встановлення умовних зв'язків між об'єктами з допомогою, в першу чергу, оператора умови (if) і оператора присвоєння (=), а також за допомогою мінімального набору функцій і процедур універсальної алгоритмічної мови програмування мікроконтролерів. Як приклад розглянемо опис двох мікроконтролерних систем, в яких використовуються різні види вхідних і вихідних пристроїв.

Перша система, яка пропонується до розгляду як приклад, термометр з індикацією температури на текстовому однорядковому монохромному дисплеї. Система складається з чотирьох основних елементів:

- термопари (властивості, параметри і функції якої розглянуті вище);
- кнопки включення / виключення, яка описана у вигляді вхідного цифрового релейного пристрою з єдиним параметром pin_in, дозволяє встановити в одиницю або скинути в нуль вхід порту мікроконтролера;
- монохромного однорядкового восьмисимвольного текстового дисплею, об'єкта типу display, який описаний у вигляді вихідного цифрового паралельного пристрою;
- безпосередньо керуючого мікроконтролера.

При цьому вважаємо, що система вимірює температуру від 0 до 100°C з кроком 0,5 °С за допомогою термопари. Значення температури виводиться на дисплей у вигляді числа з комою, тобто у форматі від 1 до 3-х цифр до коми, сама кома, одна цифра після коми і значок градуса Цельсія, який складається з двох символів "o" і "C". Кнопка використовується для включення і виключення розроблюваної системи.

У цьому випадку алгоритм роботи даної мікроконтролерної системи матиме такий вигляд (рис. 2):

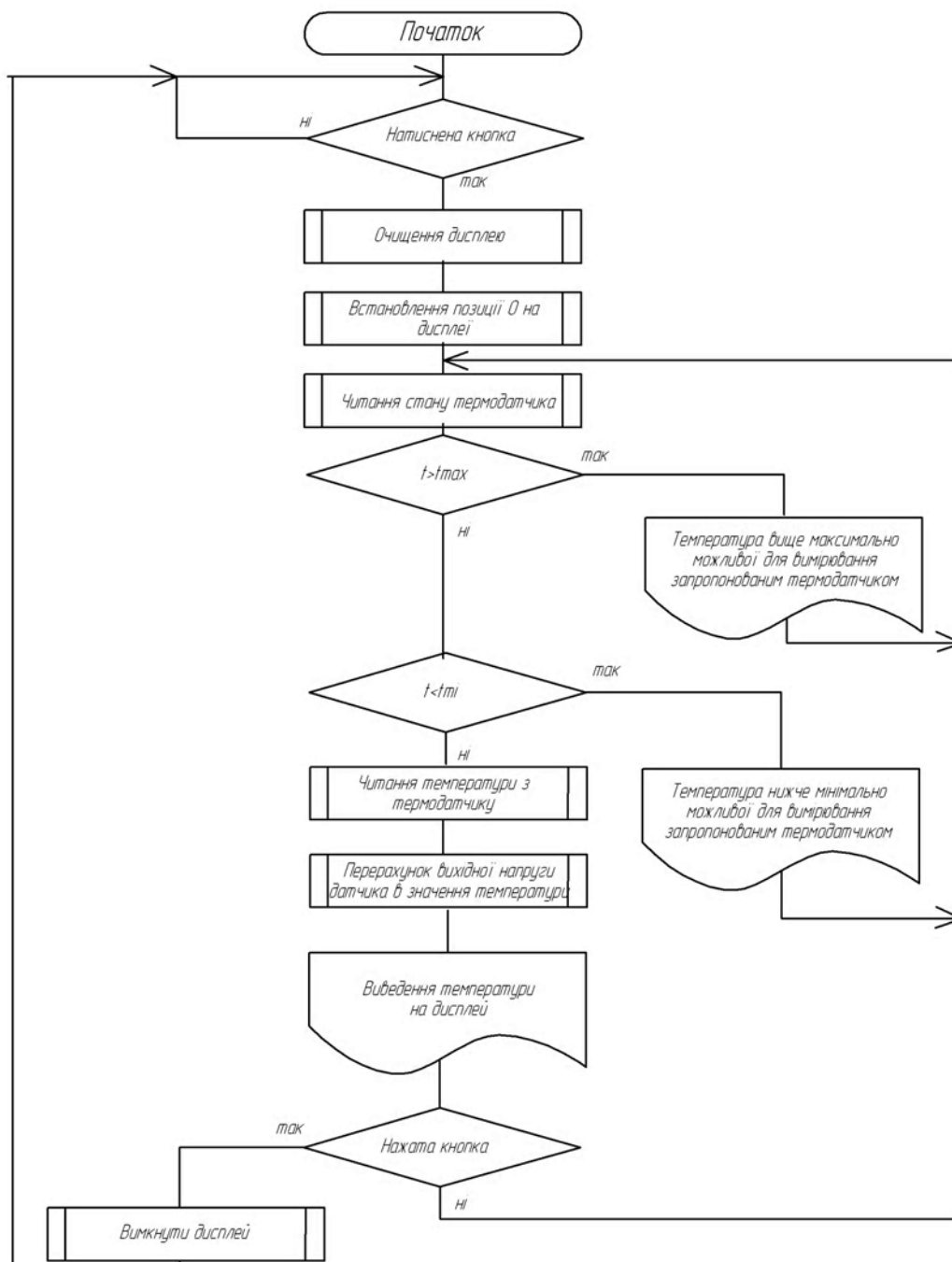


Рис. 2. Алгоритм роботи термометра з індикацією температури на текстовому однорядковому монохромному дисплеї

На універсальній алгоритмічній мові програмування (УМП) даний алгоритм роботи пристрою з урахуванням введених пристроїв у вигляді об'єктів описаний в табл. 1.

З прикладу видно, що інженеру немає необхідності вникати в тонкощі перетворення аналогового сигналу з виходу термопари в значення температури, а також в особливості системи команд управління дисплеєм, а можна використовувати тільки функції об'єктів *thermocouple* і *display*.

Друга система, яку ми розглянемо як приклад, являє собою електронний замок, який працює з використанням електронних смарткарт і управляє електронним замком і двома світлодіодами (червоним і зеленим), які індикують стан замка.

Система складається з п'яти основних елементів.

Таблиця 1

```

m2: if sw1.swon = true then
    pause(500)
endif
    if sw1.swof = true then
        display1.positionset = 0
m1: if thermocouple1.work then
    if thermocouple1.overload = true then
        display1.positionset = 7
        display1.load text = 'temp > max'
    goto m1
    endif
    if thermocouple1.notload = true than
display1.positionset = 7
        display1.load text = 'temp < min'
        goto m1
    endif
    s = thermocouple1.readtemp
temp = S/termocople1.Udt
display1.positionset = 6
If temp < 100 then
Display1.setsimbol = ' '
endif
t= int(temp/100)
if t <>0 then display1.setsimbol = simb(t)
endif
    display1.position set = 5
If temp < 10 then
display1.setsimbol = ' '
endif
temp = temp-t*100
t=int(temp/10)
if t >=10 then
display1.setsimbol =simb(t)
endif
display1.positionset = 4
temp = temp - t*10
t = int (temp)
display1.setsimbol = simb (t)
display1.positionset = 3
display1.setsimbol = ' '
display1.positionset =2
temp=temp - t
t=int(temp*10)
display1.setsimbol = simb (t)
display1.positionset = 1
display1.setsimbol = 'o'
display1.positionset = 0
display1.setsimbol = 'C'
endif
if sw1.swon=true then
display1.dispon=false
pause(500)
go to m2
endif
goto m1

```

– картоприймача / роз'єму для підключення смарткартки з контактом, що визначає наявність картки (в даному випадку незважаючи на пасивність цього пристрою, він описується як цифровий послідовний пристрій двостороннього зв'язку);

– двох світлодіодів, які описані як вихідні цифрові релейні пристрої з єдиним параметром pin_out, що дозволяє вмикатися, якщо вихід порту мікроконтролера дорівнює одиниці, і вимикатися, якщо вихід порту дорівнює нулю;

– електромеханічного замка (в електронній схемі буде виконуватися тільки роз'єм для його підключення), який працює як реле і отже описаний у вигляді вихідного цифрового релейного пристрою єдиним параметром pin_out (зелений світлодіод за бажанням інженера можна запаралелити із замком);

– енергонезалежної пам'яті, що зберігає набір дамів пам'яті дозволених карт, яка описана як послідовний пристрій двостороннього зв'язку, об'єкта типу eeprom (пам'ять може бути реалізована як всередині мікроконтролера, так і зовні у вигляді окремої мікросхеми);

– безпосередньо керуючого мікроконтролера.

При цьому вважаємо, що в початковому стані запалюється червоний світлодіод, система в разі наявності картки в роз'ємі (це визначається наявністю сигналу на контакті картоприймача) виробляє читання дампа пам'яті картки, а також реалізує алгоритм аутентифікації карти challenge / response і порівнює зчитаний дамп з дозволеними дампами для доступу, які зберігаються в незалежній пам'яті. Якщо дамп, зчитаний з карти, збігається з яким-небудь дампом, що зберігаються в пам'яті, і карта пройшла алгоритм аутентифікації, то червоний світлодіод гасне, а загоряється зелений і разом з тим подається сигнал на вхід електромеханічного замка, який після цього відкривається. Після відкриття система відміряє 10 секунд, потім замок закривається, зелений світлодіод гасне і спалахує червоний, і система переходить в початковий стан.

У цьому випадку алгоритм роботи даної мікроконтролерної системи матиме такий вигляд (рис. 3).

На універсальній алгоритмічній мові програмування даний алгоритм роботи пристрою з урахуванням введених пристроїв у вигляді об'єктів описаний в табл. 2.

З цього прикладу також видно, що інженеру немає необхідності вникати в тонкощі обміну даними між пам'яттю і мікроконтролером, а також процесом роботи мікроконтролера зі смарткарти, а просто використовуються функції об'єктів eeprom і smartcard.

Таблица 2

```

lock1.pin_out=false
vs1.pin_out=true
vs2.pin_out=false
m:8
if cardreader1.cardset=true then
  cardreader1.reset-
  byte = 0
m2:if bit<128 then
  bit=0
  byte=byte +1
  card =dump[byte]=0
m1: if <8 then

card_dump[byte]=card_dump[byte]*2+cardreader1.read
  bit = bit+1
  goto m1
endif
  goto m2
endif
  num =0
  eeprom1.setadd=0
m4: if num<16 then
  byte =0
  num=num+1
m3: byte=byte<128 then
  byte=byte+1
  ee_dump[num,byte]=ocprom1.readword
  ee_prom1.set add=eeprom1.get add+1
  goto m3
endif
  goto m4
endif
m5: if num<16 then
  byte =0
  num=num+1
m5: if byte<128 then
  byte=byte+1
  if card_dump[byte]<>ee_dump [num,byte] then
  goto m5
endif
  goto m6
endif
  goto m7
end if
  goto m8
m7: byte =0
  cardreader1.next
m14: if byte <14 then
  byte=byte+1
  bit=0
m13: if bit <8 then
  t=cardreader1.read
  if byte =13 then
  if byte =3 then
  cardreader1.write
  ch[1]=08
  ch[2]=12
  ch[3]=19
  ch[4]=81
  need[1]=22
  need[2]=06
  need[3]=19
  need[4]=41
  i=0
m10: if i<4 then
  j=0
  i=i+1
m9: if j<8 then
  t=ch[i] and 128
  cardreader1.pin[io]=t
  ch[i]=ch[i]*2
  j=j+1
  goto m9
endif
  goto m10
endif
  i=0
m12: if i<4 then
  j=0
  i=i+1
  resp[i]=0
m11: if j<8 then
  cardreader1.write
  t=cardreader1.pin[io]
  resp [i]=resp[i]*2+t
  j=j+1
  goto m11
endif
  goto m12
endif
  end if
  goto m14
end if
  i=0
m15: if i<4 then
  i=i+1
  if resp[i]<>need[i] then
  goto m8
endif
  goto m15
endif
  lock1.pin_out=true
  vs1.pin_out=false
vs2.pin_out=true
pause(10000)
goto m8

```

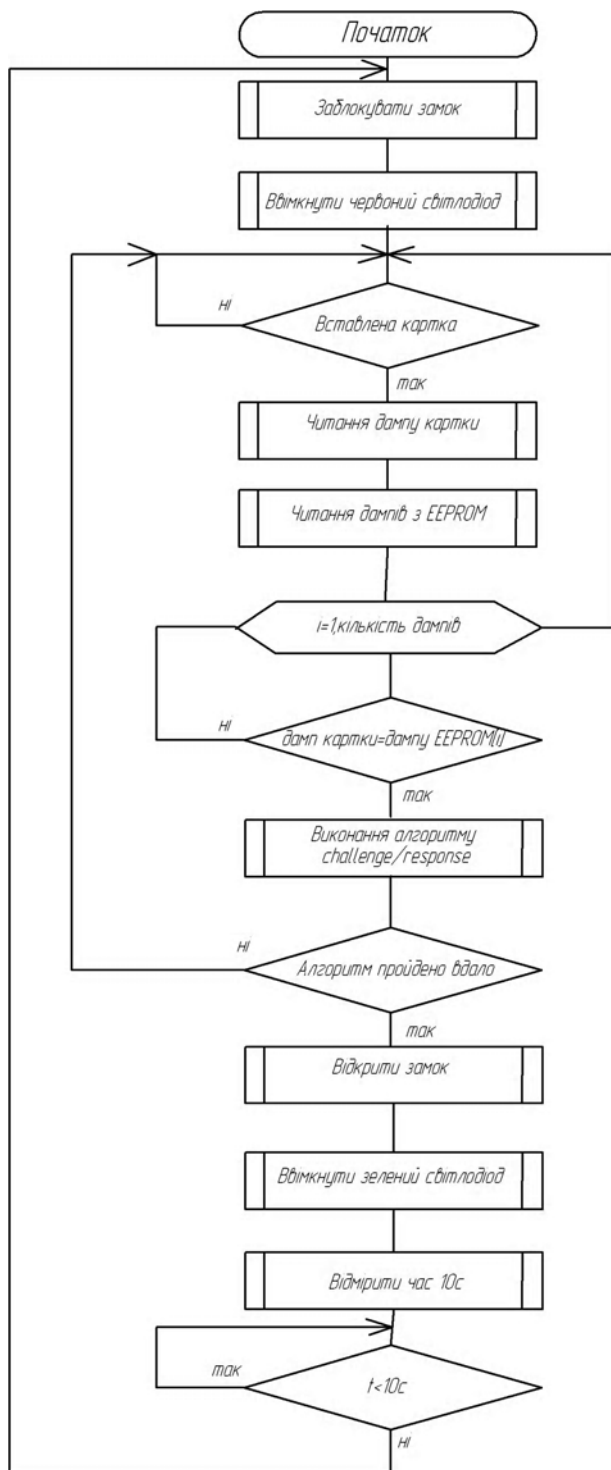


Рис. 3. Алгоритм роботи електронного замка

5. Висновки

Запропонований метод дозволяє проводити опис інженерної задачі у вигляді програми на УМПІ, що дає можливість далі шляхом трансляції проводити автоматизовану генерацію тексту програми обраного мікроконтролера. На відміну від раніше запропонованих методів дозволяє не прив'язуватися до окремих фірм виробників периферійних пристроїв та мікроконтролерів. Завдяки цьому в підсумку можна отримати оптимальне схемотехнічне рішення інженерної задачі з мінімальними витратами для виробника.

Список літератури: 1. Сташин В.В. и др. Проектирование цифровых устройств на однокристальных микроконтроллерах / В.В. Сташин, А.В. Урусов, О.Ф. Мологонцева. М.: Энергоатомиздат, 1990. 223с. 2. Vladimir Krischuk, Alexandr Maly. Development of methods of designing of means of the automated creation of the programs of microcontrollers in difficult electronic systems // "The experience of designing and application of CAD Systems in Microelectronic", Lviv, Ukraine, 2005. P. 279-280. 3. Кришук В.М., Малий О.Ю. Універсальна алгоритмічна мова програмування мікроконтролерів //Тези доповідей Міжнародної науково-практичної конференції "Сучасні проблеми і досягнення в галузі радіоелектроніки, телекомунікацій та інформаційних технологій". Запоріжжя, 2006. С. 152-153. 4. Alexandr Maly, Volodimir Krischuk The dynamic description of system of instructions of microcontrollers // CADSM-2007, Polyana, 21-25 Feb.2007. 5. Малий О.Ю. Універсальна алгоритмічна мова програмування мікроконтролерів / В.М. Кришук, О.Ю. Малий, О.Ю. Воропай // Радіоелектроніка. Інформатика. Управління. 2009. №2. С. 115-120. 6. Малий А.Ю. Математическое обеспечение автоматизации разработки и проверки программ микроконтроллеров / А.Ю. Воропай, В.Н.Кришук, А.Ю. Малий //Радіотехніка. 2009. №3. С. 165-169. 7. Малий О.Ю. Розробка методів універсалізації програмування мікроконтролерів різних сімейств / А.Ю. Малий //Радіотехніка. 2009. №4. С. 210-225.

Надійшла до редколегії 21.11.2011

Малий Олександр Юрійович, асистент каф. конструювання и технологии производства

радіоапаратури Запорізьського національного технічного університету. Адреса: Україна, 69063, Запоріжжя, вул.Жуковського, 64, тел. роб. (061)7698252, e-mail: docsasha@mail.ru.

Фарафонов Олексій Юрійович, канд.техн.наук, доцент кафедри конструювання та технології виробництва радіоапаратури Запорізьського національного технічного університету. Адреса: Україна, 69063, Запоріжжя, вул.Жуковського, 64, тел. роб. (061)7698252, e-mail: farafon@zntu.edu.ua

Демьяненко Олексій Анатолійович, студент магістратури гр.РП-116 Запорізьського національного технічного університету. Адреса: Україна, 69063, Запоріжжя, вул. Жуковського, 64.

МЕТОДИКА ОПРЕДЕЛЕНИЯ ПАРАМЕТРОВ АППАРАТНОГО ГЕНЕРАТОРА СЛУЧАЙНЫХ ЧИСЕЛ, РЕАЛИЗОВАННОГО В ПЛИС АРХИТЕКТУРЫ FPGA

Рассматриваются общие подходы к построению аппаратных генераторов случайных чисел. Предлагается методика оценивания качества случайной последовательности с использованием критерия «хи-квадрат» и результатов прохождения пакета статистических тестов NIST STS. Исследуется влияние параметров генератора случайных чисел на качество случайной последовательности.

1. Введение

Генераторы случайных чисел (ГСЧ, TRNG – True Random Numbers Generators) имеют достаточно широкий спектр применения в современных вычислительных системах. К областям применения ГСЧ можно отнести криптографию, моделирование, компьютерные игры и т.д. В настоящий момент широко используются три основных подхода к получению последовательностей случайных чисел: выборка колебаний генератора, хаотические составляющие в цепях или специальное усиление шумов резисторов или диодов. При этом вопросы влияния параметров генератора (вид генератора, количество цепей, количество инверторов в цепях) на качество формируемой случайной последовательности рассмотрены слабо. В данной работе предлагается методика определения оптимальных параметров ГСЧ в FPGA общего назначения.

Аппаратный (физический) генератор случайных чисел [1] – устройство, которое генерирует последовательности случайных чисел на основе измеряемых параметров протекающего физического процесса. Работа таких устройств часто основана на процессах, таких как:

- тепловой шум,
- фотоэлектрический эффект,
- другие квантовые явления,
- неравномерность в задержках логических элементов.

Эти процессы, в теории, абсолютно непредсказуемы. Аппаратные генераторы случайных чисел, основанные на квантовых процессах, обычно состоят из специального усилителя и преобразователя. Усилитель усиливает очень слабые сигналы, получаемые в результате проходящих физических явлений, до приемлемых размеров, которые преобразуются преобразователем к цифровому виду (рис. 1).

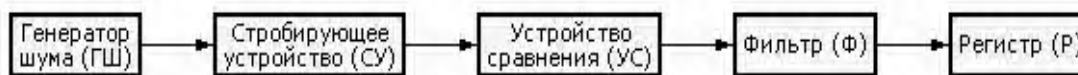


Рис. 1. Аппаратный метод генерации случайных чисел

Аппаратные генераторы случайных чисел относительно медленны и могут производить смещенные последовательности (когда определенная последовательность чисел повторяется чаще). Использование подобных генераторов зависит от потребностей конкретной предметной области и от устройства самого генератора.

Целью данной работы является описание методики определения параметров генератора случайных чисел, реализованного в программируемых логических интегральных микросхемах архитектуры FPGA.

2. Структура аппаратного генератора случайных чисел

В данный момент все большую популярность завоевывают цифровые генераторы случайных чисел. Источниками энтропии в цифровых генераторах чаще всего являются:

- задержки в логических элементах;
- нестабильность внешнего генератора синхросигнала.

В общем случае (рис.2) источник случайного шума генерирует аналоговый сигнал $n(t)$, который является результатом некоторого недетерминированного физического явления. Сигнал аналогового шума оцифровывается (например, с помощью компаратора), получается так называемый оцифрованный аналоговый сигнал $s[i]$. Недетерминированный источник и преобразователь в цифровую форму вместе формируют преобразованный в цифровую форму шумовой источник.

Оцифрованный цифровой шум передается в модуль постпроцессинга, который затем выдает последовательность m -битных случайных слов $r[i]$, так называемых внутренних случайных чисел. Прежде всего, пост-процессор должен регулировать распределение вероятностей «сырых» случайных бит $s[i]$, компенсируя таким образом внутренние несовершенства источника энтропии или цифрового преобразователя (например, отклонение компаратора напряжения). Распределение вероятностей внутреннего случайного слова $r[i]$ намного более близко к общепринятому, чем $s[i]$. Подробнее методы постпроцессинга рассмотрены ниже.

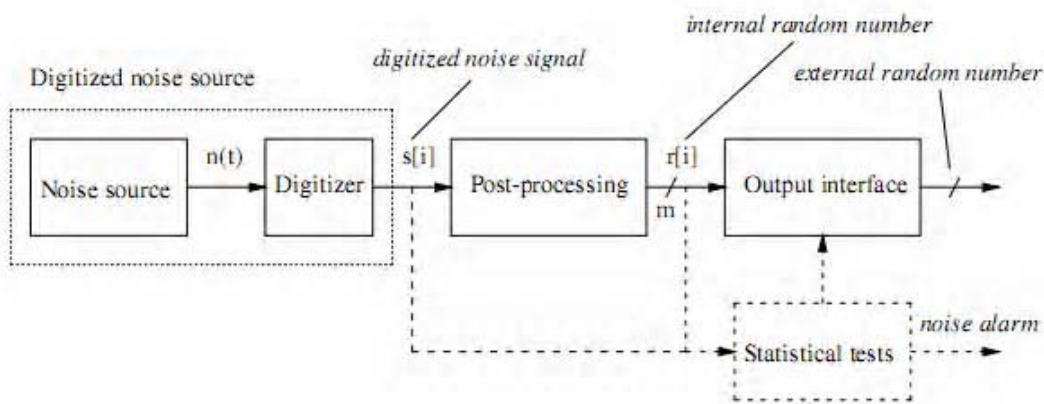


Рис. 2. Структура аппаратного ГСЧ

Шаг постпроцессинга используется, чтобы увеличить энтропию с помощью внутреннего случайного слова $r[i]$, применяя функцию компрессии к входному потоку $s[i]$, что приводит к получению потока с меньшей скоростью и большей случайностью. Это становится особенно важно, если используется источник шума с низкими характеристиками энтропии на бит. Компрессия также обеспечивает устойчивость к различного рода воздействиям.

Метод, использующий неравномерность в задержках логических элементов, идет исключительно по цифровому пути - использует кольцевые генераторы в качестве источника шума (рис.3). Если в кольцевую структуру объединить нечетное количество инверторов, выходное значение каждого инвертора будет колебаться от логического нуля к логической единице и обратно из-за неустойчивой природы цепи. В любой момент времени в цепи наблюдается периодическая квадратная волна.

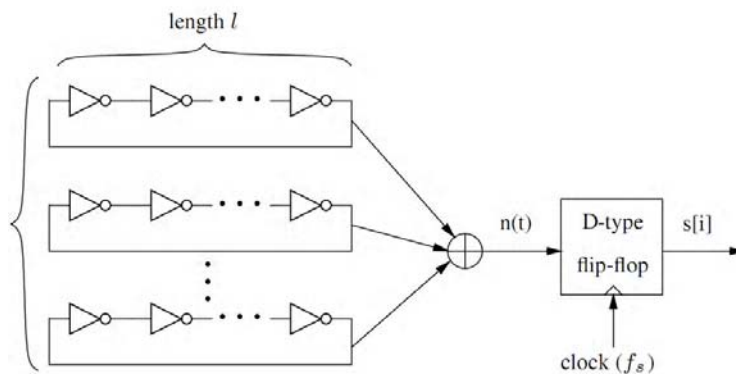


Рис. 3. Структура генератора

В идеале период волны линейно зависит от числа инверторов (т.е. от длины цепи) и задержки одного инвертора. На практике существует некоторая случайность в момент подачи сигнала. Это явление обычно называют вибрацией. Цель цифрового TRNG – обработать полученную энтропию, выбрав неопределенные зоны и недетерминированные части волны.

В общем, существуют две технологии для извлечения случайности из вибраций:

- выборка результатов кольца генераторов с помощью другого генератора (связные генераторы);

- комбинация сигналов ряда генераторов.

Структура, представленная на рис. 3, основана на втором подходе. Сигналы от k объединяются исключаящим или и используются в качестве сигнала $n(t)$.

Из этого сигнала производится выборка с регулярной частотой $f(s)$, результат заносится в D-триггер защелку. Так получается битовый поток $s[i]$.

Альтернативное решение реализации генератора случайных чисел на FPGA работает, сэмплируя высокоточный высокочастотный синхросигнал F_h нестабильным низкочастотным сигналом F_t .

Для этого используется триггер-защелка D-типа, на вход синхронизации которого поступает сигнал F_t , на вход данных – сигнал F_h . Выдача данных ведется с частотой F_1 (рис.4).

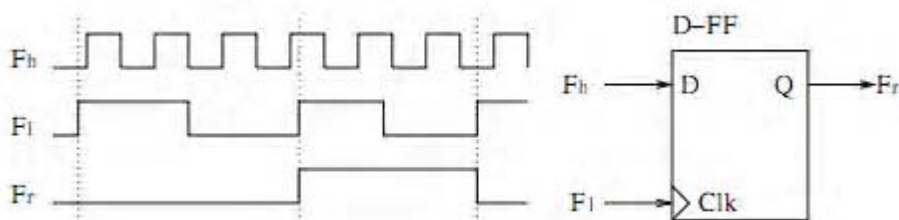


Рис. 4. Использование триггера-защелки для ГСЧ на основе нестабильности внешнего источника тактовой частоты

Для данного генератора существует несколько факторов, влияющих на случайность получаемых последовательностей.

Во-первых, скажность сигнала F_h не обязательно 50%, а следовательно, F_t может случайным образом принимать значения «1» и «0». Для выравнивания неравномерного распределения используется N-битный фильтр частоты.

Во-вторых, на качество получаемых случайных последовательностей влияет выбор частоты синхросигнала. Если изменения периода F_1 недостаточны, корреляция между битами позволит в определенной степени «предсказать» следующий бит на основе предыдущих.

Для улучшения характеристик случайных последовательностей (особенно в случае применения генератора с низкой энтропией) может использоваться постпроцессинг. Компрессия, неизбежная в ходе постпроцессинга, обеспечивает устойчивость к различного рода воздействиям.

Существует два популярных метода постпроцессинга:

- метод фон Ньюмона,
- XOR-коррекция.

XOR-коррекция предполагает обработку входных бит – из двух входных путем суммирования по модулю два получается один выходной. Таким образом, поток сжимается в два раза.

Коррекция фон Ньюмона также рассматривает пары битов, но использует первый из них в случае, если они одинаковы, в противном случае отбрасывает. Результирующий поток будет иметь переменную разрядность, но в среднем фактор сжатия составит 4.

Наряду с простыми коррекциями применяются и более сложные, такие как псевдогенератор BBS или эластичная функция, схема аппаратной реализации которой приведена на рис. 5.

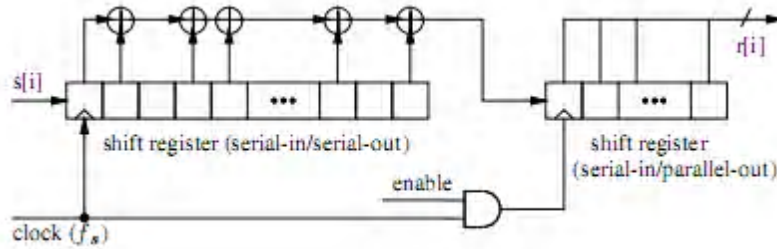


Рис. 5. Реализация алгоритма постпроцессинга, основанная на циклических кодах

Также для постпроцессинга применяется псевдогенератор (Pseudo random number generator - PRNG) BBS (Blum Blum Shub), названный так по именам трех своих создателей.

BBS использует уравнение квадратного вычета (заметим, что это псевдослучайный генератор бит вместо генератора псевдослучайных чисел; он генерирует последовательность битов (0 или 1)).

Ниже приведены шаги генерации:

– Найдем два простых числа p и q в форме $4k+3$, где k – целое число (p и q являются конгруэнтными).

– Выберем модуль $n=p*q$.

– Выберем случайное число r , взаимно-простое с n .

– Вычислим начальное число как $x_0=r^2 \bmod n$.

– Генерируем случайную последовательность $x_{i+1}=x_i^2 \bmod n$.

– Возьмем самый младший бит сгенерированного случайного целого числа (LSB - Least Significant Bit) как случайный бит.

Схема данного метода приведена на рис. 6.

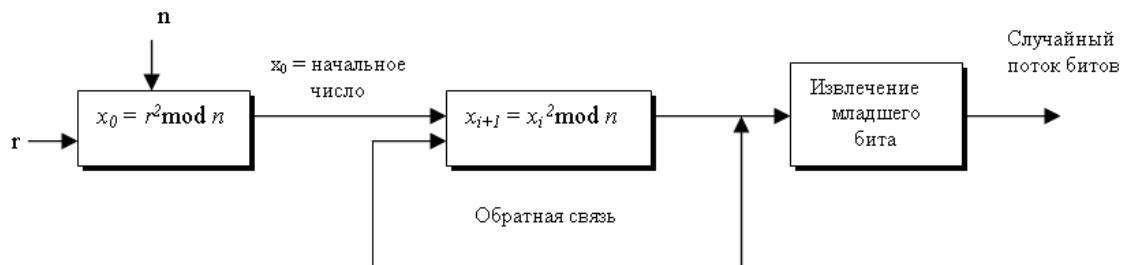


Рис. 6. BBS-метод генерации случайной последовательности

В общем случае, постпроцессинг используют, когда имеется возможность пожертвовать скоростью генерации при необходимости повысить энтропию.

3. Методы оценки качества случайной последовательности

Для оценки качества случайной последовательности применяются два вида критериев:

– эмпирические (статистические) критерии, при использовании которых компьютер манипулирует группами чисел последовательности и вычисляет определенные статистики;

– теоретические (формальные) критерии, для которых характеристики последовательности определяются с помощью теоретико-числовых методов, основанных на рекуррентных правилах, которые используются для образования последовательности.

Идеальный ГСЧ должен выдавать близкие к следующим значения статистических параметров, характерных для равномерного случайного закона:

$$m_r = \left(\sum_{i=1}^n r_i \right) / n,$$

где m_r – математическое ожидание, стремится к значению 0.5; r_i – количество попаданий в i -й интервал; n – число интервалов;

$$D_r = \frac{\sum_{i=1}^n (r_i - m_r)^2}{n};$$

здесь D_r – дисперсия, стремится к значению 0.0833; σ_r – среднеквадратическое отклонение $\sigma_r = \sqrt{D_r}$, которое стремится к 0,2886.

Для доказательства гипотезы о равномерном распределении наиболее часто используются рассмотренные ниже критерии.

Одним из стандартных наборов статистических тестов является NIST STS[2], состоящий из 15 различных тестов. Отличительная особенность NIST STS – открытость алгоритмов и однозначность интерпретации результатов анализа.

На основе открытых алгоритмов NIST STS было разработано программное обеспечение, позволяющее анализировать свойства потока случайных чисел, поступающих от аппаратного ГСЧ через COM-порт в реальном масштабе времени. Целью разработки программного обеспечения является получение зависимости интегральной оценки качества случайной последовательности от параметров конфигурации аппаратного ГСЧ.

Если критерии T_1 - T_{n-1} подтверждают, что последовательность ведет себя случайным образом, это еще не означает, вообще говоря, что проверка с помощью критерия T_n будет успешной. Однако каждая успешная проверка дает все больше и больше уверенности в случайности последовательности. Обычно к последовательности применяется около половины критериев, и если она удовлетворяет им, то считается случайной.

4. Методика проведения экспериментальных исследований свойств аппаратного генератора случайных чисел

Для реализаций аппаратного генератора с различными параметрами с помощью разработанного программного обеспечения для анализа случайных последовательностей была проведена серия экспериментов, целью которых являлось:

- сбор информации о корреляции между качеством работы генератора и его характеристиками;

- построение аналитической модели этой зависимости;

- выявление факторов, наиболее влияющих на эту зависимость.

Каждый эксперимент проводился по следующей схеме:

На первом шаге с помощью разработанного программного обеспечения, генерирующего VHDL генератора, получали код генератора с необходимыми параметрами. Данное приложение позволяет получить код генератора с заданным количеством петель и их длин, а также задать произвольную их длину.

На втором шаге с помощью программного пакета Quartus генерировали файл прошивки.

На третьем шаге с помощью утилиты Programmer из пакета Quartus ПЛИС программировали для работы в качестве генератора случайных чисел

На четвертом шаге с помощью разработанного ПО для оценки статистических свойств случайных последовательностей проводили анализ – по 100 последовательностей по 1024 4-байтных чисел каждая для каждой модификации генератора.

Результаты всех проведенных экспериментов заносили в сводную таблицу, которую анализировали с помощью статистических пакетов. Под результатом следует понимать сводный рейтинг, вычисленный на основании прохождения последовательностью формальных и статистических тестов, подробно описанных в разделе 2, максимальное значение которого составляет 1; за прохождение каждого из шести статистических тестов рейтинг последовательности увеличивался на 0,13, оставшаяся величина определялась по результатам вычисленной доверительной вероятности для формального критерия.

Для удобства анализа результаты всех измерений были занесены в таблицу с тремя независимыми – количество петель, максимальное и минимальное количество инверторов в одной петле – столбцами, и одним зависимым – вычисленный рейтинг для последовательности, порожденной генератором с заданными параметрами. Всего было проведено 128 серий испытаний по 100 последовательностей каждая.

Следует отметить, что существует множество методов аппроксимации (приближение функций многочленами, формула Тейлора, приближение функций тригонометрическими многочленами).

Поскольку необходимая функция имеет три аргумента, применение таких методов потребует большое количество вычислительных ресурсов и времени. Для решения такой задачи наиболее рациональным будет применение нейронных сетей [3], как наиболее перспективное, простое в реализации и наглядное средство для анализа сложных закономерностей.

Для анализа статистических данных использовались программные пакеты Deductor Academic и STATISTICA 6.

Для проверки адекватности обучения сети все данные были разбиты на обучающее (95%) и тестовое (5%) множества.

В качестве активационной функции была выбрана одна из самых распространенных – сигмоида.

В качестве метода обучения нейронной сети был выбран алгоритм Back Propagation.

После выбора параметров было проведено обучение сети и проверка качества обучения на тестовом наборе. Предсказанные таким образом значения с высокой точностью совпадали с экспериментально полученными данными тестового множества, что дает возможность утверждать правильность выбора структуры сети, активационной функции и подбора параметров сети на этапе обучения.

5. Выводы

Оценка полученных результатов проводилась по двум направлениям – выявление общих свойств и закономерностей и построение аналитической модели.

Анализируя полученные результаты, можно сделать следующие выводы:

- случайность в большей мере зависит от длины цепочек, чем от их количества;
- случайность возрастает при увеличении длины цепочек и в меньшей мере – при увеличении их числа;
- целесообразно использовать цепочки одинаковой длины, так как разброс в длинах цепочек снижает случайность.

Экспериментальные исследования проводились с применением ПЛИС архитектуры FPGA двух ведущих фирм-производителей – Altera ACEX1K и Xilinx Spartan 3E. Результаты экспериментальных исследований показали практическую идентичность параметров генераторов случайных чисел с одинаковыми параметрами. Это дает возможность утверждать, что данные зависимости являются общими и платформенно-независимыми.

Список литературы: 1. *Рябко Б.Я., Фионов А.Н.* Криптографические методы защиты информации. М.: Горяч.Линия-Телеком, 2005. 229 с. 2. *Харин Ю.С., Ярмола А.Н., Петлицкий А.И.* Методы и алгоритмы статистического тестирования генераторов случайных и псевдослучайных последовательностей в системах информационной безопасности // Штучний інтелект. 2006. Вып.3. С. 793-803. 3. *Руденко О.Г., Бодянский Е.В.* Основы теории искусственных нейронных сетей. Харьков: ТЕЛЕТЕХ, 2002. 317 с.

Поступила в редколлегию 12.10.2011

Саранча Сергей Николаевич, канд. техн. наук, доцент кафедры ЭВМ ХНУРЭ. Научные интересы: проектирование и моделирование цифровых систем. Увлечения и хобби: интернет-программирование, музыка. Адрес: Украина, 61000, Харьков, пр.Ленина, 14, корпус «з», ауд. 35. Контактный телефон (+38057)702-13-54, e-mail: softpro@kture.kharkov.ua

ИССЛЕДОВАНИЕ МЕТОДОВ ОПТИМИЗАЦИИ В СИСТЕМАХ ПРИНЯТИЯ РЕШЕНИЙ ПРИ ПЛАНИРОВАНИИ ТОВАРООБОРОТА

Предлагается материал разработанного оптимального плана товарооборота путем составления для него математической модели и ее решения. Анализируется результат, полученный при помощи модели, и делаются соответствующие выводы. Определяются положительные и отрицательные стороны данного подхода и приводятся соответствующие рекомендации для улучшения.

1. Введение

В данное время исследуется проблема принятия решения в задачах оптимизации плана товарооборота. Товарооборотом называют объем продажи товаров и оказания услуг в денежном выражении за определенный период времени.

Известно, что товарооборот является важнейшим показателем не только для торгующей организации, но и для национальной экономики. Во внутрифирменном значении этот показатель отражает успех организации, спрос покупателей на реализуемые товары. От товарооборота зависят все важнейшие финансово-экономические показатели торговли, включая валовой доход, уровень издержек обращения, размер и эффективность использования товарных ресурсов, фонд заработной платы, прибыль, рентабельность. Анализ товарооборота позволяет оценивать соответствие имеющихся товаров спросу населения для принятия мер по оптимизации структуры товарооборота, увеличению объемов реализации, ускорению товарооборота, ритмичности и равномерности продаж. В народно-хозяйственном значении товарооборот характеризует обеспеченность населения товарами. Выручка от оборота промышленного производства обеспечивает дальнейшие вложения в развитие промышленности, поэтому рост продаж в наибольшей степени стимулирует развитие промышленности и определяет возможности ее дальнейшего развития. В этом смысле торговля является двигателем промышленности. И поэтому оптимизация плана товарооборота – одна из главных задач в экономике, от которой зависят очень многие показатели.

Актуальность исследования. Поиски оптимальных решений привели к созданию специальных математических методов и уже в XVIII веке были заложены математические основы оптимизации (вариационное исчисление, численные методы). Однако до второй половины XX века методы оптимизации во многих областях науки и техники применялись очень редко, поскольку практическое использование математических методов оптимизации требовало огромной вычислительной работы, которую без ЭВМ реализовать было крайне трудно, а в ряде случаев – невозможно.

При постановке задачи оптимизации предполагается существование конкурирующих свойств процесса, например: количество продукции – расход сырья; количество продукции – качество продукции.

Выбор компромиссного варианта для указанных свойств и представляет собой процедуру решения оптимизационной задачи.

Обычно оптимизируемая величина связана с экономичностью работы рассматриваемого объекта (аппарат, цех, завод). Оптимизируемый вариант работы объекта должен оцениваться какой-то количественной мерой – критерием оптимальности.

В зависимости от своей постановки любая из задач оптимизации может решаться различными методами, и наоборот - любой метод может применяться для решения многих задач. Методы оптимизации могут быть скалярными (оптимизация проводится по одному критерию), векторными (оптимизация проводится по многим критериям), поисковыми (включают методы регулярного и методы случайного поиска), аналитическими (методы дифференциального исчисления, методы вариационного исчисления и др.), вычислительными (основаны на математическом программировании, которое может быть линейным, нели-

нейным, дискретным, динамическим, стохастическим, эвристическим), теоретико-вероятностными, теоретико-игровыми [2].

Исследованные и предложенные принципы оптимизации в данной работе имеют научную инновацию, а также их практическое значение. Результаты были опробованы в системах принятия оптимальных решений с использованием многих критериев.

Цель работы: рассмотрение теории принятия решений, исследование ее возможностей, а также рассмотрение плана оптимизации товарооборота.

Задачи исследования. В рамках данной работы изучить различные подходы к выбору оптимальных решений при оптимизации плана товарооборота.

Сущность исследования. В качестве основного инструментария для разработки системы принятия решения в системах оптимизации плана товарооборота использовались методы многокритериальной и эволюционной оптимизации.

2. Постановка задачи и математическая модель

Общая постановка задачи состоит в следующем. Нужно определить вектор $X^{(0)}$, обеспечивающий компромисс между величиной прибыли, валовым объемом и минимальной себестоимостью, который удовлетворяет ограничениям минимизации производственного времени.

Исходя из специализации, торговая организация может реализовать n -групп товаров $T_j (j=1, \dots, n)$, для этого используют m -видов ресурсов ($i=1, \dots, m$): складские помещения, трудовые ресурсы, запасы товара, затраты оборота, план товарооборота, минимально допустимый план товарооборота по j -й группе, объемы которых известны. Нормативные данные i -го ресурса по j -й группе, торговый доход в расчете на единицу товарооборота по j -й группе, общие объемы ресурсов приведены в таблице.

Лимитированные ресурсы и показатели	Товарная группа					Объем ресурсов
	T1	T2	T3	T4	T5	
Складские площади, м ²	9	13	8	5	11	600000
Трудовые ресурсы, чел- часы	75	70	25	40	35	400000
Товарные запасы, грн	15	21	15	10	10	900000
Затраты оборота, грн	85	115	140	60	68	600000
План товарооборота, грн	100	75	85	25	75	300000
Минимально допустимый план товарооборота по j -й группе	600	---	750	500	---	300000
Прибыль	40	15	10	35	70	

Составим математическую модель в классе экстремальных задач, на основании решения которой определим оптимальный план хозяйственной деятельности торгового предприятия; проведем анализ оптимального решения, выявим узкие места на торговом предприятии и дадим рекомендации по их решению на основе анализа дефицитных ресурсов.

Имеем задачу линейного программирования, относящуюся к подразделу задач определения оптимального ассортимента. Постановка таких задач в общем виде следующая. Имеется m -видов ресурсов в количествах $a_1, \dots, a_i, \dots, a_m$, которые могут быть использованы при производстве n -видов изделий. Задана матрица $A = \|a_{ik}\|$, где a_{ik} характеризует нормы расхода i -го ресурса на единицу k -го изделия ($k=1, 2, \dots, n$).

Эффективность выпуска единицы k -го изделия характеризуется показателем S_k , удовлетворяющим условиям линейности, согласно которому суммарная эффективность выпуска a_1 изделий с показателем c_1 и a_2 изделий с показателем c_2 равна $c_1 a_2 + c_2 a_1$. Определить план выпуска изделий (оптимальный ассортимент), при котором суммарный показатель эффективности принимает наибольшее значение [2].

Общий вид математической модели для таких задач выглядит так. Обозначив количества единиц k -х изделий, выпускаемых предприятием, через $X_k \geq 0$ (где $k=1, 2, \dots, n$), получим математическую модель задачи:

максимизировать
$$Z = \sum_{k=1}^n C_k X_k \quad (1)$$

при условиях
$$\sum_{k=1}^n a_{ik} X_k \leq a_i \quad (i=1,2,\dots,m). \quad (2)$$

Дальнейшее решение задачи можно получить используя симплексный метод.

Кроме указанных ограничений по ресурсам (2), в условие задачи, а следовательно, и в ее математическую модель могут вводиться дополнительные ограничения на планируемый выпуск продукции (ограничения по ассортименту, условия комплектности).

По общей математической модели для решения задач такого типа с учетом всех особенностей поставленной перед нами задачи строим математическую модель, где x_1, x_2, x_3, x_4, x_5 – количество единиц изделий вида T_1, T_2, T_3, T_4, T_5 соответственно. На основе этого составим целевую функцию:

$$P(X) = 40x_1 + 15x_2 + 10x_3 + 35x_4 + 70x_5 \rightarrow \max \quad (3)$$

при условиях:

$$\begin{cases} 9x_1 + 13x_2 + 8x_3 + 5x_4 + 11x_5 = 6000000, \\ 75x_1 + 70x_2 + 25x_3 + 40x_4 + 35x_5 = 400000, \\ 15x_1 + 21x_2 + 15x_3 + 10x_4 + 10x_5 = 900000, \\ 85x_1 + 115x_2 + 140x_3 + 60x_4 + 68x_5 = 600000, \\ 100x_1 + 75x_2 + 85x_3 + 25x_4 + 75x_5 = 300000, \\ X_1 \geq 600, \\ X_3 \geq 750, \\ X_4 \geq 500, \end{cases} \quad (4)$$

$$X_j > 0 \quad j=1,5.$$

У нас имеются ограничения по двум параметрам: по объему ресурсов (нельзя использовать больше ресурсов, чем у нас есть) и по минимальному плану товарооборота по j -й группе товаров (вероятно, что при меньшем товарообороте мы понесем убытки).

Если решать задачу «вручную», то необходимо привести систему ограничений к каноническому виду, т.е. представить в виде уравнений:

$$\begin{cases} 9x_1 + 13x_2 + 8x_3 + 5x_4 + 11x_5 + x_6 = 6000000, \\ 75x_1 + 70x_2 + 25x_3 + 40x_4 + 35x_5 + x_7 = 400000, \\ 15x_1 + 21x_2 + 15x_3 + 10x_4 + 10x_5 + x_8 = 900000, \\ 85x_1 + 115x_2 + 140x_3 + 60x_4 + 68x_5 + x_9 = 600000, \\ 100x_1 + 75x_2 + 85x_3 + 25x_4 + 75x_5 + x_{10} = 300000, \\ x_1 - x_{11} = 600, \\ x_3 - x_{12} = 750, \\ x_4 - x_{13} = 500, \\ X_j > 0 \quad j=1,13. \end{cases} \quad (5)$$

После этого по этой системе можно составить симплекс-таблицу и решить ее. Но в силу того, что наша модель будет рассчитана с помощью программной реализации, систему ограничений в каноническом виде и симплекс-таблицу строить не надо, так как программа их построит сама автоматически.

1. Величина прибыли, получаемой предприятием, определяется с помощью соотношения:

$$F_1(X) = \sum_{j=1}^n C_j^{(1)} X_j \rightarrow \max, X_j \in Q = \{1,2,\dots,n\}, \quad (6)$$

где Q – множество видов продукции, выпускаемой предприятием.

2. Показатель качества выпускаемой продукции задается соотношением:

$$\sum_{i=1}^S P_i * X_i \rightarrow \max . \quad (7)$$

Для конкретных значений функция цели примет вид:

$$F_2(X) = 10x_1 + 12x_2 + 8x_3 + 16x_4 + 11x_5 \rightarrow \max . \quad (8)$$

3. Минимизация себестоимости:

$$\sum_{i=1}^S C_i * X_i \rightarrow \min . \quad (9)$$

4. Минимизация производственного времени:

$$\sum_{i=1}^S T_i * X_i \rightarrow \min . \quad (10)$$

Тогда задача исследования может быть сформулирована таким образом.

Определить оптимальный план $X^{(0)} \in O$ производства продукции, удовлетворяющий указанным критериям (1) - (4).

Ограничениями на выпуск продукции различных видов служат производственные ресурсы b_1, b_2, \dots, b_m . С учетом норм затрат ресурсов на единицу каждого типа продукции указанные ограничения можно записать в виде:

$$AB \leq B^T; \quad (11)$$

$$x \geq 0, \quad (12)$$

$$B^T = \{b_1, b_2, \dots, b_m\}; \quad (13)$$

$A = \{a_{ij}\}, i=1, m, j=1, n$ – матрица норм затрат ресурсов на единицу каждого типа продукции.

Выражение (9) описывает условия, которые необходимо учесть в годовой производственной программе. Строкам матрицы A соответствуют все виды ресурсов (группы машин, запасы материалов), рассматриваемые в задачах. Соответствующие строкам матрицы A компоненты вектора B указывают ограничения видов ресурсов или объёмов производства, которые установлены для годовой производственной программы предприятия. Неравенство (10) представляет собой обычные условия неотрицательности, вытекающие из смысла задачи.

3. Методы решения задач

Один из возможных методов решения состоит в том, что вначале находятся три оптимальных вектора производства $x^{(i)}, i = \overline{1, 3}$, каждый из которых соответствует одному из локальных критериев (1) - (4). Затем определяется выпуклая линейная комбинация $X^{(0)}$, представляющая собой оптимальную (компромиссную) программу относительно указанных критериев:

$$X^{(0)} = v_1 x^{(1)} + v_2 x^{(2)} + v_3 x^{(3)}, \quad (14)$$

$$\sum_{i=1}^3 v_i = 1, v \geq 0. \quad (15)$$

Для анализа результатов исследуемой задачи необходимо рассматривать методы эволюционной оптимизации, а также методы многокритериальной оптимизации.

Сначала рассмотрим решение задач методами многокритериальной оптимизации.

Поиск оптимального взаимодействия производится с помощью принципов решения многокритериальных задач. Критерии задачи не однородны, так как часть критериев оптимизации стремится к минимальному значению, а один – к максимальному. Приведенные критерии оптимизации находятся в существенном экономическом противоречии, так как с сокращением сроков поставки товара от производителя к потребителю возрастают транспортно-заготовительные издержки и затраты организации, связанные с хранением.

В качестве критериев оптимизации принимаются три параметра. Критерий оптимизации издержек:

$$F_1(X) = \sum_{i=1}^N \sum_{j=1}^N c_{i,j} * x_{i,j} \rightarrow \min . \quad (16)$$

Критерий оптимизации сроков поставок:

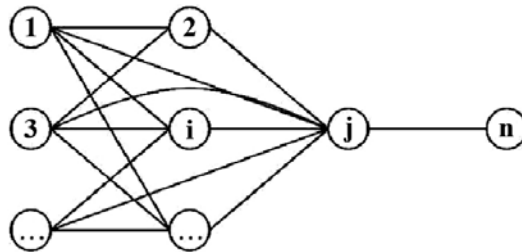
$$F_2(X) = \sum_{i=1}^N \sum_{j=1}^N t_{i,j} * x_{i,j} \rightarrow \min . \quad (17)$$

Критерий оптимизации коэффициентов загрузки:

$$F_3(X) = \sum_{i=1}^N \sum_{j=1}^N k_{i,j} * x_{i,j} \rightarrow \max . \quad (18)$$

В общем случае при взаимодействии элементов (рисунок) в рамках одного технологического процесса, а также при движении продукции по каналам распределения необходимо учитывать следующие основные критерии:

1. Снижение затрат на товародвижение.
2. Сокращение затрат времени на доставку продукции.
3. Учет загрузки складского хозяйства и транспорта.



Графическое представление взаимодействия элементов

В статье предложена многокритериальная модель, которая учитывает перечисленные выше критерии. Рассматривается вертикальная интеграция без вступления в какие-либо альянсы и т. п. Взаимодействие начинается лишь при условии, что оно выгодно финальному звену - торговой компании. На практике отдельные звенья товаропроводящей сети имеют свои интересы, однако в критериях или ограничениях они не учитывались, так как рассматривается движение сквозного материального потока [5].

Согласно (16)-(18), эта модель является многокритериальной с противоречивыми критериями. Рассмотрим решения задач по каждому локальному критерию. Исходя из рассмотренных моделей, каждая из них представляет собой однокритериальную оптимизационную задачу, которая обычно может быть решена методами математического программирования.

Учитывая однотипность математических моделей каждой из локальных задач, являющихся задачами линейного программирования, в качестве метода решения каждой из них может быть предложен симплекс-метод, который по существу представляет собой вычислительный алгоритм [1].

В результате объединения трех локальных оптимизационных задач рассматривается объединенная многокритериальная компромиссная модель, которая по существу является игровой моделью, и решение может реализовано одним из методов решения матричных игр, в частности, матричным методом [1].

Чтобы достичь поставленной задачи, необходимо найти оптимальное решение для каждой функции цели. Для этого будем использовать вместо традиционных методов оптимизации, таких как математическое программирование, методы эволюционной оптимизации.

Рассмотрим также решение указанных задач методами эволюционной оптимизации.

1. Максимизация прибыли.

Расчет показателей качества продукции относится к задачам линейной оптимизации. В общем виде её можно записать так:

$$\sum_{i=1}^S (P_i - C_i) * X_i \rightarrow \max . \quad (19)$$

Эту задачу обычно решают симплекс-методом.

Идея симплекс-метода состоит в последовательном продвижении по базисам опорных планов вплоть до получения оптимального решения или доказательства неразрешимости задачи. При этом значение целевой функции должно увеличиваться.

2. Определение валового объема выпускаемой продукции.

Для решения этой задачи с использованием ГА в качестве общей математической модели применяем формулу:

$$\sum_{i=1}^S P_i * X_i \rightarrow \max . \quad (20)$$

Для конкретных значений функция цели примет вид:

$$F_2(x) = 10x_1 + 12x_2 + 8x_3 + 16x_4 + 11x_5 \rightarrow \max . \quad (21)$$

3. Третьей функцией цели представим минимизацию себестоимости, которая имеет общий вид:

$$\sum_{i=1}^S C_i * X_i \rightarrow \min . \quad (22)$$

Запишем эту функцию с конкретными значениями:

$$F_3(x) = 3x_1 + 4x_2 + 4x_3 + 2x_4 + x_5 \rightarrow \min . \quad (23)$$

4. И, наконец, в роли четвертой функции цели будет выступать минимизация производственного времени:

$$\sum_{i=1}^S T_i * X_i \rightarrow \max , \quad (24)$$

$$F_4(x) = 2x_1 + 2x_2 + x_3 + x_4 + 3x_5 \rightarrow \max . \quad (25)$$

Остановимся на применении генетических алгоритмов следующих видов.

Для решения задачи, представленной моделями (1)-(2), используем генетический алгоритм типа метода муравьиных колоний [1].

Основу поведения муравьев составляет самоорганизация, механизмы которой обеспечивают теоретически оптимальное поведение. Принципы его состоят в достижении системой некоторой глобальной цели в результате низкоуровневого взаимодействия ее элементов.

Муравьиный алгоритм применяется следующим образом: в начальный момент времени, в который входит эта функция базы знаний, находится количество муравьев, равное числу кластеров, куда входит эта функция. При этом каждый муравей имеет строгую принадлежность тому кластеру, из которого он начал свое движение. Принадлежность кластеру проявляется в том, что муравей более восприимчив к феромону, оставленному муравьями из «своего» кластера:

$$F_1(X) = \sum_{j=1}^n C_j^{(1)} X_j \rightarrow \max, X_j \in Q = \{1, 2, \dots, n\}, \quad (26)$$

где Q – множество видов продукции, выпускаемой предприятием.

Для конкретных значений функция цели примет вид:

$$F_2(x) = 10x_1 + 12x_2 + 8x_3 + 16x_4 + 11x_5 \rightarrow \max . \quad (27)$$

Муравьиный алгоритм применяется на двух этапах анализа знаний системы. Вначале он запускается на пространственной (многомерной) модели базы, после чего на основании его работы делаются первоначальные выводы. Затем модель упрощается : удаляются некоторые связи между функциями , отдельные функции объединяются в более крупные структурные единицы , структура знаний отображается на двумерное пространство. После этого 1 алгоритм запускается на упрощенной плоской модели знаний [4].

Также для решения задач воспользуемся генетическими алгоритмами. С их помощью можно оптимизировать работу нефтяных трубопроводов; распределять инструменты в металлообрабатывающих цехах. Генетические алгоритмы имитируют процесс естественного отбора в природе. Для решения задачи, более оптимального с точки зрения некоторого критерия, все решения описываются набором чисел или величин нечисловой природы:

- отбор сильнейших наборов хромосом, которым соответствуют наиболее оптимальные решения;
- скрещивание – получение новых индивидов при помощи смешивания хромосом наборов отобранных индивидов;
- мутации – преобразование хромосомы, случайное изменение одного или несколько генов (чаще одного).

В результате смены поколений вырабатывается такое решение поставленной задачи, которое уже нельзя дальше улучшать.

Для рассмотрения данной задачи используем минимизацию себестоимости:

$$\sum_{i=1}^S C_i * X_i \rightarrow \min . \quad (28)$$

Генетический алгоритм – это простая модель эволюции в природе, реализованная в виде компьютерной программы. В нем используются аналог механизма генетического наследия и аналог естественного отбора. При этом сохраняется биологическая терминология в упрощенном виде. В сущности генетические алгоритмы являются разновидностью методов поиска с элементами случайности и имеют целью нахождение лучшего, а не оптимального решения задачи. Это связано с тем, что для сложной системы часто требуется найти хотя бы удовлетворительное решение, а проблема достижения оптимума отходит на второй план.

В процессе работы генетического алгоритма все указанные выше операторы применяются многократно и ведут к изменению исходной популяции. Поскольку операторы отбора, скрещивания, мутации и редукции по своей сути направлены на улучшение отдельной особи, то результатом их работы является постоянное улучшение популяции.

Генетические алгоритмы используют аналогию между естественным отбором и процессом выбора наилучшего решения из множества возможных. Его суть состоит в том, что более приспособленные особи имеют больше возможностей.

Преимуществом генетических алгоритмов перед другими является простота их реализации, относительно высокая скорость работы, параллельный поиск решения сразу несколькими особями, позволяющий избежать попадания в “ловушку” локальных оптимумов (нахождения первого попавшегося, но не самого удачного оптимума). Недостаток — сложность выбора схемы кодирования, возможность вырождения популяции, сложность описания ограничений планирования [4].

4. Выводы

Изучена задача по оптимизации плана товарооборота в системах принятия решений. Также детально проработана типичная модель этого вида задач. Рассмотрена детерминированная задача. Построена модель оптимального плана товарооборота. Для решения применялся симплексный метод с искусственным базисом.

Линейное программирование применимо для построения математических моделей тех процессов, в основу которых может быть положена гипотеза линейного представления реального мира: экономических задач, задач управления и планирования, оптимального размещения оборудования и пр. Симплексный метод линейного программирования отлично подошел для решения задачи оптимизации товарооборота.

Научная новизна: результатом проведенных исследований является решение оптимизационной задачи в системах принятия решений при планировании товарооборота, с применением различных методов эволюционной и многокритериальной оптимизации, а также использование генетических алгоритмов в управлении запасами.

Практическая значимость: результатом применения предложенных методов является нахождение оптимальных показателей в системах принятия решений при планировании товарооборота деятельности предприятия.

Список литературы: 1. Гвоздинский А.Н., Якимова Н.А., Губин В.А. Методы оптимизации в системах принятия решений. Харьков: ХНУРЕ, 2006. 325 с. 2. Бондаренко М.Ф., Гвоздинский А.Н. Оптимизационные задачи в системах принятия решений. Харьков: ХТУРЕ, 1998. С.179-191. 3. Гвоздинский А.Н., Клименко Е.Г. Применение генетических алгоритмов для решения оптимизационных задач. Харьков: ХНУРЭ.2001.С.390-391. 4. Гвоздинский А.Н., Малышкин В.А. Применение методов эволюционной

оптимизации для решения задач производственного планирования. ХНУРЕ, 2011. С.97-102. **5. Просвиркин Н.Ю.** Экономико-математическая многокритериальная модель управления материальными потоками в сетевых интегрированных структурах. СГАУ им.Королева, Самара, 2007. С.743-755.

Поступила в редколлегию 18.10.2012

Гвоздинский Анатолий Николаевич, канд.тех.наук, профессор кафедры искусственного интеллекта ХНУРЭ. Научные интересы: оптимизация процедур принятия решений в сложных системах управления. Адрес: Украина, 61661, Харьков, ул. Академика Ляпунова, 7, кв.9. тел. 702-38-23.

Бушнов Сергей Валерьевич, студент, бакалавр специальности интеллектуальные системы принятия решений, факультет КИ ХНУРЭ. Адрес: Украина, 61115, ул. 17-го Партсъезда, 34-А, кв.6, тел. 093-94-20-920, e-mail: sergii.bushnov@gmail.com

УДК 519.7

А.Н. ГВОЗДИНСКИЙ, В.А.МАЛЫШКИН, И.В.ТИЩЕНКО

ПРИМЕНЕНИЕ МЕТОДОВ ЭВОЛЮЦИОННОЙ ОПТИМИЗАЦИИ В ЗАДАЧАХ УПРАВЛЕНИЯ ЗАПАСАМИ

Предлагается подход, основанный на эволюционных методах, к решению проблем управления запасами, возникающих при поиске рационального плана раскроя. Показывается возможность сравнительно легкой адаптации алгоритма к различным вариантам постановки задачи.

Введение

В данной работе приводится формальная постановка задачи, на решение которой изначально был направлен первый генетический алгоритм, и общая схема работы самого алгоритма. Описываются составляющие генетического алгоритма и их назначение. Для решения конкретной задачи разработчик, исследователь может сконструировать свою конкретную схему работы генетического алгоритма из его составляющих. При этом разработчик руководствуется особенностями предметной области, формализации задачи, структурой используемых данных, возможно, даже результатами тестирования других схем генетического алгоритма. Разумеется, в строении всех таких алгоритмов есть общие элементы и последовательность действий, которые являются основой генетических алгоритмов и отличают их от многих других. Данная статья посвящена опыту конструирования конкретных вариантов генетического алгоритма.

Состояние проблемы. Управление запасами – это балансирование между двумя целями, взаимоисключающими друг друга в своих полярных точках: сокращение совокупных затрат, направленных на содержание запасов, и обеспечение максимальной надежности производственного процесса. Для предприятий, имеющих в своем составе цеха штамповки, специфичной является задача раскроя, неразрывно связанная с задачей управления запасами.

Управление запасами и решение задачи раскроя на таких производствах осложняется следующими обстоятельствами:

- большое количество заготовок и, следовательно, большая размерность задачи управления запасами (десятки и сотни типоразмеров заготовок);
- случайный график поставок, обусловленный случайностью вырубки заготовок из полосы, что приводит к сложности расчета издержек производства и хранения.

Задачу управления запасами предлагается решить с помощью генетического метода и показать его преимущество.

Генетический алгоритм — это математическая модель эволюции популяции искусственных особей. В популяции каждая особь характеризуется своей хромосомой S_k . Формально хромосома есть цепочка символов $S_k = S_{n_1}, \dots, S_{kN}$, где N — длина цепочки. Хромосома определяет приспособленность (пригодность) особи $S = 1, 2, \dots, n$, где n — численность популяции. Цель состоит в максимизации функции приспособленности $f(S_k)$, т.е. в нахождении особи с максимальной приспособленностью. Важно отметить, что

генетический алгоритм ищет решение, как можно более близкое к оптимуму, но не гарантирует нахождение точного максимума функции приспособленности.

Эволюция популяции моделируется последовательностью поколений $S = 0, 1, 2, \dots, N$. В каждый следующий момент времени его состав меняется в целях увеличения пригодности особей. Это и есть процесс эволюции популяции.

Для каждого следующего поколения отбираются особи с относительно большими значениями приспособленностей. Хромосомы приспособленных особей «скрещиваются» и подвергаются малым «мутациям». Отбор, скрещивание, мутация — все это является генетическими операторами, процедурами, применение которых позволяет получить новое поколение популяции.

Формально генетический алгоритм — это любой алгоритм поиска приближенного решения поставленной задачи, построенный по схеме генетического алгоритма, где S -е поколение популяции обозначается как $[0, 3]$:

Шаг 0. Создать случайную начальную популяцию $S_k(0)$.

Шаг 1. Вычислить приспособленность $f(S_k)$ каждой особи S_k к популяции $S_k(t)$.

Шаг 2. Производя отбор особей из $S_k(t)$ в соответствии с их приспособленностями $f(S_k)$ и применяя генетические операторы (скрещивания и мутации) к отобранным особям для получения потомства, сформировать популяцию следующего поколения $S_k(t+1)$.

Шаг 3. Повторять шаги 1, 2 для $t = 0, 1, 2, \dots, N$ до тех пор, пока не выполнится некоторое условие окончания эволюционного поиска (прекращается рост максимальной приспособленности в популяции, число поколений t достигает заданного предела.).

Возможны различные варианты генетического алгоритма, которые отличаются по схемам отбора особей из текущего поколения в новое, операторам скрещивания и мутации хромосом особей, по форме представления хромосом. Традиционный вид генетического алгоритма базируется на следующей частной схеме: *цепочки символов в хромосомах бинарные (символы S_k принимают значения 0 либо 1), длина цепочек постоянна ($N = const$); метод отбора — пропорционально-вероятностный; скрещивания производятся по односточечной схеме.*

1. Постановка задачи

На заготовочный участок радиоцеха поступили листы фольгированного гетинакса размером $l_1 * l_2$ мм. Его необходимо разрезать на заготовки для печатных плат: П1 размером $l_3 * l_4$ мм, П2 — $l_5 * l_6$ мм, П3 — $l_7 * l_8$ мм, П4 — $l_9 * l_{10}$ мм, необходимость в которых П1, П2, П3, П4 единиц соответственно.

Построить модель и на ее основании составить экстремальную задачу, решение которой дает минимальное количество разрезанных выходных листов. Найти решение задачи в условиях минимизации отходов раскроя. Предусмотреть возможность максимизации количества комплектов печатных плат при условии выполнения комплектности П1:П2:П3:П4= $k_1:k_2:k_3:k_4$.

Исходные данные: $l_1 * l_2$, П1, П2, П3, П4, k_1, k_2, k_3, k_4 .

Используемые переменные: $\Pi_i = a_i$ — необходимое количество заготовок i -го типа; C_j — отход при раскрое единиц исходного материала j -м способом; $j = \overline{1, m}$ — количество вариантов раскроя; a_{ij} — количество заготовок i -го типа ($i = \overline{1, 4}$), полученных при раскрое единицы исходного материала по j -му варианту; i — вид заготовки; q — количество изделий (П1, П2, П3, П4); a — количество разрезанных листов; K — количество типов листов, поступающих на раскрой ($K=1$); l_i — комплектность- количество заготовок i -го вида в одном комплекте; x_i — количество единиц исходного материала, который раскраивается j -м вариантом.

2. Формализация исследуемой задачи

Сформируем ограничения для данной задачи:

$$x_j \geq 0, j = \overline{1, m}, \quad (1)$$

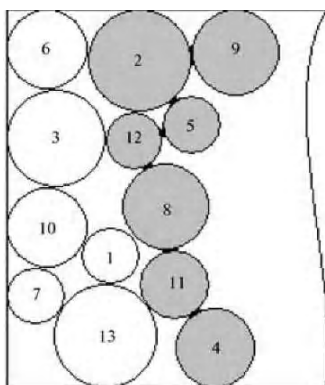
$$\sum_{j=1}^m a_{ij} x_j = \Pi_i. \quad (2)$$

всех номеров предметов в некотором порядке, отражающем их размещение. Физический внешний вид особи, конкретные координаты размещения предметов на полосе можно получить, выложив предметы на полосу в порядке их перечисления в хромосоме по некоторым правилам переработки – декодирования. Эти правила называются процедурой декодирования, или декодером. Различные декодеры могут работать по разной схеме, но суть одна – выложить предметы на полосу согласно некоторому правилу в порядке, указанном в хромосоме. Правила являются эвристическими и содержательно строятся так, чтобы размещение было по возможности плотным.

В качестве критерия оптимизации, критерия оценки пригодности особи в поставленных задачах рассматривается длина полосы, занятая размещением предметов.

Пустоты учитываются и нумеруются, начиная с числа, на единицу большего количества предметов, но со знаком минус. Декодер, работающий с такой структурой данных, называется блочным. Размещение представлено в виде совокупности кортежей (блоков): $(1, 2, \dots, N)x_j$, где $j = 1, 2, \dots, N$.

А именно: $(1; 2)x_1, (3; 4; -9; 2)x_2, (3; 5; 6)x_3, (7; -10; 5; 6)x_4, (7; 8; 6)x_5, (-11)x_6$. Одна из особенностей блочного декодера в том, что размещая очередной предмет, он сначала проверяет все образовавшиеся ранее пустоты, подойдут ли они по размеру для достижения большей плотности размещения (рисунок).



Формальная запись края размещения

Для лучшего описания понятия края упаковки можно провести физическую аналогию. Пусть торец полосы W — дно двумерного стакана, нальем в него воду. Все хотя бы частично намокшие предметы — это край полосы.

При помощи специальной структуры данных краевой декодер хранит информацию о крае текущего размещения и располагает каждый следующий предмет вплотную к краю, минимизируя общую занятую предметами длину полосы.

Для решения задачи предлагается модификация генетического алгоритма, использующая процедуру декодирования как параметр. Сам генетический алгоритм (т.е. схема его работы, смены поколений), структура хромосомы, операторы мутации, скрещивания, селекции не чувствительны к геометрии размещаемых предметов. Например, оператор мутации случайным образом переставляет два номера предметов в хромосоме. Оператор скрещивания двух хромосом порождает двух потомков. Начальный участок дочерней хромосомы совпадает со случайным участком хромосомы одного родителя, а конец состоит из оставшихся предметов, перечисленных в том порядке, в котором они следуют в хромосоме другого из родителей.

Например, оператор скрещивания (для случайных параметров $p = 3, q = 4$) для родителей $S_{p1} = (1, 2, 3, 4, 5, 6, 7, 8, 9)$ и $S_{p2} = (4, 7, 9, 1, 2, 5, 3, 8, 6)$ порождает потомка $S_{c1} = (3, 4, 5, 6, 7, 9, 2, 8)$ и для тех же родителей (для удобства понимания по-другому записанных) $S_{p2} = (4, 7, 9, 1, 5, 3, 8, 6)$ и $S_{p1} = (1, 2, 3, 4, 5, 6, 7, 8, 9)$ потомка $S_{c2} = (9, 1, 2, 5, 3, 4, 6, 7, 8)$. Оператор мутации (для случайных параметров $a = 2, b = 7$) исходную особь $S = (9, 8, 7, 6, 5, 4, 3, 2, 1)$ преобразует в особь вида $S_m = (9, 3, 7, 6, 5, 4, 8, 2, 1)$.

Генетический алгоритм показал, что оценочная полиномиальная сложность алгоритма и относительно невысокие времена работы программы не зависят от типа задачи, в то время

как оценочная сложность блочного декодера пропорциональна квадрату размерности задачи (числу предметов), а краевого — линейно зависит от размерности.

Разработанный алгоритм ориентирован на расширение и модификацию. Смена декодера позволяет решать задачу раскроя полубесконечной полосы как на прямоугольные, так и на круглые предметы. Предположительно, если воспользоваться таким эвристическим приемом, как годограф Хэкманна в качестве декодера, то тот же алгоритм возможно использовать для раскроя полубесконечной полосы на предметы нерегулярной формы.

Обе изначально поставленные задачи раскроя предполагают размещение предметов не на полубесконечной полосе, как обсуждалось при формализации задач, а на наборе прямоугольных листов. Данный алгоритм был модифицирован и для решения задач раскроя в таких, действительно прикладных, а не абстрактных постановках.

В реальном производстве, например, металлургическом, все листы имеют одинаковую ширину. Это выгодное обстоятельство дает возможность не рассматривать набор отдельных листов, а условно разбить на них непрерывную полосу. Предложено оставить в математической постановке задачи полубесконечную полосу, но добавить ограничение: предметы не должны пересекать границы между листами.

При модификации алгоритма остались неизменными все генетические операторы и сама схема работы генетического алгоритма.

Выводы

Полученные результаты позволяют сделать вывод о перспективности применения эволюционных методов для решения задач управления запасами, возникающих на этапе как планирования, так и реализации производственных процессов. Основным преимуществом эволюционных, генетических методов можно назвать сравнительно простую (по сравнению с традиционными методами решения) адаптацию алгоритма к различным вариантам постановки задачи. С помощью одного и того же варианта алгоритма можно решать целый ряд схожих задач. Вероятностный характер исходных данных компенсируется наличием в системе гарантийного запаса, уровень которого можно менять в зависимости от величины спроса на товар, или, например, стабильности поставок. Кроме того, этот способ может быть дополнен подходом, суть которого заключается в постепенной корректировке исходного расписания в процессе его реализации в зависимости от степени отличия реальных параметров задачи от ранее прогнозируемых.

Научная новизна: результатом приведенных исследований является решение оптимизационной задачи управления запасами с помощью методов эволюционной оптимизации с использованием генетических алгоритмов.

Практическая значимость: результатом практического использования полученных в работе итогов является нахождение оптимального расписания поставок товаров в складские помещения в условиях ограниченных объемов.

Список литературы: 1. Гвоздинский А.Н., Клименко Е.Г. Методы аналитической обработки информации // Радиоэлектроника и информатика 2000. №4. С. 111-112. 2. Норенков И.П., Разуляев С.Ю. Расчет динамических расписаний // Вестник МГТУ им. Н.Э.Баумана. 2001. №2(43) С. 14-185. 3. Норенков И.П. Генетические алгоритмы решения проектных и логистических задач // Информационные технологии. 2000. №9 С. 140-212. 4. Норенков И.П. Генетические методы структурного синтеза проектных решений // Информационные технологии. 1998. № 1. С. 40-257.

Поступила в редколлегию 14.11.2011

Гвоздинский Анатолий Николаевич, канд. тех. наук, профессор кафедры искусственного интеллекта ХНУРЭ. Научные интересы: оптимизация процедур принятия решений в сложных системах управления. Адрес: Украина, 61166, Харьков, ул. Академика Ляпунова, 7, кв.9. тел. 702-38-23.

Мальшкин Владимир Александрович, магистрант специальности интеллектуальные системы принятия решений, факультет КН ХНУРЭ. Адрес: Украина, 61166, Харьков, ул. Академика Ляпунова, 7, кв. 164, тел. 0956605746 e-mail: royallifeua@gmail.com

Тищенко Иван Вячеславович, студент-бакалавр специальности интеллектуальные системы принятия решений, факультет КН ХНУРЭ. Адрес: Украина, 61166, Харьков, ул. Авиационная, 1, кв.1, тел. 0504065212 e-mail: shamanitos@ Rambler.ru

ПОВЫШЕНИЕ ТЕСТОПРИГОДНОСТИ КРИТИЧЕСКИХ СИСТЕМ УПРАВЛЕНИЯ

Разрабатывается и тестируется на прикладных примерах метод повышения тестопригодности оборудования защит из состава управляющей системы безопасности АЭС. Отличительной особенностью метода от существующих [1-16] является использование функциональных элементов защит, построенных на базе арифметических операций, без использования логических операций и операций отношения. Это позволяет контролировать работоспособность данных элементов по их реакции на изменения входного непрерывного сигнала от канала ввода в АЦП через все элементы защит, в которых используется данный сигнал, до дискретного выходного элемента, формирующего команду защиты на конкретный исполнительный механизм. Данный метод позволяет обеспечить контроль и диагностирование целого ряда неисправностей типа «несрабатывание», относящихся к категории скрытых в существующих реализациях оборудования защит, которые используют логические операции и операции отношения.

1. Введение

Одним из основных показателей, характеризующих надежность оборудования защит из состава управляющих систем безопасности (УСБ) атомных электростанций (АЭС), является вероятность правильного выполнения дискретной функции по формированию последовательности команд защитных действий с учетом наличия отказов типа «несрабатывание». Критерием такого вида отказа является отсутствие команды защиты при наличии «исходного» события, т.е. при появлении на входах оборудования защит УСБ любой совокупности данных, которая должна вызвать формирование команды.

Для введения в специфику языка, определенного стандартами, техническими условиями, эволюцией технической диагностики, как науки, занимающейся развитием теории и практики восстановления работоспособности вычислительных систем, рассмотрим следующие понятия и определения [17].

Модель – структура элементов и/или процессов, с определенной степенью адекватности описывающая объект и/или явление. Логический анализ – процесс определения логического состояния линий объекта, или его компонентов при наличии или отсутствии неисправностей в условиях существования частичной неопределенности в компонентах: модель, входные воздействия, реакции.

Техническое состояние – совокупность исправного и всех неисправных состояний. Логическое состояние характеризуется значениями сигналов на линиях объекта при наличии или отсутствии неисправностей. В качестве объекта исследования выступает вычислительная система, компьютер, цифровое устройство и их модели, описанные в виде конечных автоматов. Эквивалентным является понятие – объект тестирования – Unit Under Test (UUT). Компоненты UUT: логические элементы, интегральные микросхемы, примитивные элементы. Последние характеризуются наличием таблицы переходов (истинности) и отсутствием структуры. Объект тестирования – изделие или его составные части, техническое состояние которых подлежит определению. Диагностирование – процесс определения технического состояния с заданной точностью.

Дефект – каждое отдельное несоответствие изделия требованиям нормативной документации. Повреждение – вид дефекта, определяемый событием, заключающимся в нарушении исправного состояния при сохранении работоспособного. Несущественный дефект – событие, заключающееся в нарушении работоспособного состояния при сохранении состояния правильного функционирования. Отказ – событие, заключающееся в нарушении состояния правильного функционирования. Техническое состояние объекта – совокупность исправного и множества наперед заданных неисправных состояний. Исправным называется состояние объекта, при котором он соответствует всем требованиям нормативно-технической документации. Неисправным называется состояние объекта при наличии в

нем дефекта. Работоспособным называется состояние объекта, при котором он может выполнять свои функции при наличии повреждения. Состояние объекта, при котором он может выполнять свои функции при наличии несущественного дефекта, называется состоянием правильного функционирования. Состояние неправильного функционирования определяется объектом, имеющим отказ. Если объект имеет катастрофические отказы или морально устарел на данный момент времени, его техническое состояние определяется как предельное или не подлежащее техническому обслуживанию. Контроль (проверка) – определение технического состояния объекта с точностью до исправного (работоспособного) и множества неисправных (неработоспособных).

2. Постановка задачи исследования

Ввиду того, что отказ типа «несрабатывание» для УСБ в целом может быть причиной возникновения нештатной ситуации или аварии, разработка методов контроля и диагностирования, позволяющих выявлять такого рода отказы, является актуальной задачей и предметом различного рода исследований и конструкторских решений. В общем случае к отказам УСБ данного типа могут приводить комбинации как однотипных (отказы по общей причине), так и разнотипных видов скрытых неисправностей в резервированных компонентах УСБ, имеющих временную корреляцию [1-9].

Применительно к типовой структуре УСБ, наличие скрытых неисправностей на несрабатывание означает, что в элементах оборудования защит (электронных компонентах, блоках, программном обеспечении), реализующих алгоритмические функции, скрытая неисправность типа «несрабатывание» гарантировано может быть обнаружена не в момент ее возникновения, а только при появлении комбинации входных сигналов, соответствующих срабатыванию защиты.

Данная проблема актуальна для всех типов оборудования, реализующего дискретные функции, независимо от используемой элементной базы и принципов построения: аналоговые приборы на транзисторах или реле, программно-логические интегральные схемы (ПЛИС), микропроцессоры с инструкциями в виде программного кода.

Цель исследования – существенное уменьшение времени подготовки тестовых данных для совместного (concurrent) анализа в рабочем режиме логических схем управления критическими системами.

Задачи: 1) Обзор существующих методов совместного тестирования систем управления критическими объектами. 2) Разработка моделей и методов совместного тестирования систем управления. 3) Верификация совместного тестирования систем управления моделями критических объектов.

3. Модели процесса тестирования

Предлагаются технологичные и эффективные процесс-модели тестирования диагностирования функциональных нарушений в цифровых системах [18-19]. Используются регистровые или матричные (табличные) структуры данных, которые ориентированы на параллельное выполнение логических операций при поиске дефектных компонентов систем управления критическими объектами.

Проблема синтеза или анализа компонентов произвольной структуры может быть сформулирована в виде взаимодействия на основе симметрической разности (хог-операции на булеане) в кибернетическом пространстве ее модели F с входными воздействиями T и реакциями L : $f(F, T, L) = \emptyset \rightarrow F \Delta T \Delta L = \emptyset$.

Киберпространство – совокупность взаимодействующих по метрике информационных процессов и явлений, использующих в качестве носителя компьютерные системы и сети. В частности, компонент пространства представлен k -мерным вектором (кортежем) $a = (a_1, a_2, \dots, a_j, \dots, a_k)$, $a_j = \{0, 1\}$ в двоичном алфавите. Нуль-вектор есть k -мерный кортеж, все координаты которого равны нулю: $a_j = 0, j = \overline{1, k}$. Метрика β кибернетического (двоичного) пространства определяется единственным равенством, которое формирует нуль-вектор для хог-суммы расстояний d_i между ненулевым и конечным числом точек

(объектов), замкнутых в цикл: $\beta = \bigoplus_{i=1}^n d_i = 0$.

Расстояние (по Хэммингу) между двумя объектами (векторами) a и b определяется в виде производного вектора: $d_1 = d(a, b) = a_j \oplus b_j$. Иначе: метрика β векторного логического

двоичного пространства есть равная нулю-вектору хог-сумма расстояний между конечным числом точек (вершин) графа, образующих цикл. Сумма n -мерных двоичных векторов, задающих координаты точек цикла, равна нулю-вектору. Данное определение метрики оперирует отношениями, что позволяет сократить систему аксиом с трех до одной и распространить ее действие на любые конструкции n -мерного киберпространства. Классическое задание метрики для определения взаимодействия одной, двух и трех точек в векторном логическом пространстве является частным случаем β -метрики при $i=1,2,3$ соответственно:

$$M = \begin{cases} d_1 = 0 \leftrightarrow a = b; \\ d_1 \oplus d_2 = 0 \leftrightarrow d(a, b) = d(b, a); \\ d_1 \oplus d_2 \oplus d_3 = 0 \leftrightarrow d(a, b) \oplus d(b, c) = d(a, c). \end{cases}$$

Метрика β кибернетического многозначного пространства, где каждая координата вектора (объекта) определена в алфавите, составляющем булеан на универсуме примитивов мощностью p : $a_j = \{\alpha_1, \alpha_2, \dots, \alpha_r, \dots, \alpha_m\}$, $m = 2^p$, есть равная \emptyset -вектору (по всем координатам) симметрическая разность расстояний между конечным числом точек, образующих цикл: $\beta = \bigoplus_{i=1}^n d_i = \emptyset$. Равенство пустому вектору симметрической разности по координатного теоретико-множественного взаимодействия подчеркивает равнозначность компонентов (расстояний), формирующих уравнения, где единственная координатная операция $d_{i,j} \Delta d_{i+1,j}$, используемая, например, в четырехзначной модели Кантора, определяется соответствующей Δ -таблицей:

Δ	0	1	x	\emptyset
0	\emptyset	x	1	0
1	x	\emptyset	0	1
x	1	0	\emptyset	x
\emptyset	0	1	x	\emptyset

\cap	0	1	x	\emptyset
0	0	\emptyset	0	\emptyset
1	\emptyset	1	1	\emptyset
x	0	1	x	\emptyset
\emptyset	\emptyset	\emptyset	\emptyset	\emptyset

\cup	0	1	x	\emptyset
0	0	x	x	0
1	x	1	x	1
x	x	x	x	x
\emptyset	0	1	x	\emptyset

\bar{a}	0	1	x	\emptyset
$\bar{\bar{a}}$	1	0	\emptyset	x

Здесь также приведены таблицы истинности для других базовых теоретико-множественных операций, далее используемых по тексту. Число примитивных символов, образующих замкнутый относительно теоретико-множественных координатных операций алфавит, может быть увеличено. При этом мощность алфавита (булеана) определяется выражением $m = 2^p$, где p – число примитивных символов. Введенная метрика представляет не только теоретический интерес, но имеет и практическую направленность на обобщение и классификацию задач технической диагностики путем создания модели хог-отношений на множестве из четырех основных компонентов. Процедуры синтеза тестов, моделирования неисправностей и поиска дефектов можно свести к хог-отношениям на графе (рис. 1) полного взаимодействия четырех вершин (функциональность, устройство, тест, дефекты) $G = \{F, U, T, L\}$.

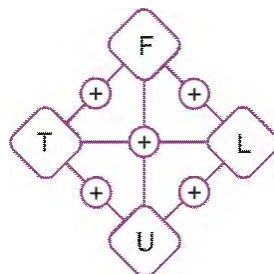


Рис. 1. Граф взаимодействия компонентов технической диагностики

Такой граф порождает четыре базовых треугольника, которые формируют 12 практически полезных триад отношений, формулирующих задачи технической диагностики:

$T \oplus F \oplus L = 0$	$T \oplus L \oplus U = 0$	$T \oplus F \oplus U = 0$	$F \oplus L \oplus U = 0$
1) $T = F \oplus L$	4) $T = L \oplus U$	7) $T = F \oplus U$	10) $F = L \oplus U$
2) $F = T \oplus L$	5) $L = T \oplus U$	8) $F = T \oplus U$	11) $L = F \oplus U$
3) $L = T \oplus F$	6) $U = T \oplus L$	9) $U = T \oplus F$	12) $U = F \oplus L$

Введение вершины U в граф взаимодействия компонентов технической диагностики расширяет функциональные возможности модели, появляются новые свойства полученной системы. Введение в структуру новой вершины должно иметь весомые аргументы в пользу ее целесообразности. Что касается представленного на рис. 1 графа, содержательно все задачи можно классифицировать в группы следующим образом. Группа 1 – теоретические эксперименты (на модели функциональности), без устройства: 1) Синтез теста по модели функциональности для заданного списка неисправностей. 2) Построение модели функциональности на основе заданного теста и списка неисправностей. 3) Моделирование неисправностей функциональности на заданном тесте. Группа 2 – реальные эксперименты (на устройстве), без модели функциональности: 4) Синтез теста путем физической эмуляции дефектов в устройстве. 5) Определение списка неисправностей устройства при выполнении диагностического эксперимента. 6) Верификация теста и дефектов в эксперименте на реальном устройстве. Группа 3 – тестовые эксперименты (верификация), без дефектов: 7) Синтез теста путем сравнения результатов моделирования модели и реального устройства. 8) Синтез функциональности по реальному устройству и заданному тесту. 9) Верификация теста и модели функциональности относительно реального устройства с существующими неисправностями. Группа 4 – эксперименты в процессе функционирования, на рабочих воздействиях: 10) Проверка правильности поведения реального устройства на существующих или заданных дефектах. 11) Проверка работоспособности устройства относительно существующей модели в процессе функционирования. 12) Верификация функциональности и списка дефектов относительно поведения реального устройства.

Наиболее популярными задачами из перечисленного выше списка являются: 1, 3, 5, 8, 9. Можно ввести и другую классификацию типов задач, которая дает возможность определить на графе $G = (F, U, T, L)$ все концептуальные пути решения целевых проблем: синтеза тестов, определения модели функциональности, генерирования модели дефектов и проектирования устройства. Все конструкции, представленные в отношениях, обладают замечательным свойством обратимости. Компонент, вычисляемый с помощью двух других, может быть использован в качестве аргумента для определения любого из двух исходных. Потому здесь можно говорить о транзитивной обратимости каждой триады отношений на полном графе, когда по двум любым компонентам всегда и однозначно можно восстановить или определить третий. При этом формат представления каждого компонента должен быть одинаковым по форме и размерности (векторы, матрицы). На основе предложенной метрики и моделей тестирования далее рассмотрены более подробно методы диагностирования дефектов или функциональных нарушений. Модель поиска функциональных нарушений в системе использует уравнение пространства $f(F, T, L, U) = 0 \rightarrow F \oplus T \oplus L \oplus U = 0$, которое трансформируется к виду $L = (T \oplus F) \oplus (T \oplus U)$. Диагностирование дефектов (функциональных нарушений) сводится к сравнению результатов модельного $(T \oplus F)$ и натурального $(T \oplus U)$ экспериментов, которое формирует список функциональных нарушений L , присутствующих в объекте диагностирования. Формула-модель процесса поиска блока F_i с функциональными нарушениями сводится к выбору решения посредством определения хог-взаимодействия между тремя компонентами: $L = F_i \leftarrow [(T \oplus F_i) \oplus_{i=1}^p (T \oplus U_i)] = 0$.

Одним из эффективных и технологичных подходов решения проблемы тестирования является метод сигнатурного анализа. Его математическая основа – получение остатка (сигнатуры) от деления сколь угодно длинной конечной двоичной последовательности на

определенное двоичное число (образующий полином). Практическое решение такой задачи сводится к использованию регистра сдвига с обратными связями из разрядов 7,9,12,16, которые заведены совместно с линией X, являющейся входом сигнатурного анализатора, на сумматор по модулю 2, как показано на рис. 2.

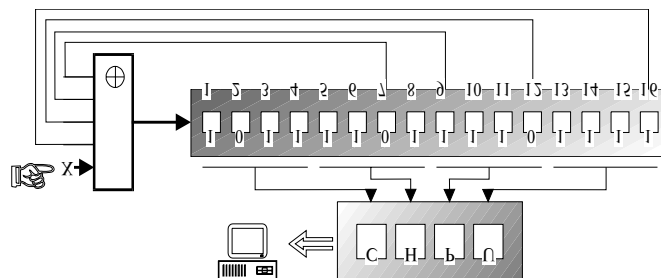


Рис. 2. Структура сигнатурного анализатора

После подачи двоичной последовательности на вход X полученный код-сигнатура состояний разрядов регистра с вероятностью $P=0,9998$ отображает исходный входной вектор произвольной длины. Триггеры регистра априорно должны быть обнулены. Идентификация двоичного вектора длиной не более 2^{16} выполняется с вероятностью $P=1$. Для удобства 16-разрядная двоичная сигнатура записывается в шестнадцатеричном алфавите (0,1,...,9,A,C,F,H,P,U) четырьмя символами, которые служат паспортом для каждого вывода микросхемы или внешних контактов разъема при выполнении тестового диагностирования. Сигнатура контакта, полученная в виде реакции на заданный тест без наличия в схеме неисправностей, называется эталонной. Если в реальном объекте экспериментальная сигнатура на контакте не равна эталонному значению, то рассматриваемый контакт или его предшественники подозреваются неисправными. Количество полных циклов подачи теста для поиска дефекта с использованием сигнатурного анализа в худшем случае может быть равно числу контрольных точек для снятия зондом анализатора экспериментальных сигнатур, расположенных на логическом пути от неисправного выхода до входа. Для уменьшения времени цикла подача теста осуществляется на частотах, близких к рабочим реального ЦУ. Понятие сжатой двоичной последовательности можно использовать в качестве входного слова для входов логических элементов цифровой системы управления. В данном случае имеется структура, представленная на рис. 3, где аналоговые сигналы от датчиков $T = (T_1, T_2, \dots, T_1, \dots, T_n)$ критической системы преобразуются блоком ADC (Analog Digital Coder) в цифровой код произвольной, но фиксированной длины.

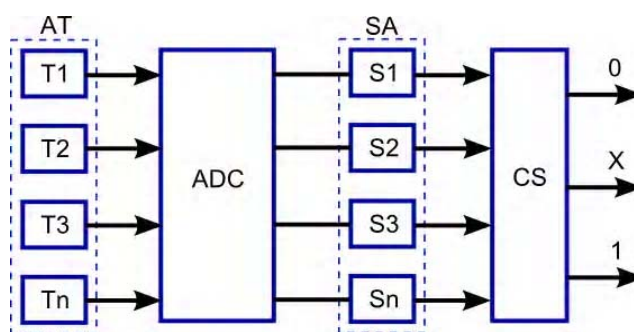


Рис. 3. Структура проверки исправности

Затем двоичные последовательности сжимаются в сигнатуры $T = (S_1, S_2, \dots, S_i, \dots, S_n)$, которые подаются на входы логического устройства управления (CS – Control System). При этом достигается вероятность $P = 0,9998$, что две различные последовательности будут иметь одинаковые сигнатуры на различных входах или на двух соседних тактах. Платой за быстроедействие такой системы контроля является аппаратурная избыточность в виде системы сигнатурных анализаторов на всех входах CS. При этом состояния выходов CS определяются на алфавите $A = \{0,1,X = (S_1, S_2, \dots, S_j, \dots, S_m)\}$, где S_j – шестнадцать битов

двоичного слова, которые интерпретируются как мантисса, определенная в открытом интервале $[0 < S_j < 1]$. Таким образом, естественные флуктуации аналоговых датчиков трансформируются в шестнадцатеричные сигнатуры, которые подаются на логическую схему, где каждая (входная, выходная) переменная есть регистровая, фиксирующая практически любые изменения на аналоговых терминалах, что дает возможность контролировать работоспособность логической структуры управления критическими объектами. Следует также добавить, что неоднородность битовой информации в сигнатуре (слове) позитивно влияет на активизацию выходов, что дает возможность практически в каждом цикле съема информации из терминальных устройств отслеживать работоспособность системы управления критическими объектами.

4. Технологии тестирования систем управления критических объектов

Существующие методы решения данной проблемы, описаны в [10-14] и сводятся к проверке работоспособности элементов оборудования защит путем контроля их реакции на специальные тестовые воздействия, поскольку использование данных о рабочих воздействиях от объекта, как правило, недостаточно. Данные методы реализуют контроль работоспособности на срабатывание либо отдельных блоков и устройств, участвующих в реализации функции защит, либо всего оборудования защит или его части (как минимум инициирующей части защит). Методы обоих типов имеют ряд существенных ограничений и недостатков, подробно рассмотренных и проанализированных в [15].

При этом основное ограничение вытекает из самой структуры элементов оборудования защит («сравнение с эталоном», «и», «или», «2 из 4-х»), построенных на базе дискретных функций, с выходом, определяемым только двумя состояниями 0 (режим ожидания) или 1 (срабатывание защиты). Это в принципе не позволяет обеспечить выполнение непрерывного контроля работоспособности данных элементов на срабатывание. Метод, предполагающий изменение структуры элементов защит, как средство повышения эффективности контроля (проверки) и диагностирования скрытых неисправностей на несрабатывание, предложен в [15, 16]. Основная идея данного метода – функциональный элемент защит на базе арифметических операций формирует значение на всем диапазоне от 0 до 1. При этом в нем отсутствуют какие-либо ветвления (условные переходы), определяющие отличия режима ожидания от режима срабатывания защит. Функция работает одинаково в обоих из этих режимов, что позволяет непрерывно контролировать работоспособность соответствующего функционального элемента. Все функции, которые используются в элементах, строятся на базе арифметических операций (сложение, вычитание, умножение, деление), без применения логических операций и операций отношения, за исключением выходных пороговых элементов формирования команд на исполнительные механизмы (ИМ), что связано с физическими принципами работы приводов ИМ.

В настоящей статье представлены результаты дальнейшего исследования метода, предложенного в [15, 16], в части его теоретического обоснования.

5. Модель процесса совместного тестирования

На рис. 4 представлена концептуальная схема контроля и диагностирования оборудования защит УСБ на базе функциональных элементов, использующих арифметические операции. Здесь $f(x)$ – функция на базе арифметических операций; $fd(x)$ – функция на базе логических операций и операций отношения; Min-Max – диапазон изменения значения технологического параметра; 0-1 – диапазон изменения значений в элементах на базе арифметических операций; 0/1 – дискретные значения (0 или 1) в элементах на базе логических операций и операций отношения; (- - -) – цифровые каналы передачи данных и команд; (—) – нецифровые каналы передачи данных и команд. Основная идея метода – использование в диагностическом оборудовании (В) идентичных ему защит (А) функциональных элементов (блоков), реализованных в программном коде. При этом как в оборудовании защит, так и в диагностическое оборудование поступают функционально-идентичные входные данные (переменные). С помощью арифметических операций обеспечивается возможность вычисления рассогласования по каждой выходной переменной в каждом функциональном блоке, при этом различие сигналов вычисляется в самом диагностическом оборудовании.

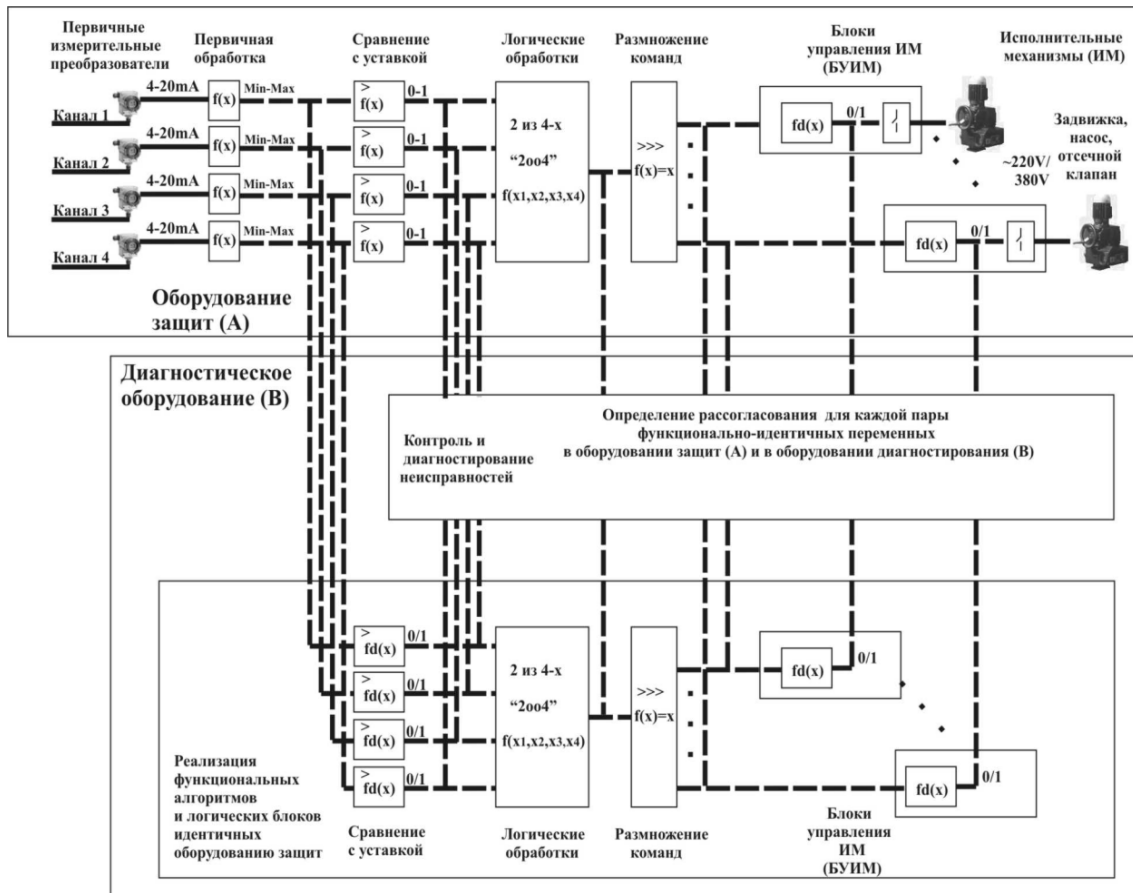


Рис. 4. Концептуальная схема контроля и диагностирования оборудования защит УСБ

Теоретическое обоснование метода может быть сведено к следующим ключевым моментам: 1) разработка базовых функциональных элементов, использующих арифметические операции и позволяющих комбинировать все остальные типы элементов, применяемые в алгоритмах защит; 2) обоснование наличия реакции каждого функционального элемента (изменение его выходного значения) при изменении любого входного непрерывного сигнала, в пределах разрешающей способности АЦП, имеющего корреляцию с данными функциональными элементами по условиям проектных алгоритмов (т.е. обоснование «прохождения» изменения от канала ввода в АЦП через все элементы защит, в которых участвует данный сигнал, до дискретного выходного элемента, формирующего команду защиты на конкретный исполнительный механизм).

Для функциональных элементов сравнения с эталоном («>=», «<=»), использующих операции отношения, реализация с применением арифметических операций может быть представлена в форме линейной функции вида:

$$y = ax + b, \text{ «>=»}: y = \frac{(x - p_2)}{(p_1 - p_2)}, \text{ «<=»}: y = \frac{(p_2 - x)}{(p_2 - p_1)},$$

$$a = \frac{1}{(p_1 - p_2)}, \quad b = \frac{-p_2}{(p_1 - p_2)}, \quad x \in \mathbb{R}, y \in R_{0-1} = [0;1],$$

где a, b – коэффициенты масштабирования (постоянные); p_1 – пороговое значение (эталона); p_2 – предел диапазона измерения, противоположный знаку эталона.

Для функциональных элементов «не», «и», использующих логические операции, реализация с применением арифметических операций может быть представлена как линейная функция следующего вида:

$$\text{«не»}: y = 1 - x,$$

$$\text{«и»}: y = (1/n) * (x_1 + x_2 + \dots + x_n),$$

$$x_1, \dots, x_n, y \in R_{0-1} = [0;1].$$

Все остальные функциональные элементы, использующие логические операции («или», «2 из 4»), могут быть скомбинированы при помощи базовых «не» и «и».

С учетом линейности базовых функциональных элементов, применяющих арифметические операции, аналитическое выражение, описывающее состояние любого i -го элемента в алгоритме защит, может быть представлено как:

$$y_i = a_0 + a_1 x_1 + a_2 x_2 + \dots + a_n x_n, \quad (1)$$

где x_i – значение внешних входных переменных (входные непрерывные сигналы); y_i – значение выхода i -го элемента в алгоритме защит (промежуточного, внутри алгоритма или выходного, в форме выходной команды на механизм).

С учетом зависимости от времени выражение (1) может быть переписано так:

$$y_i(t) = a_0 + a_1 x_1(t) + a_2 x_2(t) + \dots + a_n x_n(t). \quad (2)$$

С учетом (2) «прохождение» (отклик) изменения по входным непрерывным сигналам между моментами времени $t-1$ и t на выходе любого функционального элемента может быть представлено так:

$$\Delta y_i(t) = a_0 + a_1 \Delta x_1(t) + a_2 \Delta x_2(t) + \dots + a_n \Delta x_n(t). \quad (3)$$

В случае изменения между моментами времени $t-1$ и t значения только одного входного непрерывного сигнала из n всех возможных (изменения по остальным входным сигналам равны 0) выражение (3) может быть преобразовано в следующий вид:

$$\Delta y_i(t) = a_k \Delta x_k(t),$$

$$\Delta x_1(t) = \Delta x_2(t) = \dots = \Delta x_{k-1}(t) =$$

$$\Delta x_{k+1}(t) = \dots = \Delta x_n(t) = 0. \quad (4)$$

Формула (4) описывает аналитическое выражение для зависимости между величиной изменения входного непрерывного сигнала Δx_k и соответствующего ему изменения выхода функционального элемента, имеющего корреляцию с данным входом по условиям проектного алгоритма защит УСБ. Последнее теоретически подтверждает возможность контроля и диагностирования всех функциональных элементов оборудования защит УСБ за счет наличия реакции («отклика») данных элементов (изменение их выходов) при изменении значений соответствующих входных сигналов, в том числе за счет наличия постоянно присутствующих «технологических» и «электрических» шумовых колебаний.

Проведем анализ предлагаемого метода на примере одной из 4-канальных защит САОЗ в соответствии с рис. 5 при наличии неисправности в элементе оборудования, реализующем функцию мажоритарной обработки “2оо4” (2 из 4-х).

Строки схемы, относящиеся к части А, соответствуют значениям переменных в оборудовании защит, а аналогичные строки (с подстройкой рассогласования) в части В соответствуют значениям этих же переменных, но в диагностическом оборудовании. При этом рассматриваемая неисправность в элементе мажоритарной обработки “2оо4” (2 из 4-х) характеризуется формированием выходного значения, не соответствующего входным значениям от 4-х каналов (значение на выходе 0,6256995 меньше номинального 0,8256995, т.е. того, которое должно быть при соответствующей комбинации входных значений). Данный тип неисправности может привести к отказу оборудования защит типа «несрабатывание по требованию» при наличии исходного события и поэтому является наиболее критичным в

части влияния на безопасность. Как следует из рис. 5, признак неисправности “fault” сформирован до наступления исходного события, по результату рассогласования значений выходных переменных функционального блока мажорирования “2оо4” (2 из 4-х) в оборудовании защит (А) с одной стороны и оборудовании диагностирования (В) – с другой.

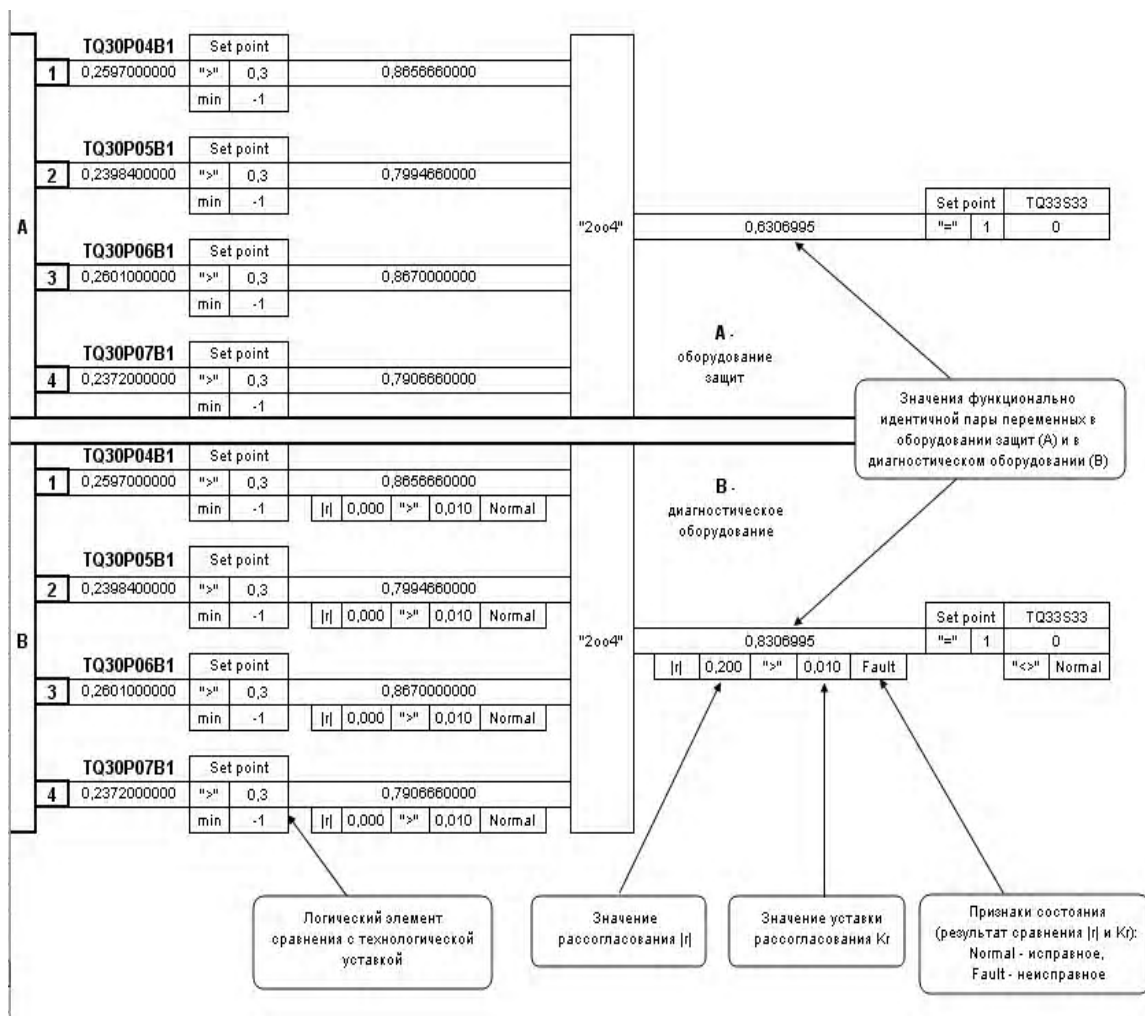


Рис. 5. Диагностирование неисправности типа «несрабатывание» в элементе “2оо4” («2 из 4-х») оборудования защит УСБ (4-канальная защита САОЗ по давлению >0,3 кгс/см²)

6. Заключение

Разработанный метод повышения тестопригодности оборудования защит, использующий функциональные элементы на базе арифметических операций, характеризуется такими особенностями:

1) Обеспечивает проверку и диагностирование следующих видов скрытых неисправностей типа «несрабатывание»: дефекты функциональных элементов, характеризуемые несоответствием значений входных и выходных переменных проектному или конструкторскому алгоритму функционирования (с учетом предыстории входных, выходных значений, в случае наличия «памяти» в алгоритме функционального элемента); неисправности связей между функциональными элементами, характеризуемые отсутствием или искажением данных между источником и приемником.

2) Повышение тестопригодности обеспечивается за счет использования в функциональных элементах оборудования защит только арифметических операций (сложение, вычитание, умножение, деление) без применения логических операций и операций отношения.

3) Метод обеспечивает контроль прохождения любого изменения значения входного сигнала, в пределах разрешающей способности используемых АЦП, от входа через все функциональные элементы («сравнение с эталоном», «и», «или», «2 из 4»), в которых

участвует данный сигнал, до дискретного элемента управления исполнительным механизмом.

4) Предложена сигнатурная структура контроля работоспособности логической схемы управления критическими объектами, основанная на преобразовании естественных флуктуаций аналоговых датчиков в шестнадцатеричные сигнатуры, подаваемые на логическую схему, где каждая (входная, выходная) переменная есть регистровая, активизирующая практически любые изменения от входов до выходов.

Список литературы: 1. *Безопасность атомных станций. Информационно-управляющие системы* / М.А. Ястребенецкий, В.Н. Васильченко, С.В. Виноградская и др. К.: Техника, 2004. 470 с. 2. *Instrumentation and control systems important to safety in Nuclear Power Plants: Nuclear Energy Series* / International Atomic Energy Agency. Vienna: IAEA, 2002. No. NS-G-1.3. 91 p. 3. *Safety of Nuclear Power Plants: Design, Safety Standards Series* / International Atomic Energy Agency. Vienna: IAEA, 2000. No. NS-R-1. 125 p. 4. *Software for Computer Based Systems Important to Safety in Nuclear Power Plants: Safety Standards Series* / International Atomic Energy Agency. Vienna: IAEA, 2000. No. NSG-1.1. 150 p. 5. *International Electrotechnical Commission (IEC) 60880 – 2004, Nuclear Power Plants — Instrumentation and Control Systems Important to Safety — Software Aspects for Computer-Based Systems Performing Category A Functions*. 6. *International Electrotechnical Commission (IEC) 60987 – 2007, Nuclear Power Plants – Instrumentation and Control Important to Safety – Hardware Design Requirements for Computer-Based Systems*. 7. *Institute of Electrical and Electronic Engineers (IEEE) 7-4.3.2, Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations*. 8. *Макдональд Д.* Промышленная безопасность, оценивание риска и системы аварийного останова: Пер. с англ. / Д. Макдональд. М.: ИДТ, 2007. 409 с. 9. *Смит Д.* Безотказность, ремонтпригодность и риск: Пер. с англ. / Д. Смит. М.: ИДТ, 2007. 432 с. 10. *НП 306.2.141-2008.* Общие положения безопасности атомных станций. К: ГКЯРУ, 2008. 42 с. 11. *Protecting against common cause failures in Digital I&C Systems of Nuclear Power Plants: Nuclear Energy Series* / International Atomic Energy Agency. Vienna: IAEA, 2009. No. NP-T-1.5. 65 p. 12. *Ястребенецкий М.А.* Информационные и управляющие системы АЭС Украины: результаты и проблемы / М.А. Ястребенецкий // Проблемы обеспечения безопасности информационных и управляющих систем АЭС // Сб. науч. тр. Одесса: «Астропринт», 2010. С. 9-19. 13. *Modern Instrumentation and Control for Nuclear Power Plants: Technical Reports Series* / International Atomic Energy Agency. Vienna: IAEA, 1999. No. 387. 629 p. 14. *Application of the Single Failure Criterion: Safety Series* / International Atomic Energy Agency. Vienna: IAEA, 1990. No. 50-P-1. 134 p. 15. *Герасименко К.Е.* Методы непрерывного контроля и диагностирования оборудования управляющих систем безопасности энергоблоков АЭС по функции защит / К.Е. Герасименко // *Радіоелектронні і комп'ютерні системи*. 2010. №3 (44). С. 152-156. 16. *Герасименко К.Е.* Использование непрерывных функций в элементах оборудования защит АЭС для диагностирования неисправностей типа «несрабатывание по требованию» / К.Е. Герасименко // *Радіоелектронні і комп'ютерні системи*. 2011. №1 (49). С. 29-33. 17. *Бондаренко М.Ф., Кривуля Г.Ф., Рябцев В.Г., Фрадков С.А., Хаханов В.И.* Проектирование и диагностика компьютерных систем и сетей. К.: НМЦ ВО. 2000. 306 с. 18. *Хаханов В.И., Литвинова Е.И., Чумаченко С.В., Гузь О.А.* Логический ассоциативный вычислитель // *Электронное моделирование*. 2011. № 1(33). С. 73-89. 19. *Hahanov V., Wajeb Gharibi, Litvinova E., Chumachenko S.* Information analysis infrastructure for diagnosis // *Information an international interdisciplinary journal*. 2011. Japan. Vol.14. № 7. P. 2419-2433.

Поступила в редколлегию 16.12.2011

Герасименко Константин Евгеньевич, заведующий отделом информационно-управляющих систем ЧАО «СНПО «Импульс»». Научные интересы: техническая диагностика цифровых систем управления объектами с повышенными требованиями к безопасности. Адрес: Украина, 93405, Северодонецк, пл. Победы 2, тел. 60194. E-mail: gerasyumenko.k.e@yandex.ua.

Хаханов Владимир Иванович, декан факультета КИУ ХНУРЭ, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanov@kture.kharkov.ua.

ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ В СТОХАСТИЧЕСКИХ СИСТЕМАХ УПРАВЛЕНИЯ ЗАПАСАМИ

Рассматривается подход к решению задачи управления запасами с использованием стохастического динамического программирования и техники обучения с подкреплением, совместимой с несепарабельным критерием. В отличие от известных алгоритмов предлагаемый алгоритм обучения формирует реальное распределение прогнозируемых затрат, соответствующих рассматриваемым состояниям. Идея состоит в аппроксимации функции оценивания с помощью регистрируемых затрат. Используемая при этом стратегия базируется на робастных процедурах, гарантирующих сходимость вычислительного алгоритма. Приводится пример использования предложенного подхода для управления гидроресурсами электростанций.

1. Введение

Регулирование уровня запасов в сложных технологических комплексах зачастую определяет динамику их структуры (при этом состояние определяется уровнем запасов), а неопределенность данных (запросы, поступления, стоимость, наличие средств производства) делает структуру стохастичной. Оптимальное управление такой системой требует разработки стратегии использования запасов, позволяющей минимизировать математическое ожидание стоимости (или максимизировать математическое ожидание выигрыша) на некоторой совокупности стратегий. Таким образом, речь идет о том, чтобы знать, в какой пропорции надо использовать запасы для удовлетворения запросов и в какой степени желательно консервировать эти запасы для последующего времени [1].

Классическое решение для оптимизации выбора стратегии по сепарабельному критерию (т.е. когда стратегия может строиться шаг за шагом во времени и изменяться) может основываться на динамическом программировании (в частности, математическое ожидание является сепарабельным). К сожалению, в большинстве случаев оптимизация математического ожидания чересчур рискованная. Таким образом, целесообразно: выбрать критерий оптимизации, отличный от математического ожидания и обеспечивающий робастные решения; обеспечить возможность оптимизации такого критерия, даже если он не совместим с классической техникой динамического программирования. К классическим статистическим характеристикам случайной переменной величины C , представляющим интерес для рассматриваемой задачи, кроме математического ожидания $E(C)$ и дисперсии $\text{var}(C) = E(C - E(C))^2$ следует отнести распространенные в системах управления рисками характеристик Risk-At-Value (RaV) и Value-At-Risk (VaR). RaV означает (для некоторого порога затрат C') вероятность того, что $C \geq C'$, а VaR для заданного порога риска соответствует минимальному C' , для которого $P(C > C') \leq \alpha$. Представления типа RaV и VaR являются наиболее полными в смысле отображения всего распределения затрат. Стохастическое динамическое программирование служит принципом классической декомпозиции для динамической оптимизации. Оно используется для оптимизации по всем сепарабельным критериям. В частности, одним из таких критериев является математическое ожидание. Однако, если принимать во внимание оценки риска VaR , то возникает проблема несепарабельности, при которой нельзя применить стандартное стохастическое динамическое программирование. Эта статья посвящена применению техники обучения с подкреплением, совместимой с несепарабельным критерием [2]. На примере обучения с подкреплением целесообразно определить стратегию, оптимизирующую компромисс «математическое ожидание/риск» несепарабельного типа $(1 - \alpha)E + \alpha VaR$.

Целью данной работы является решение задачи определения стратегии управления запасами в стохастических системах с применением методов машинного обучения с подкреплением и динамического программирования.

2. Общая схема управления запасами с применением стохастической оптимизации

Общая схема стохастической динамической оптимизации и схема ее применения для управления запасами приведены на рис.1 (в левом и правом фрагментах соответственно). В самом общем ракурсе проблема стохастической динамической оптимизации может быть проиллюстрирована эволюционирующей во времени системой, которая характеризуется некоторым внутренним состоянием и пытается оптимизировать свою траекторию по заданному критерию.

Для описания такой проблемы используются:

– стохастический процесс $p(t)$ (здесь рассмотрим дискретное время от 0 до T): $p(0), \dots, p(T)$;

– уравнение эволюции системы: $x_{t+1} = f(x_t, u(p(t), t, x_t), p(t))$ и стратегия (регулятор) $u(\cdot, \cdot, \cdot)$, которая пытается на нее повлиять так, чтобы минимизировать математическое ожидание $c_1(x_1, u_1) + c_2(x_2, u_2) + \dots + c_T(x_T, u_T)$ (u_t означает $u(p(t), t, x_t)$).

Таким образом, можно сказать, что $u()$ является регулятором, минимизирующим математическое ожидание суммарных затрат.

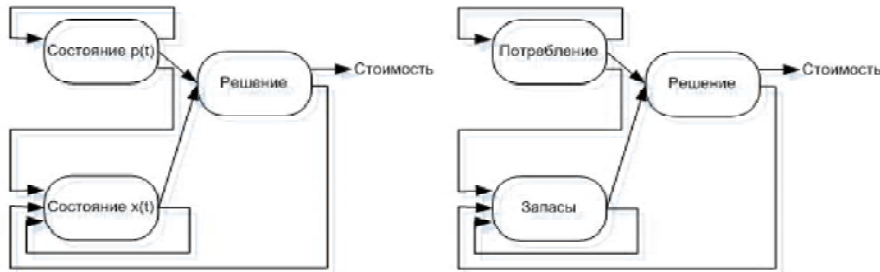


Рис. 1. Общая схема стохастической динамической оптимизации для управления запасами

В основе принципа декомпозиции Беллмана лежит способ определения $V(p(t), t, x_t)$ и, соответственно, математического ожидания $c_t(x_t, u_t) + c_{t+1}(x_{t+1}, u_{t+1}) + \dots + c_T(x_T, u_T)$ с заданными $p(t)$ и $x(t)$ для нахождения оптимальной стратегии $u()$; т.е. определение такого V , чтобы для оптимального u выполнялось следующее равенство:

$$V(\tilde{p}, t, \tilde{x}) = E(c_t(x_t, u_t) + c_{t+1}(x_{t+1}, u_{t+1}) + \dots + c_T(x_T, u_T) | p(t) = \tilde{p}), \quad (1)$$

где $x_t = \tilde{x}$ и $x_{t+1} = f(x_t, u(p(t), t, x_t), p(t))$.

Уравнение (1) определяет V как функцию величины, зависящей от стратегии u . Можно показать, что каждой оптимальной стратегии $u()$ соответствует своя величина $V()$. Таким образом, достаточно определить для решения задачи в каждый момент времени $u(p(t), t, x_t) \in \arg \min_u c_t(x_t, u) + V(p(t+1), t+1, x_{t+1})$. Функция $V(\cdot, \cdot, \cdot)$ называется функцией Беллмана или функцией валоризации, или же, с позиций обучения с подкреплением, критической функцией или функцией стоимости. Здесь мы будем рассматривать $V(\cdot, \cdot, \cdot)$ с одной стороны как математическое ожидание, а с другой – как компромисс вида $(1-\alpha)E + \alpha \text{VaR}$, позволяющий решать задачу оптимизации не только по математическому ожиданию затрат, но и по риску.

Такая задача является актуальной для всех систем управления запасами, независимо от их технологического назначения. В частности, в работе [2] рассматривается управление гидравлическими запасами при использовании различных типов электростанций для удовлетворения стохастических запросов на поставку электроэнергии. При этом необходимо знать в каждый конкретный момент времени (в функции предварительно определенной модели и для текущих уровней запасов), какая часть запросов должна быть удовлетворена с помощью тепловых электростанций (ТЭЦ), а какая часть – с помощью гидроэлектрических станций (ГЭС).

Формально стохастический процесс соответствует заявкам на электроэнергию $d(t)$; продукция ТЭЦ $p_c(t)$ выбирается в соответствии с принимаемой стратегией (из диапазона $p_{\min}(t) - p_{\max}(t)$, определяемого эксплуатационными ограничениями теплоцентрали); продукция ГЭС $p_h(t)$ выбирается в соответствии с принимаемой стратегией (из диапазона 0–

p_h^{\max}); уравнение эволюции запасов имеет следующий вид: $stock(t+1) = stock(t) + a(t) - p_h(t) - dev(t)$, где $a(t)$ – приток воды. Стратегия должна удовлетворять следующим ограничениям: $p_c(t) + p_h(t) = d(t)$, $stock(t) \geq 0$ и $stock(t) \leq stock_{\max}(t)$. Функция стоимости c определяется суммой $c(t, p_c(t))$, где $c(t, \cdot)$ – функция стоимости, зависящая от t (возрастающая и выпуклая), а $c(stock(t))$ – функция штрафа за конечные запасы (убывающая). Стоимости $c_t(x_t, u_t)$ равны $c(t, p_c(t))$ для $t < T$ и $c(stock(T))$ для $t = T$. Состояние определяется парой $(d_1(t), stock(t))$ (т.е. внутренним состоянием $stock(t)$ и состоянием стохастического процесса $d_1(t)$). Стратегия $u()$ определяет $p_c(t)$ и $p_h(t)$. Динамическое программирование, используемое для поиска стратегии, работает для внутреннего состояния x_t и стохастического процесса $d_1(t)$ ($a(t)$ – детерминированная составляющая).

Рассмотрим возможность применения методов обучения с подкреплением для решения рассматриваемой задачи. Основная идея таких методов состоит в следующем: формируются модуль-актер и/или начальный модуль-критик; реализуется симуляция текущих действий модуля-актера; если используется текущий модуль-критик (алгоритмически участвующий в симуляции), модифицируются решения актера в целях оптимизации заданной функции; производится возврат к этапу симуляции.

В методах обучения с подкреплением (Q-обучение, алгоритм TD(λ)), в настоящее время широко используемых для оптимизации матожиданий, начинают применять понятия риска. В настоящей статье предлагается рассмотреть использование компромиссного критерия «матожидание»/«VaR».

Сравнивая методы обучения с подкреплением с методами классического динамического программирования, можно отметить следующее:

- классические процедуры стохастической оптимизации на основе динамического программирования обладают свойствами робастности; они реализуются с помощью алгоритмов, время вычисления для которых может быть определено заранее, а результаты вычислений являются достаточно устойчивыми;

- обучение с подкреплением хорошо подходит для работы с предысторией и, в частности, не требует структурного определения стохастического процесса, тогда как методы, основанные на стохастической оптимизации цепной предыстории с помощью динамического программирования, предполагают наличие модели;

- обучение с подкреплением позволяет естественным способом учитывать более совершенные характеристики распределения стоимости, чем сепарабельные критерии; в частности, матожидание, взвешенное риском, а не просто матожидание. Это является принципиальным моментом для данной статьи; здесь используются симуляции для представления распределения будущих стоимостей, отличные от средних будущих стоимостей. Метод, основанный только на численных переоценках (например, динамическое программирование), не может здесь работать, потому что критерий не является сепарабельным. Поэтому целесообразно разработать итеративный метод, позволяющий гарантировать сходимость функции оценивания на конечном горизонте;

- методы с подкреплением более приспособлены для работы с недискретизированными данными или с большим числом переменных состояния, когда они связаны с методами экстраполяции. Следует отметить, что объединение динамического программирования и методов, использующих экстраполяцию, все-таки менее удобно, чем обучение с подкреплением.

3. Комбинированные алгоритмы

Рассмотрим вопросы выбора схемы и алгоритмов оценивания эффективности стратегий принятия решений в системах управления запасами с использованием комбинированного критерия.

Алгоритм TD(λ) был уже успешно проверен на практике для решения широкого круга задач; если требуется симуляция, то он хорошо сочетается с комплексными ограничениями с малыми затратами на программирование и время вычислений. Алгоритм TD(λ) основан на использовании функции оценивания (аналога оценок Беллмана) для симуляций.

В частности, он может базироваться на симуляции различных форм величины λ (параметра, принимающего значения в диапазоне $[0,1]$). Мы используем алгоритм TD(1), который позволяет удобное введение понятия риска. В отличие от других алгоритмов TD(λ) он формирует реальное распределение будущих стоимостей, соответствующих рассматриваемым состояниям, а не будущих стоимостей, полученных по функции оценивания; оценивание «VaR» является, таким образом, прямым. Предлагаемая идея состоит в аппроксимации $V(\cdot, t, \cdot)$ с помощью регистрируемых стоимостей $c_t(x_t) + c_{t+1}(x_{t+1}) + \dots + c_T(x_T)$. Используемая при этом стратегия является «неоптимистической», т.е. базирующейся на медленных процедурах (точнее, последовательно осуществляется некоторое количество симуляций перед началом использования функции оценивания). Такой метод является робастным по отношению к сходимости выбираемых стратегий, так как функция оценивания может сходиться без обязательной сходимости стратегии.

Введение понятия риска можно осуществить посредством использования представления типа «VaR». Модуль, позволяющий экстраполировать «VaR» для непрерывных уровней запасов на основе конечного числа примеров, предполагает необходимость введения ограничений на функции оценивания, что приводит к задаче выпуклой квадратичной оптимизации с линейными ограничениями. Структура реализации предлагаемого подхода приведена на рис. 2.



Рис. 2. Комбинированная схема формирования функции оценивания

Стохастическое динамическое программирование реализует первую функцию оценивания в случае без «VaR» и с немного упрощенным стохастическим процессом, чтобы реализовать первую функцию оценивания (т.е. начальный пункт итераций обучения с подкреплением). Комбинирование обучения с подкреплением и динамического программирования позволяет использовать гарантированную устойчивость динамического программирования и возможность методов обучения с подкреплением трансформировать оптимальное управление по матожиданию в стратегию, использующую понятие риска (после обучения с подкреплением во время фазы симуляции). Модуль экстраполяции реализует здесь экстраполяцию выпуклыми функциями, без критериев регуляризации (т.е. без штрафа на вторую производную или других штрафов этого типа). Динамическое программирование является реализуемым только для предварительных расчетов, не принимающих во внимание понятие риска; будучи теоретически факультативным, оно тем не менее обеспечивает прочную базу для операций пересчета, компенсируя, как правило, трудности параметризации обучения с подкреплением. Модуль «Критик», располагающий полной траекторией, позволяет учитывать более понятия RaV и VaR. Можно рассмотреть, например, случайную переменную C , ассоциируемую со стоимостью управления; оптимизацию $C1 = (1 - \alpha)E(C) + \alpha P(C > K)$, где E – символ матожидания, а P – вероятность, т.е.

учет RaV ; или оптимизация $C2 = (1 - \alpha)E(C) + \alpha K$, где значения K такие что $P(C > K) = 1 - \eta$, т.е. учет VaR . В случае RaV , строго говоря, надо было бы для оптимизации, например, риска превышения стоимости управления величины K , использовать функции, основанные не только на состоянии и времени, но также и на предыдущей стоимости управления. В случае VaR это не является необходимым: критерий $C2$ изменяет в функции предыдущей стоимости только одну константу и, соответственно, предыдущая стоимость не влияет на модуль «Актер».

Фиксированными являются следующие параметры: частота принятия решений (количество симуляций между каждым $V()$ и π ; в соответствии с общепринятой терминологией это «степень оптимизма» метода); шаг спуска и его эволюция во времени; порог доверия применительно к понятию риска (обычно 5%); параметр компромисса между матожиданием и VaR .

Модуль «Критик» должен иметь возможность осуществлять представления в виде ломаной кривой функции оценивания $V(p, t, x)$ для заданного актера. Такая линия позволяет реализовать оптимизацию для выбора решения, соответствующую квадратичной выпуклой оптимизации с линейными ограничениями. В нашем случае p – это величина, характеризующая стохастический процесс (заявка), t – дата, x – уровень запасов. Таким образом, для каждого этапа $n = 1, 2, 3, \dots$ необходимо произвести следующие действия:

- провести симуляции (процедура симуляции и особенно дискретизации будет подробнее приведена далее);

- получить «примеры» (p, t, x, c) , где t – дата, p – состояние стохастического процесса, x – уровень запасов, c – стоимость (по результатам проведенной симуляции);

- аппроксимировать матожидание будущей стоимости ломаными линиями (для шага времени и для состояния процесса: ломаная линия соединяет таким образом пары (уровень запасов, средняя будущая стоимость));

- аппроксимировать VaR ломаными линиями (для шага времени и для состояния процесса: ломаная линия соединяет таким образом пары (уровень запасов, квантиль стоимости)) и определить итоговые величины, комбинируя VaR и матожидание.

Обе аппроксимации (VaR и матожидание) получают итеративным путем: на каждой итерации определяются симуляции VaR и матожидания для полученной стратегии управления; затем стратегия управления адаптируется для полученной функции оценивания. Пусть V_n^E и V_n^V – функции оценивания VaR и матожидания соответственно, которые были получены на n -м этапе (в виде ломаных линий). На основании V_n^E и V_n^V строятся V_{n+1}^E и V_{n+1}^V .

Пусть $V_{n+1}^E(p, t, \cdot, c)$ и $V_{n+1}^V(p, t, \cdot, c)$ – кусочно-линейные функции (с фиксированными точками изломов CS_i), которые минимизируют (с указанными ниже ограничениями) сумму термов $T1, T2, T3, T4$:

$$T1 = \alpha_1 \sum_{(p,t,x,c)} (c - V_{n+1}^E(p, t, x))^2 \quad (2)$$

(терм вызова квадроплета (p, t, x, c) , используемый в симуляции с применением матожидания);

$$T2 = \alpha_2 \sum_{(p,t,cs_i)} (V_n^E(p, t, cs_i) - V_{n+1}^E(p, t, cs_i))^2 \quad (3)$$

(терм вызова симуляции с предыдущего матожидания);

$$T3 = \alpha_3 \sum_{(p,t,abs_j,c)} (V_{n+1}^V(p, t, abs_j) - VaR_j)^2, \quad (4)$$

(терм вызова оцениваемого VaR , где величины abs_j – аппроксимации VaR , полученные по данным (p, t, x, c));

$$T4 = \alpha_4 \sum_{(p,t,cs_i)} (V_n^V(p, t, cs_i) - V_{n+1}^V(p, t, cs_i))^2, \quad (5)$$

(терм вызова предыдущей функции VaR).

Таким образом можно найти функцию, «не слишком отличную» от предыдущей функции и «не слишком далекую» от результатов симуляции. При этом принимаются во внимание следующие ограничения: $V_{n+1}^E(p, t, \cdot)$ и $V_{n+1}^V(p, t, \cdot)$ – убывающие выпуклые функции; неравенство $V_{n+1}^V \geq V_{n+1}^E$ в каждой точке (это, строго говоря, не гарантируется, но вероятность этого высока); задача не очень сложна и использование «пар» не является, следовательно, проблематичным; V_{n+1} получают в виде линейной комбинации V_{n+1}^E и V_{n+1}^V .

Выбор параметров $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ производится следующим образом: α_1 и α_3 равны 1, деленной на число термов в T1 и T3 соответственно; α_2 и α_4 равны числу игр в N уже прошедших симуляциях для рассматриваемого шага времени, деленному на число термов в T2 и T4 соответственно. Такой выбор аналогичен технике градиентного спуска; тем не менее следует понимать, что актер эволюционирует после каждого действия критика, а алгоритм градиентного спуска является разновидностью алгоритма с фиксированной точкой.

В базовом варианте предлагаемый алгоритм определяется такой последовательностью действий:

1. Определение первой функции оценивания с помощью динамического программирования.
2. Симуляции с помощью стратегии оптимизации, основанной на функции оценивания (т.е. во время симуляций получается решение, которое оптимизирует предсказанную стоимость с помощью функции оценивания).
3. Определение новой функции оценивания, которая соединяется аппроксимативно по результатам симуляции.
4. Переход к 2, в случае исчерпания лимита времени.

Формализация алгоритма зависит от параметров $1 \leq DR$ (DR – размер сигнала подкрепления), $1 \leq \Delta \leq DR$ и $0 \leq \tau \leq 1$, где τ – уровень рассматриваемого риска:

Опишем подробнее этапы алгоритма.

Для всех T (от начального T , равного 1) с шагом, равным Δ , выполнить следующие действия: симулировать N траекторий с заданным шагом, начиная с T : начальные уровни запасов для этих траекторий, начиная с T , распределяются не поровну, так как маленькие уровни запасов являются более критичными. Пусть $L = (p_i, stock_i, pdt_i, c_i)$ – перечень квадроуплетов (уровень заявок, начальный запас, шаг времени, стоимость (по результатом симуляции между шагом времени pdt и шагом времени T)), полученных в результате симуляции; L содержит $N \times DR$ пар, так как каждая симуляция позволяет получить будущую стоимость для каждого значения pdt между началом ($N0$) и ($N0 + DR - 1$).

Для всех k между 0 и $DR - 1$ и всех состояний p стохастического процесса между 0 и ($DR - 1$) в момент ($k + N0$) выполнить следующие действия: сформировать список L' из таких элементов L , для которых $p_i = p$ и $pdt_i = debut + k$. Пусть VaR_j соответствует квантилю $1 - \tau$ из c_i для $stock_i$ в интервале $[a_j, a_{j+1}]$ среди элементов одного подмножества (abs_j, VaR_j) для рассматриваемых значений k и p . Поместить каждый (abs_j, VaR_j) в определенный терм T3. Сформировать термы T1, T2, T4. Оптимизировать T1 + T2 + T3 + T4 для получения новой функции оценивания. Симулировать большое число траекторий и вычислить полученные индикаторы риска (матожидание, кривая VaR).

4. Анализ сходимости комбинированного алгоритма

Покажем, что функция стоимости для достаточно большого N стремится к функции оценивания оптимальной стратегии, полученной по рассмотренному методу, для нормы L^2 . Для этапа алгоритма, соответствующего $debut = T - 1$, сходимость соответствующей рекуррентной процедуры является очевидной. Рассмотрим гипотезу такой сходимости для этапа $debut + 1$ (с точностью ϵ). Проблема оптимизации с ограничениями по T1 (ограничения по T3 и T4 можно не рассматривать, поскольку они имеют отношение лишь к минимизации математического ожидания, а влияние T2 является слабым для большого

числа симуляций) сводится к асимптотическому приближению функции оценивания F к кусочно-линейной функции, соответствующей оптимальной стратегии.

Если шаг дискретизации пространства поиска выбрать в соответствии с рекомендациями, приведенными в [2], то можно получить точность $\epsilon' + \epsilon$ для сколь угодно малого ϵ' при ограниченном N . При этом гарантируется, что функции оценивания после оптимизации T_1 для $k=0$ будут наилучшими (для нормы L^2). Отметим необходимость выполнения условия $\Delta=1$ в случае учета рисков при реализации стратегий управления. При этом среднее значение RaV для x в интервалах $[a_j, a_{j+1}]$ является в асимптотике однозначно определяемым, так как число соответствующих дискрет по определению пропорционально отношению N/K . Поскольку функция VaR в общем случае инверсна функции RaV, то и оценки также являются сходящимися. Таким образом, для достаточно большого K (K - число сегментов $[a_j, a_{j+1}]$) выполняется условие $\min(K, 1/\max(cs_{i+1} - cs_i)) \rightarrow \infty$ (при $N \rightarrow \infty$), а следовательно, кривая значений VaR, полученная по рассмотренному алгоритму, сходится к реальной кривой VaR, построенной по результатам проведения и анализа большого числа стандартных симуляций.

5. Результаты моделирования

Моделирование рассмотренного метода проводилось для тестовой задачи управления запасами гидроресурсов электростанций с ограничениями, приведенными в [3]. Были получены кривые для комбинированного критерия $(1-\alpha)E + \alpha VaR$ при $+\alpha = 0; 0.1; 0.2; 0.4; 0.6; 0.8$ с использованием 1800 сценариев. При этом наилучшие результаты соответствуют значению $+\alpha = 0.6$. Анализ результатов показал существенное снижение уровня риска получения неудовлетворительных решений (в среднем на 55 %) при использовании комбинированного метода по сравнению с результатами, полученными с применением стандартной процедуры динамической оптимизации с сепарабельным критерием.

6. Выводы

Научная новизна полученных результатов состоит в совершенствовании методов получения оптимальных стратегий принятия решений при управлении запасами на основе применения комбинированного метода, использующего положительные свойства алгоритмов машинного обучения с подкреплением и динамической оптимизации Беллмана, который позволяет эффективно описывать динамику развития исследования стохастического процесса с учетом возможного риска и существующих ограничений.

Эффективность реализации предложенного подхода во многом зависит от правильного выбора коэффициентов в несепарабельном критерии $(1-\alpha)E + \alpha VaR$.

Практическая значимость заключается в возможности практической реализации предложенного подхода в интеллектуальных системах принятия решений с учетом возможных рисков, в частности, в системах управления запасами гидроресурсов электростанций.

Перспективным представляется развитие предложенного подхода для фиксированных уровней риска и высокого уровня неконтролируемых возмущений.

Список литературы: 1. *Dempster M. Intraday. FX trading: An evolutionary reinforcement learning approach. Intelligent data engineering and automated learning // Proceedings of the IDEAL 2002 International Conference. 2002. P. 347-358.* 2. *Гришко А., Удовенко С., Чалая Л. Применение гибридных методов машинного обучения в компьютерных трейдинговых системах // Системные технологии. 2010. №3(68). С. 84-92.* 3. *Sarma M., Thomas S., Shah A. Performance Evaluation of Alternative VaR Models. Mumbai: Indira Gandhi Institute of Development Research, 2000. P. 5-10.*

Поступила в редколлегию 11.12.2011

Гришко Андрей Александрович, аспирант кафедры ЭВМ ХНУРЭ. Научные интересы: методы вычислительного интеллекта, машинное обучение в системах интеллектуальной обработки данных. Адрес: Украина, 61166, Харьков, пр. Ленина, 14.