ISSN 1684–8853

ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ

НАУЧНЫЙ ЖУРНАЛ

5(48)/2010

НАУЧНЫЙ ЖУРНАЛ

ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ

5(48)/2010

РЕЦЕНЗИРУЕМОЕ ИЗДАНИЕ

Учредитель
ОАО «Издательство “Политехника”»

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

Зам. главного редактора
Г. Ф. Мощенко

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

Редакционная коллегия:
В. Г. Анисимов,
доктор технических наук, профессор
Е. А. Крук,
доктор технических наук, профессор
В. Ф. Мелехин,
доктор технических наук, профессор
А. В. Смирнов,
доктор технических наук, профессор
В. И. Хименко,
доктор технических наук, профессор
А. А. Шалыто,
доктор технических наук, профессор
А. П. Шепета,
доктор технических наук, профессор
З. М. Юлдашев,
доктор технических наук, профессор

Редактор: А. Г. Ларионова
Корректор: Т. В. Звертановская
Дизайн: А. Н. Колешко, М. Л. Черненко
Компьютерная верстка: С. В. Барашкова
Ответственный секретарь: О. В. Муравцова

Адрес редакции: 190000, Санкт-Петербург,
Б. Морская ул., д. 67, ГУАП, РИЦ
Тел.: (812) 494-70-44
Факс: (812) 494-70-18
E-mail: 80x@mail.ru
Сайт: www.i-us.ru

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

Журнал входит в «Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные научные результаты диссертации на соискание ученой степени доктора и кандидата наук».

Журнал распространяется по подписке. Подписку можно оформить через редакцию, а также в любом отделении связи по каталогам: «Роспечать»: № 48060, № 15385; «Пресса России»: № 42476.

© Коллектив авторов, 2010

ЛР № 010292 от 18.08.98.
Сдано в набор 23.08.10. Подписано в печать 12.10.10. Формат 60×84 1/8.
Бумага офсетная. Гарнитура SchoolBookC. Печать офсетная.
Усл. печ. л. 11,4. Уч.-изд. л. 14,6. Тираж 1000 экз. Заказ 477.
Оригинал-макет изготовлен в редакционно-издательском центре ГУАП.
190000, Санкт-Петербург, Б. Морская ул., 67.
Отпечатано с готовых диапозитивов в редакционно-издательском центре ГУАП.
190000, Санкт-Петербург, Б. Морская ул., 67.

УДК 510.647

ОБ ОДНОМ МЕТОДЕ ВЫЧИСЛЕНИЯ ВЕРОЯТНОСТЕЙ ЛОГИЧЕСКИХ ФУНКЦИЙ

В. В. Дубаренко,
доктор техн. наук
В. Г. Курбанов,
канд. физ.-мат. наук
А. Ю. Кучмин,
канд. техн. наук
Институт проблем машиноведения РАН

Приводится комбинаторный метод вычисления вероятностей сложных логических функций, причем элементы логических функций расположены в некотором лексикографическом порядке, т. е. нет необходимости хранить их в памяти ЭВМ в символьном виде. Описываются приближенные методы вычисления вероятности сложной логической функции по заданным вероятностям ее базисных переменных.

Ключевые слова *— лингвистическая переменная, сложная логическая функция, логико-вероятностный метод.*

Введение

Многообразие решений в задачах управления динамическими объектами связано с операциями над логическими переменами (ЛП) и логическими функциями (ЛФ). Формальная логика, после того как определены ЛП, позволяет путем логического вывода выражать одни ЛФ через другие ЛФ или ЛП. Формализм преобразования ЛФ достаточно хорошо отработан, позволяет эквивалентно представлять их в различных логических базисах, решать задачи логического вывода, интерпретации решений и другие, не прибегая к понятиям атрибута ЛФ. Можно сказать, что формальная логика является самодостаточной.

Понятие лингвистической переменной и его применение к принятию приближенных решений, введенное Л. Заде в 1976 г. [1], практически не повлияло на развитие логики. На наш взгляд, формальная логика осталась неизменной. В большинстве работ по нечеткой логике и ее приложениям изменился лишь порядок операций над ЛП и их атрибутами, в результате чего интерпретация задач существенно усложнилась.

Введением концепции ЛФ нами сделана попытка упорядочить понятия, относящиеся к принятию решений при управлении динамическими объектами, обозначить проблемы, возникающие при учете различных ограничений, и показать возможные пути их решения.

Комбинаторный метод вычисления вероятностей сложных ЛФ

С введением понятия атрибута ЛФ, например вероятности, возникают проблемы, связанные с определением математических операций с атрибутами, при проведении операций над ЛФ. Отсюда появляются определения: «логико-вероятностная задача», «нечеткая логика» и др. Для отделения логических задач от задач, связанных с операциями над атрибутами ЛФ, нами принята следующая концепция:

— логические переменные являются атомами логической системы;

— если с ЛП связывается атрибут, ЛП называется логико-лингвистической переменной;

— атрибутом может быть любой упорядоченный набор данных;

— логические операции имеют приоритет перед операциями над атрибутами;

— символьное выражение, описывающее ЛФ, является алгоритмом над атрибутами ЛП.

Благодаря арифметическим свойствам ЛФ, которые они проявляют при их представлении в виде алгебраических структур по mod2, оказывается возможным сведение логических задач к «арифметическим» или подобным арифметическим. Это в общем случае позволяет представлять логические системы как линейные структуры, уравнения которых не содержат конъюнктивных элементов.

Тогда для таких систем логический вывод представляется как процедура обращения **[0, 1]**-матриц, а принятие решения — как многократное повторение этой процедуры при изменении начальных условий.

Линеаризация систем уравнений логического типа, содержащих конъюнкции из компонентов вектора состояний, позволяет за счет его расширения упорядочить причинно-следственные связи в комбинаторных задачах математического программирования и сравнительно просто определить их сложность, а также оценить логическую замкнутость и непротиворечивость исходной нелинейной системы логических уравнений.

При большом числе ЛП число логических слагаемых k в выражении для ЛФ может быть велико, так как k может достигать значений $2^n - 1$, где n — число ЛП. Запись таких функций в символьном виде в ЭВМ требует больших затрат памяти. Еще раз отметим, что для сравнительно богатых содержанием логических задач, требующих больших размерностей ЛФ, вряд ли является целесообразным использовать представление этих функций и их систем в ЭВМ в символьном виде. Практическую реализацию символьных преобразований в ЭВМ можно осуществлять на основе вычислительной комбинаторики. В частности, если условиться, что элементы ЛФ расположены в некотором лексикографическом порядке, то нет необходимости хранить ее в памяти ЭВМ в символьном виде, а для однозначной идентификации любого члена логической суммы достаточно задать число элементов множества ЛП и порядковый номер этого члена. Тогда номера индексов конъюнктивных элементов каждого члена логической суммы могут быть определены комбинаторными методами. Оперирование с упорядоченными множествами позволяет заменить хранение ЛФ, представленных в символьном виде, на хранение сравнительно простых программ, обеспечивающих быстрое вычисление индексов переменных и атрибутов для их использования в последующих численных операциях, в том числе и для вычисления вероятностей.

Характеристики логико-вероятностных систем с упорядоченными элементами

Для формального описания алгоритмов вычисления вероятности ЛФ с упорядоченными элементами введем следующие определения.

Базисный вектор вероятностей логической системы — упорядоченное множество вероятностей элементов базисного вектора **x**: $\mathbf{P}_x^T = <p_{x1}, p_{x2}, \ldots, p_{xn}>$.

Фундаментальный вектор вероятности логической системы — упорядоченное множество элементов декартова произведения базисного вектора $\mathbf{P}_x$, дополненного 1 на месте последнего элемента: $\mathbf{P}_s^T = <p_{x1}, p_{x2}, \ldots, p_{xn}, p_{x1}p_{x2}, p_{x1}p_{x3}, \ldots, p_{x1}p_{x2}p_{x3}, \ldots, p_{x1}p_{x2}, \ldots, p_{xn}, 1>$.

Фундаментальный вектор вероятности ЛФ — упорядоченное множество вероятностей элементов вектора $\mathbf{S}_f$, дополненного 1 на месте последнего элемента:

$\mathbf{P}_{Sf} = <p_{f1}, p_{f2}, \ldots, p_{fn}, p_{f1f2}, p_{f1f3}, \ldots, p_{f1f2f3}, \ldots, p_{f1f2 \cdots fm}, \ldots, p_{f1f2 \cdots fk}, 1>$.

Комбинаторные операции над ЛФ с упорядоченными элементами

При определении фундаментального вектора **S** как упорядоченного множества нами формально не был определен закон упорядочения. В частности, порядок следования элементов s_i, которые можно рассматривать как конъюнкции из компонентов базисного вектора, может подчиняться закону, согласно которому:

— индексы компонентов x_i базисного вектора **x**, входящих в компоненты s_i фундаментального вектора **S**, представляются в виде кортежей (векторов) $\mathbf{A}_i$ целых чисел, располагающихся в порядке возрастания;

— для каждого s_i соответствующий ему кортеж индексов $\mathbf{A}_i$ имеет значение их произведения не меньше, чем значение произведения индексов, входящих в кортеж $\mathbf{A}_{i-1}$, и не больше, чем в кортеж $\mathbf{A}_{i+1}$;

— ни одно сочетание входящих в кортежи индексов базисного вектора не повторяется. Таким образом, вектору **S** ставится в соответствие таблица **A**, строками которой являются кортежи $\mathbf{A}_i$ целых чисел, обозначающих индексы компонентов базисного вектора ЛП.

Алгоритм упорядочения элементов фундаментального вектора **S** в соответствии с этим законом в нотации языка Паскаль можно представить в следующем виде:

```
begin
for i:=1 to k do A[i]:=i; {Первое подмножество}
p:=k;
while p>=1 do
begin
write (A[1],...,A[k]); {Напечатать}
if A[k]=n then p:= p-1 else p:=k;
if p>=1 then {Цикл с уменьшением на 1}
for i:=k downto p then A[i]:=A[p]+i-p+1;
end
end
```

В основе алгоритма лежит комбинаторная процедура генерирования всех k-элементных подмножеств множества $\{1, \ldots, n\}$, названная [2] лексикографическим упорядочением.

Все элементы вектора **S** после лексикографического упорядочения могут быть разбиты на

m	A				m	A			
1	1	2	3	4	9	1	3	5	6
2	1	2	3	5	10	1	4	5	6
3	1	2	3	6	11	2	3	4	5
4	1	2	4	5	12	2	3	4	6
5	1	2	4	6	13	2	3	5	6
6	1	2	5	6	14	2	4	5	6
7	1	3	4	5	15	3	4	5	6
8	1	3	4	6					

группы. Номер группы определяется по признаку количества элементов в каждом слагаемом. Число групп в **S** равно n. Число элементов n_i в группе с номером i равно $n_i = C_n^i = \dfrac{n!}{i!(n-i)!}$.

Пример 1. Пример генерирования последовательности k-элементных подмножеств множества $\{1, \ldots, n\}$ при помощи рассмотренной выше процедуры, где $n = 6$, т. е. результаты упорядочения элементов фундаментального вектора, представлен в таблице.

Рассмотрим нахождение индексов **S** в 4-й группе для числа переменных базисного вектора, равного шести, т. е. $n = 6$, $i = 4$. Пусть требуется определить набор индексов 5-го ($m = 5$) элемента 4-й группы ($i = 4$). Как видно из таблицы, для $m = 5$ имеем **A** = <1 2 4 6>. Число членов этой группы $n_4 = C_6^4 = \dfrac{6!}{4!(6-4)!} = 15$. Другой задачей является определение порядкового номера m по заданному упорядоченному набору индексов. Например, для заданного набора индексов < 1 3 5 6 > ЛП, составляющих один из элементов фундаментального вектора **S**, из таблицы определяем, что этому набору соответствует порядковый номер $m = 9$. Поэтому, как будет показано далее, отказ от символьной формы представления в ЭВМ ЛФ, замена ее формой индексных векторов и применение комбинаторных методов их обработки имеет принципиальное значение, так как при большом числе ЛП и слагаемых ЛФ позволяет существенно экономить память ЭВМ и сокращать время вычислений.

Алгоритм вычисления вероятности сложной ЛФ

Пусть задана произвольная ЛФ в полиномиальной нормальной форме (ПНФ), аргументы которой логически совместны: $f_{1,\,2,\,\ldots,\,k} = f_1 \oplus f_2 \oplus \ldots \oplus f_m \oplus \ldots \oplus f_k$. Требуется вычислить вероятность этой функции при заданном базисном векторе вероятностей логической системы $\mathbf{P}_x$.

Поскольку составляющими f_i являются другие ЛФ f_m и глубина вложенности ЛФ не ограничена, то без приведения их к каноническому виду невозможно определить логическую зависимость одной ЛФ от другой. Поэтому приведение ЛФ к каноническому виду является основным условием при вычислении ее вероятности. При этом любая ЛФ [3] может быть представлена в форме полинома Жегалкина (канонической ПНФ) единственным образом — посредством компонентов вектора **S** в качестве ее аргументов.

Несложно показать, что значение вероятности логической суммы двух совместных ЛФ в алгебре Жегалкина $f_{1,\,2} = f_1 \oplus f_2$ определяется выражением

$$p_{f1,2} = p_{f1} + p_{f2} - 2p_{f1f2};$$

для трех ЛФ —

$$p_{f1,2,3} = p_{f1} + p_{f2} + p_{f3} - 2(p_{f1f2} + p_{f1f3} + p_{f2f3}) + 4p_{f1f2f3};$$

для четырех ЛФ —

$$p_{f1,\,2,\,3,\,4} = p_{f1} + p_{f2} + p_{f3} + p_{f4} - 2(p_{f1f2} + p_{f1f3} + p_{f1f4} + p_{f2f3} + p_{f2f4} + p_{f3f4}) + 4(p_{f1f2f3} + p_{f1f2f4} + p_{f1f3f4} + p_{f2f3f4}) - 8p_{f1f2f3f4}$$

и т. д., где вероятности p_{f1}, p_{f2}, p_{f3}, p_{f4}, p_{f1f2}, p_{f1f3}, p_{f1f4}, p_{f2f3}, p_{f2f4}, p_{f1f2f3}, p_{f1f2f4}, p_{f1f3f4}, p_{f2f3f4}, $p_{f1f2f3f4}$ являются элементами вектора $\mathbf{P}_{Sf}$.

В общем случае вероятность произвольной ЛФ, приведенной к каноническому виду, можно представить как произведение вектора-строки **R** на $\mathbf{P}_{Sf}$:

$$p_{f1,\,2,\,\ldots,\,k} = \mathbf{R}\mathbf{P}_{Sf},$$

где **R** — вектор-строка, содержащая $k + 1$ группу упорядоченных элементов r_i; i — обозначает номер группы; k — число логических слагаемых в исходной ЛФ.

Все элементы в i-й группе равны, а их число n_i определяется числом сочетаний из k по i: $n_i = C_k^i$.

Значения элементов r_i для каждой i-й группы, кроме последней $(k + 1)$-й, определяются выражением $r_i = (-1)^{r_{k+1}}(-2)^{i-1}$, $(k + 1)$-я группа содержит всего один элемент.

Значение нормирующего коэффициента r_{k+1} для единственного элемента $(k + 1)$-й группы равно 0 или 1. При $r_{k+1} = 0$ вычисляется вероятность ЛФ, а при $r_{k+1} = 1$ — вероятность отрицания ЛФ.

При представлении ЛФ в дизъюнктивной нормальной форме (ДНФ), т. е. в виде

$$f_{1,\,2,\,\ldots,\,k} \Leftrightarrow f_1 \vee f_2 \vee \ldots \vee f_m \vee \ldots \vee f_k \Leftrightarrow fd_{1,\,2,\,\ldots,\,k},$$

где символ $\vee$ — дизъюнкция, выражение для вероятности этой ЛФ совпадает с общеизвестным вы-

ражением для вероятности совместных событий [4], которая может быть вычислена по формуле

$$p_{fd1,\,2,\,\ldots,\,k} = \mathbf{RdP}_{Sf},$$

а элементы rd_i вектора **Rd** — по формуле $rd_i = (-1)^{(i-1)+r_{k+1}}$.

При представлении любой ЛФ как произведения идентификационной строки на фундаментальный вектор вероятность этой функции можно рассматривать как алгебраическую сумму, каждый элемент которой может вычисляться независимо от других компонентов только по его порядковому номеру.

Пример 2. Логическая функция f_i имеет шесть слагаемых $k = 6$, тогда число компонентов вектора $\boldsymbol{P}_{Sf}$ будет равно $n_{sf} = 2^k = 64$. Требуется вычислить вероятность 50-го компонента ($m = 50$). Напомним, что число элементов в i-й группе $n_i = C_k^i$.

Порядковые номера элементов вектора $\mathbf{P}_{Sf}$, которые являются последними в каждой i-й группе, могут быть вычислены по формуле $j_i = \sum_{m=1}^{m=i} C_k^m$.

Вектор-строка **J** из этих элементов для нашего примера имеет вид **J** = < 6 21 41 56 62 63 64 >. При заданном $m = 50$, $j_3 = 41$, $j_4 = 56$, т. е. $41 < m < 56$, определяем, что элемент находится в 4-й группе. Порядковый номер элемента в группе будет $m - j_3 = 50 - 41 = 9$. В соответствии с таблицей 9-й элемент 4-й группы есть $\mathbf{A}_9$ = < 1 3 5 6 >. Он идентифицирует номера индексов слагаемых ЛФ, которые должны быть логически перемножены. После логического умножения компонентов слагаемых ЛФ в соответствии с $\mathbf{A}_9$ будет получен компонент, являющийся конъюнкцией компонента базисного вектора.

Вычисление значения вероятности этого компонента осуществляется путем арифметического умножения вероятностей, составляющих ее ЛП.

Таким образом, вероятность ЛФ определяется как сумма 2^k элементов. Эти элементы могут быть разбиты на k групп. Число элементов в каждой группе определяется числом сочетаний из k элементов по i, где i — номер группы. Элементы, входящие в группы с нечетными номерами, имеют положительный знак, а с четными — отрицательный. Вычисление значения каждого элемента осуществляется алгоритмически, так как затруднительно указать формулу, по которой можно было бы вычислить вероятность путем подстановки исходных данных, а алгоритм можно описать и реализовать в ЭВМ в виде вычислительной процедуры.

Пример 3. Рассмотрим пример вычисления вероятности ЛФ f, представленной в форме ДНФ:

$$f \Leftrightarrow x_1x_3 \vee x_1x_4x_5 \vee x_2x_4 \vee x_2x_3x_5.$$

Исходные данные для расчета вероятности ЛФ имеют следующие значения:

— размерность базисного вектора $n = 5$;

— число логических слагаемых в ЛФ $k = 4$;

— базисный вектор вероятностей ЛФ $\mathbf{P}_f$ = [0.6 0.7 0.9 0.6 0.7].

В соответствии с введенными выше определениями: $f \Leftrightarrow fd_{1,\,2,\,\ldots,\,4}$, а $p = p_{fd1,\,2,\ldots,\,k} = \mathbf{RdP}_{Sf}$.

Вычисления вероятности ЛФ для этого примера дают следующие результаты:

1) число слагаемых функции вероятности $n_p = 2^k = 16$;

2) **Rd** = [1 1 1 1 –1 –1 –1 –1 –1 –1 –1 1 1 1 1 –1 0];

3) $\mathbf{P}_{Sf}$ = [0.54 0.25 0.42 0.44 0.23 0.23 0.27 0.18 0.16 0.27 0.16 0.16 0.16 0.16 0.16 0]^T;

4) вероятность ЛФ $p = 0.81$.

Приближенные методы вычисления вероятности сложной ЛФ по заданным вероятностям ее базисных переменных

При значениях $k > 50$ процедура вычисления на ЭВМ вероятности ЛФ по формуле $p_{f1,\,2,\,\ldots,\,k} = \mathbf{RP}_{Sf}$ требует больших затрат машинного времени. Однако, как показали численные расчеты, не все слагаемые $p_{f1,\,2,\ldots,\,k}$ дают соизмеримый вклад. Оказывается, что при определенных условиях частью слагаемых при вычислении $p_{f1,\,2,\,\ldots,\,k}$ можно пренебречь из-за их малого значения, при этом объем вычислений резко сокращается. По существу, задача состоит в определении числа первых номеров групп, которые определяют основной вклад в значение вероятности p, а влиянием остальных можно пренебречь.

Для оценки вклада отдельных групп слагаемых в выражении для вероятности ЛФ в ее результирующее значение допустим, что:

— логическая функция имеет k логических слагаемых и представлена в форме, в которой каждое логическое слагаемое является конъюнкцией из компонентов базисного вектора **x**;

— все логические слагаемые независимы и совместны;

— вероятности всех логических слагаемых равны M.

Для иллюстрации процесса вычисления p сформируем вспомогательный вектор $\mathbf{SPS} = \mathbf{D}_r\mathbf{P}_{Sf}$, где $\mathbf{D}_r$ — диагональная матрица, диагональными элементами которой являются элементы вектора-строки **Rd**. Тогда процесс вычисления вероятности ЛФ p можно представить как суммирование элементов вектора **SPS**:

$$p(i) = \sum_{j=1}^{j=i} sps(j),$$

где i — номер слагаемого функции вероятности ЛФ. При таких допущениях для каждой i-й груп-

пы можно вычислить составляющую вероятности ЛФ, которая определяет вклад в p всех элементов i-й группы.

Значение $sps(i)$ при задании ЛФ в ПНФ определяется по формуле

$$sps(i) = (-2)^{i+1}(M)^i C_k^i,$$

где C_k^i — число сочетаний из k по i.

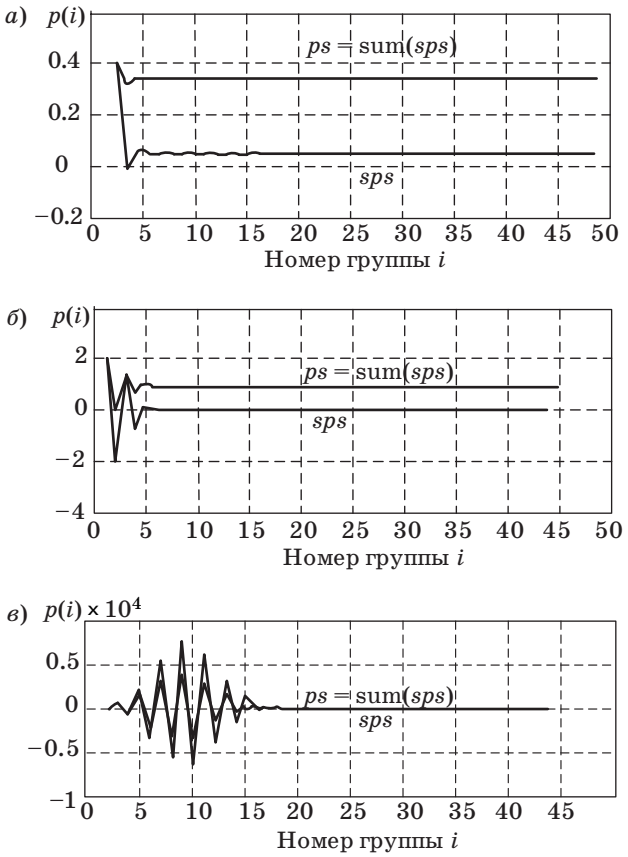
При задании ЛФ в ДНФ $sps(i)$ определяется по формуле

$$sps(i) = (-1)^{i+1}(M)^i C_k^i.$$

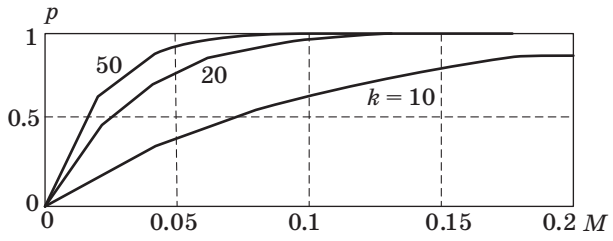
Тогда вероятность ЛФ p можно представить как функцию:

$$p(i) = \sum_{j=1}^{j=i} sps(j).$$

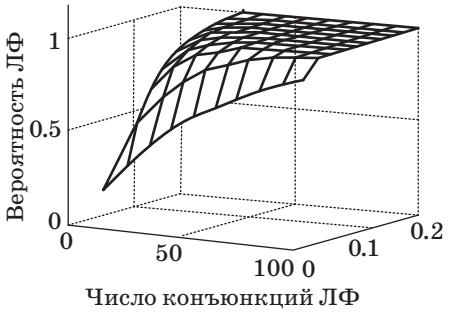
В соответствии с принятой моделью была составлена программа и проведены расчеты влияния M на то, какие группы слагаемых нужно учитывать, а какими можно пренебречь. На рис. 1–3 показаны результаты этих расчетов. На рис. 1, *а*—*в* изображено по две кривые линейно-



■ ***Рис. 1.*** *Графики приближенного вычисления вероятности ЛФ: а — $p(k) = 0.338$; $M = 0.01$; $i \le 4$; б — $p(k) = 0.8779134$; $M = 0.05$; $i \le 8$; в — $p(k) = 1$; $M = 0.3$; $i \le 25$*



■ ***Рис. 2.*** *Графики функции вероятности ЛФ $p = f(M)$ при фиксированных значениях числа логических слагаемых $k = 10, 20, \ldots, 50$*



■ ***Рис. 3.*** *Трехмерный график функции вероятности ЛФ от двух аргументов: числа логических слагаемых k и их вероятности M*

интерполированных функций: $sps(i)$ и $p(i)$. Вероятность ЛФ равна p при $k = n = 41$, число групп i, необходимых для точного расчета вероятности ЛФ, не превышает определенного значения. Из результатов расчетов видно влияние на поведение $p(i)$ заданного значения M слагаемых ЛФ. При малых значениях M ($0.01 < M < 0.05$) $p(i)$ сходится к вероятности p ЛФ при числе групп $i < 7$, т. е. из 41 группы следует, что для расчета вероятности ЛФ нужно учитывать только 7 первых групп. При значениях $0.05 < M < 0.15$ (см. рис. 3) $p(k, M)$ быстро сходится к 1 при числе слагаемых ЛФ $k < 30$. Анализ графиков (см. рис. 1–3) позволяет сделать следующий вывод: при числе слагаемых ЛФ больше 50 и их среднеарифметической вероятности $M > 0.15$ нет необходимости точно вычислять вероятность ЛФ, так как она практически равна 1, с другой стороны, чем меньше среднеарифметическое значение вероятности слагаемых ЛФ, тем меньшее число слагаемых в выражении для ее вероятности нужно учитывать при вычислениях (см. рис. 2).

Нами рассмотрен пример, в котором ЛФ представлена суммой других ЛФ, которые совместны и независимы. При несовместных ЛФ вычисление вероятности p ЛФ проводится для слагаемых только 1-й группы, при этом сумма вероятностей слагаемых не должна превышать 1 (по определению несовместных ЛП и ЛФ [5]).

Если слагаемые ЛФ совместны и логически зависимы, то при вычислении ее вероятности в об-

щем случае нельзя дать оценку числу групп слагаемых, которые необходимо учитывать при расчетах.

Однако оценку числа удерживаемых при расчете вероятности ЛФ групп слагаемых, при принятых выше допущениях, можно считать первым приближением. Если приближенный расчет вероятности ЛФ не удовлетворяет по точности, то число слагаемых можно итерационно увеличивать до тех пор, пока значение вероятности при последующих итерациях не будет удовлетворять заданной точности. Полученные в результате численных приближенных расчетов оценки точности вычислений вероятности ЛФ в зависимости от учета только первых групп слагаемых и отбрасывания остальных имеют важное принципиальное значение, так как позволяют при точном расчете p существенно уменьшить число слагаемых при суммировании. При большом числе базисных переменных такой двухэтапный подход к вычислению p может оказаться единственно возможным. Следует заметить, что именно лексикографическое упорядочение фундаментального вектора ЛФ и, соответственно, слагаемых в выражении для ее вероятности дает возможность при проведении вычислений контролировать вычислительный процесс и принимать решение о его прекращении при достижении требуемой точности.

Заключение

Таким образом, прежде чем приступать к точным расчетам вероятности ЛФ, содержащей большое число слагаемых, необходимо предварительно оценить погрешность, вносимую составляющими, входящими в соответствующую группу слагаемых, определяющих эту вероятность. Это позволит существенно уменьшить объем вычислений путем «отсечения хвостов», дающих малый вклад в значение вероятности ЛФ. Кроме того, описанный подход к вычислению вероятности сложной ЛФ является одним из возможных. Для сравнения его с другими подходами и оценки эффективности предложенных алгоритмов должны быть приняты общие критерии сложности. Одним из таких критериев может служить критерий сложности, в соответствии с которым мера сложности вычислительной процедуры определяется числом шагов алгоритма. Поскольку число слагаемых в выражении для вероятности ЛФ возрастает по экспоненте от числа составляющих ее логических слагаемых, то вряд ли можно ожидать, что будет найден алгоритм, принципиально уменьшающий экспоненциальную сложность вычислительных процедур. В случае приведения ЛФ к ортогональному виду (совершенной ДНФ) число ее слагаемых также имеет экспоненциальную зависимость от исходной размерности [6]. Поэтому вычисление вероятности ЛФ «в лоб», без предварительных приближенных оценок числа «удерживаемых» членов, приведет к неоправданно большим затратам машинного времени или памяти ЭВМ.

Нами были рассмотрены возможности вычисления вероятности сложных ЛФ аналитическими методами, однако существует подход, когда эта вероятность может быть вычислена методом прямого статистического эксперимента, путем генерирования значений базисных ЛП и последующей статистической обработки. Однако обоснование корректности постановки статистического эксперимента представляет самостоятельную задачу, которая по сложности может оказаться не проще рассматриваемой.

Литература

1. **Заде Л. А.** Понятие лингвистической переменной и его применение к принятию приближенных решений. — М.: Мир, 1976. — 165 с.
2. **Липский В.** Комбинаторика для программистов: Пер. с польск. — М.: Мир, 1988. — 213 с.
3. **Толковый** словарь по вычислительным системам / Под общ. ред. В. Иллингуорта. — М.: Машиностроение, 1990. — 560 с.
4. **Вентцель Е. С.** Теория вероятностей. — М.: Наука, 1969. — 425 с.
5. **Кук С. А.** Сложность процедур вывода теорем // Кибернетический сборник. Новая серия. М.: Мир, 1975. Вып. 12. С. 5–15.
6. **Рябинин И. А.** Надежность и безопасность сложных систем. — СПб.: Политехника, 2000. — 248 с.

УДК 519.81

НЕЧЕТКИЕ ДЕРЕВЬЯ РЕШЕНИЙ
(НЕЧЕТКИЕ ПОЗИЦИОННЫЕ ИГРЫ)

В. Г. Чернов,
доктор экон. наук, профессор
Владимирский государственный университет

Рассматривается решение задачи альтернативного выбора на основе нечетких деревьев решений (нечетких позиционных игр), особенностью которых является использование нечетких качественных оценок последовательности решений и состояний природы.

Ключевые слова *— нечеткое множество, функция принадлежности, нечеткое дерево решений.*

Введение

Деревья решений — один из часто используемых методов выбора наилучшего направления действий на множестве имеющихся вариантов.

Многие задачи альтернативного выбора требуют анализа последовательности решений и состояний среды, когда одна совокупность стратегий и состояний порождает другое состояние подобного типа. Если имеют место два (или более) последовательных множества решений, причем последующие решения основываются на результатах предыдущих, и/или два (или более) множеств состояний среды (т. е. появляется целая цепочка решений, вытекающих одно из другого, которые соответствуют событиям, происходящим с некоторой вероятностью), то такие ситуации могут быть представлены формальными моделями в виде позиционных или многоэтапных игр. Графическое представление такой игры называется деревом решений.

Строя дерево решений, лицо, принимающее решение, определяет в соответствии со своими представлениями последовательность решений и состояний среды с указанием предполагаемых вероятностей и выигрышей (проигрышей) для любых комбинаций альтернатив и состояний среды. Таким образом, можно утверждать, что концепция ожидаемого значения является неотъемлемой частью метода деревьев решений. Согласно классификации, предложенной в работах [1–3], рассматриваемая модель выбора относится к индивидуальным, многоэтапным, многокритериальным.

Традиционно при использовании деревьев решений в задачах альтернативного выбора применяются точечные оценки вероятностей состояний природы, выигрышей или проигрышей, т. е. это означает, что по сути предполагаемые, ожидаемые значения в явном виде в процессе решения задачи не представлены.

Кроме того, в традиционных вариантах использования деревьев решений отсутствует возможность применять качественные оценки параметров задачи.

Деревья решений с оценками в виде нечетких чисел

Отражение концепции ожидаемого значения может быть обеспечено, если от точечных оценок перейти к оценкам в виде нечетких чисел, а также использовать качественные оценки в форме лингвистических высказываний, формализуемых соответствующими нечеткими множествами. При использовании нечетких чисел для оценки состояний среды принципиальных изменений в процессе прохождения дерева решений не потребуется. Все вычисления будут выполняться в базисе так называемых «мягких вычислений», а конечный результат будет представлен в форме нечетких чисел.

Моделирование уровня неопределенности в данном случае может быть реализовано путем расширения базового множества соответствующих оценок, а также выбора вида функции принадлежности.

При использовании нечетких чисел в конечной оценке наряду с числовым результатом будет получено и распределение его истинности в виде соответствующей функции принадлежности. При-

чем характер функции (степень размытости) принадлежности будет характеризовать и степень нечеткости решения. Все это будет дополнительной информацией для лица, принимающего решение.

Если нечеткое дерево решений содержит только количественные оценки состояний природы и альтернатив в виде соответствующих нечетких чисел, то для нахождения наилучшего решения используются традиционные методы расчетов, реализованные в базисе «мягких вычислений».

Результат получается в виде набора нечетких чисел с соответствующими функциями принадлежности

$$M = \{\mu_f(z) : f = \overline{1, F}\}, \qquad (1)$$

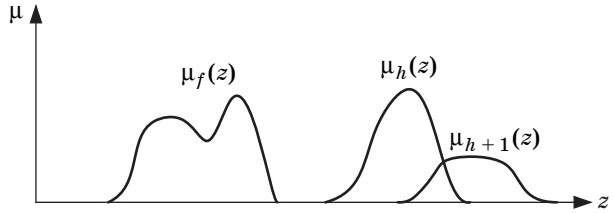
где $\mu_f(z)$ — функция принадлежности нечеткого числа, представляющего интегральную оценку ветви дерева с номером f; F — общее количество результатов в виде нечетких чисел, получаемых в результате обработки дерева решений, равное числу ветвей; $z \in R$, R — множество вещественных чисел.

Полученные для каждой ветви нечеткого дерева решений интегральные оценки будут иметь форму нечетких чисел с некоторыми функциями принадлежности $\mu_f(z)$, $\mu_h(z)$, $\mu_{h+1}(z)$, приведенными на рис. 1.

При количественных оценках интегральные заключения, представленные нечеткими числами (1), располагаются в естественном порядке на оси вещественных чисел. Поэтому если по условиям задачи нас интересует максимизация эффекта, то оценка наилучшего решения должна находиться в области больших значений на числовой оси. Если же решается обратная задача, например минимизации риска, то оценка наилучшего решения должна находиться в области меньших значений на числовой оси.

Поскольку функции принадлежности, представляющие оценки по соответствующим ветвям дерева, могут иметь произвольный характер, выбор наилучшей альтернативы следует выполнять по координате центра тяжести

$$CG_f = \frac{\sum\limits_i \mu_f(z_i) z_i}{\sum\limits_i \mu_f(z_i)}. \qquad (2)$$



■ ***Рис. 1.*** *Возможные оценки ветвей дерева решений: f, h, h + 1 — номера ветвей дерева решений*

Согласно оценке (2), в качестве наилучшей альтернативы следовало бы рассматривать альтернативу, соответствующую ветви дерева с номером $h + 1$. Однако это решение не может рассматриваться как вполне обоснованное. Дело в том, что функция принадлежности может интерпретироваться как функция распределения оценок истинности принимаемого решения. Нетрудно видеть (см. рис. 1), что оценка истинности решения по ветви дерева с номером h будет выше, чем у ветви с номером $h + 1$. Для разрешения этой ситуации можно использовать мультипликативную оценку

$$R_f = CG_f \mu_f(CG_f), \qquad (3)$$

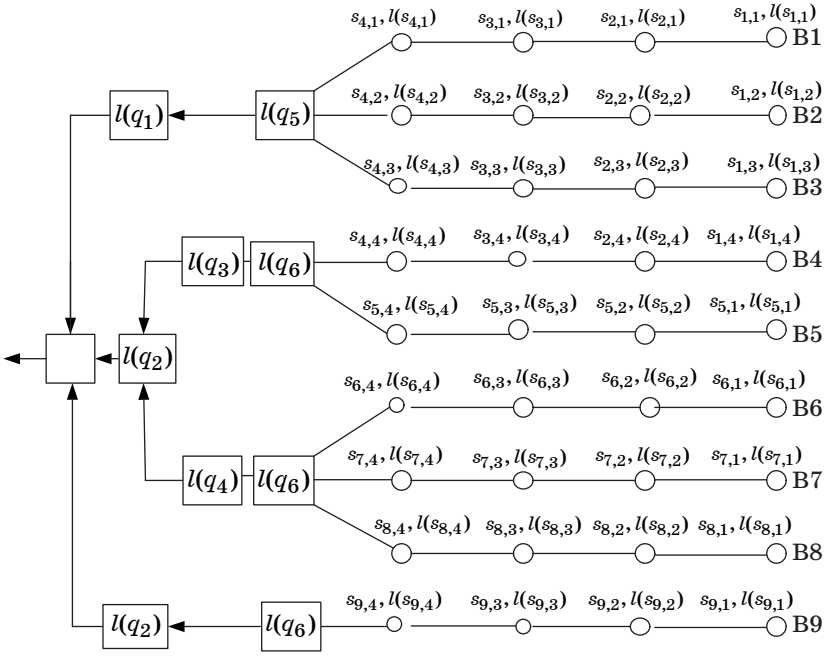
т. е. произведение значения координаты центра тяжести CG_f и значения истинности в этой точке. Вполне возможно, что для ситуации, представленной на рис. 1, $R_h > R_{h+1}$, и тогда решение, соответствующее ветви с номером h, можно рассматривать в качестве более предпочтительного. Кроме того, можно использовать еще и оценку надежности принимаемого решения [4], которая учитывает степень расплывчатости нечетких значений.

Деревья решений с качественными оценками

В практике принятия решений могут возникнуть ситуации, когда некоторые или все состояния природы и оценки альтернатив могут иметь только качественные представления. Поскольку нечеткие числа — это по сути нечеткие множества, элементы которых определены на множестве вещественных чисел, то обработка нечеткого дерева решений при наличии количественных и качественных оценок может осуществляться на основе операций над нечеткими множествами без привлечения арифметических операций. В случае, когда в дереве решений присутствуют как количественные, так и качественные оценки, все они должны быть приведены к единому универсальному множеству $U = [0, 1]$, что выполняется простыми формальными преобразованиями.

Деревья решений могут использоваться как метод многокритериального выбора наилучшей альтернативы в различных приложениях: в технических — выбор на множестве альтернативных проектов реконструкции некоторого объекта или системы, выбор варианта действий при ликвидации последствий чрезвычайных ситуаций техногенного или природного характера; в экономических — выбор, например, инвестиционного решения.

Дальнейшее рассмотрение проведем на примере дерева решений, представленного на рис. 2,



■ ***Рис. 2.*** *Пример дерева решений: $s_{i,f}$ — состояние природы; $l(s_{i,f})$ — качественная, нечеткая оценка состояния природы; q_j — варианты альтернативных решений; $l(q_j)$ — качественная, нечеткая оценка альтернативного решения*

в котором для обеспечения общности состояния природы, соответствующие конкретным ветвям дерева, обозначены как $s_{i,f} \in S = \{s_{i,f} : i = \overline{1, I}, f = \overline{1, F}\}$, а множество оценок альтернатив — как $Q = \{q_j : j = \overline{1, J}\}$.

Для каждого состояния природы имеется множество лингвистических оценок

$$L(s_{i,f}) = \{l_k(s_{i,f}) : k = \overline{1, K}, \; i = \overline{1, I}, \; f = \overline{1, F}\}$$

и соответствующие нечеткие множества, определяемые функциями принадлежности

$$\mu_{k,i}(z),\ k \in [1, K],\ i \in [1, I],$$

где $z \in U = [0, 1]$ — формальная переменная.

Множество оценок альтернатив также представляется нечеткими множествами

$$L(q_j) = \{l_p(q_j) : p = \overline{1, P}\};$$
$$\mu_{p,j}(y),\ p \in [1, P],\ j \in [1, J],$$

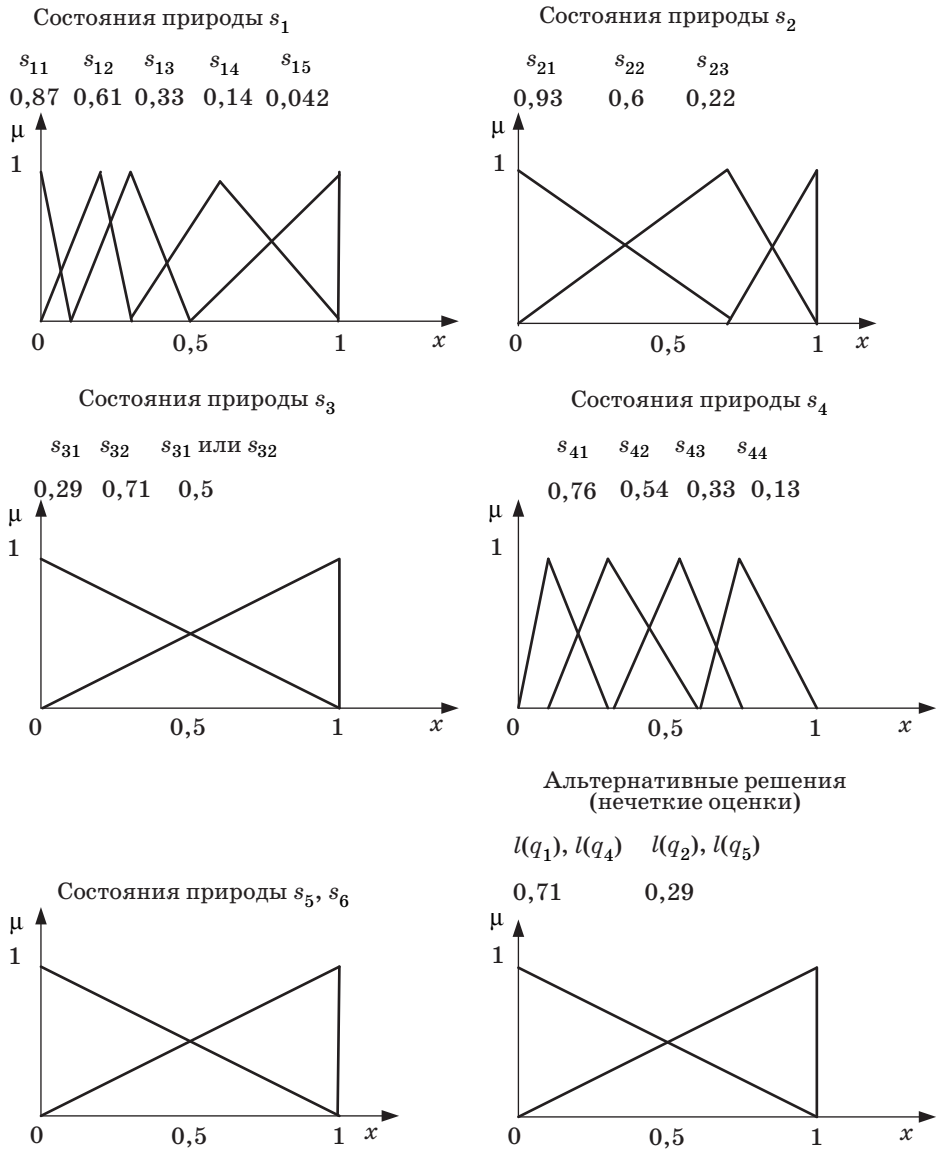
где $y \in U = [0, 1]$ — формальная переменная.

Примеры возможных нечетких оценок состояний природы и альтернативных решений приведены на рис. 3. Треугольные функции принадлежности выбраны только из соображений простоты. При решении конкретной задачи лицо, принимающее решение, может по своему усмотрению определять число состояний природы, альтернативных решений, соответственно выбирать вид функций принадлежности и задавать их параметры. Отметим, что для отдельных ветвей дерева решений значения состояний природы могут совпадать.

Задача заключается в нахождении интегрального решения наилучшего в смысле некоторого, заранее выбранного критерия. Например в принятии решения относительно определенной суммы инвестиций при ожидаемых результатах и состояниях природы (величине ожидаемого спроса, размера предприятия, в которое предполагаются инвестиции). Для конкретного вида решаемой задачи наименованиям состояний природы и альтернативных решений, а также используемым нечетким оценкам придается предметная ориентация $l(q_6)$.

Пусть некоторая ветвь нечеткого дерева решений $\tau_f \in T$, где T — множество ветвей, $f = \overline{[1, F]}$, содержит N_f состояний природы и H_f оценок альтернатив.

Согласно утверждениям Р. Беллмана, Л. Заде [5, 6], задача достижения нечетко поставленной цели при нечетком ограничении решается на основе принципа слияния. При этом нечеткое решение определяется как нечеткое подмножество множества, получающегося в результате слияния нечетких целей и нечетких ограничений. Тогда нечеткое множество, представляющее оценку альтернативного решения, соответствующее ветви с номером f, может определяться как пересечение [3]:



■ ***Рис. 3.*** *Примеры нечетких оценок* (*числовые значения — координаты центров тяжести фигур, представленных соответствующими функциями принадлежности*)

$$M_f = \bigcap_{\substack{\text{по всем } j\in[1,\,N_f]\\ i\in[1,\,H_f]}} (\mu_{p,\,j},\ \mu_{k,\,i}), \qquad (4)$$

где M_f — функция принадлежности, соответствующая нечеткому множеству, представляющему интегральную оценку по f-й ветви дерева, поскольку в данном случае отсутствует компенсация между большими и малыми степенями принадлежности (оценками по различным критериям) [3].

Для выбора наилучшего решения необходимо сравнить нечеткие множества, полученные по соотношению (4). Если рассматривать функции принадлежности (4) как совокупность материальных точек, массы которых равны значениям функции принадлежности, то обобщенной характеристикой такой системы является координата центра тяжести, вычисляемая по соотношению (2). Тогда значение функции принадлежности, соответствующее координате центра тяжести, может быть использовано в качестве критерия оценки найденного решения, так как саму функцию принадлежности можно интерпретировать как функцию распределения истинности для найденного решения.

Рассмотренный вариант нахождения наилучшего альтернативного решения с помощью нечеткого дерева решений позволяет полностью отразить концепцию ожидаемого значения. Однако вполне возможно, что при каких-то оценках состояний природы и альтернатив, особенно когда используются только качественные оценки,

для каких-то ветвей дерева нечеткое множество, получаемое по соотношению (4), будет пустым: $M_f = \varnothing$. Такая возможность следует из отсутствия компенсации между большими и малыми степенями принадлежности (оценками по различным критериям). При этом подобная ситуация не обязательно будет иметь место только для заведомо плохих альтернатив. Кроме того, это может получиться и для большого числа альтернатив из множества рассматриваемых, что, естественно, будет снижать уровень обоснованности принимаемого решения.

В табл. 1 и на рис. 4 представлен пример подобной ситуации для нечеткого дерева решений (см. рис. 2) и нечетких оценок (см. рис. 3), где альтернативные решения, представленные ветвями *4, 5, 8*, составляющие 30 % от общего количества альтернатив, получили ненулевые оценки. При этом предварительный анализ не указывает на то, что остальные решения заведомо плохие.

Последний столбец таблицы представляет мультипликативную оценку, полученную по формуле (2).

Преодолеть проблему пустых пересечений можно следующим образом [7]. Множество сегментов каждой ветви дерева, за исключением последнего, представляющего решение, разбиваются на подмножества, дающие непустые пересечения. В результате для каждой ветви будет получен некоторый набор непустых нечетких множеств с соответствующими функциями принадлежности

$$M_f = \{\mu_{f1}, \mu_{f2}, ..., \mu_{fr}\},$$

где r — число непустых подмножеств, которое может быть получено на множестве сегментов ветви дерева с номером f; $\mu_{fi} \neq 0$, но $\bigcap_i \mu_{fi} = \varnothing$.

Для получения решения используем операцию над нечеткими множествами, предложенную в работе [7] и названную «геометрическая проекция нечетких множеств». Следует отметить, что название оказалось не совсем удачным, так как возникали аналогии с известной операцией «проекция нечетких множеств», что приводило к различным недоразумениям. Поэтому в дальнейшем для этой операции будем использовать наименование «тень нечеткого множества» и обозначать ее «*Sh*» (shadow — тень). Определим эту операцию следующим образом.

Тень нечеткого множества $\tilde{A}$ на нечеткое множество $\tilde{B}$ должна удовлетворять следующим условиям:

1) $Sh(\tilde{A}, \tilde{B})$ — нечеткое множество;

2) $Sh(\tilde{A}, \tilde{A}) = \tilde{A}$;

3) $Sh(\tilde{A}, \tilde{B}) = \varnothing$, если хотя бы одно из множеств $\tilde{A}$ или $\tilde{B}$ пустое или множества $\tilde{A}$ и $\tilde{B}$ ортогональны.

Процедуру построения тени нечеткого множества $\tilde{A}$ на нечеткое множество $\tilde{B}$ определим следующим образом (рис. 5):

$$Sh_\varphi(\tilde{A}, \tilde{B}) = \{\varphi[\mu_{\tilde{A}}(y), \mu_{\tilde{B}}(x)] / [y, x' = f(y)]\},$$

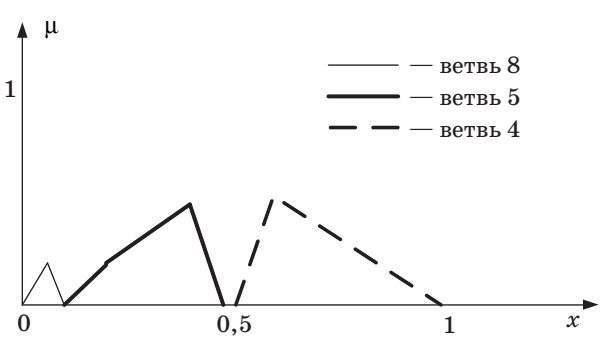
где $f(y) = \dfrac{CG[\mu_{\tilde{B}}(x)]}{CG[\mu_{\tilde{A}}(y)]} y$ — проекционная функция;

$CG[\mu_{\tilde{B}}(x)], CG[\mu_{\tilde{A}}(y)]$ — координаты центров тяжести фигур, ограниченных функциями принадлежности $\mu_{\tilde{B}}(x)$, $\mu_{\tilde{A}}(y)$; φ — функционал, задающий вид преобразований над функциями принадлежности.

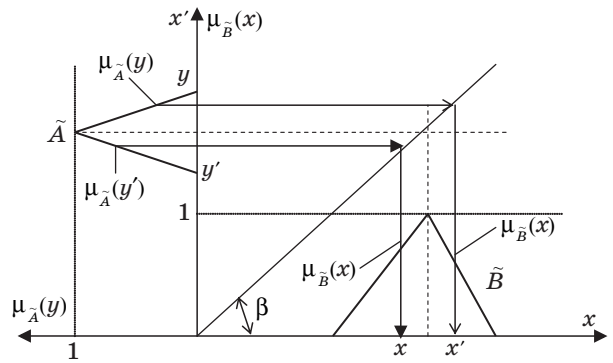
Смысл этой операции состоит в том, что в зависимости от взаимного расположения нечетких множеств и, соответственно, угла наклона проек-

■ ***Таблица 1***. *Ранжировка решений (операция пересечения)*

Ветвь 1	Ветвь 2	Ветвь 3	Ветвь 4	Ветвь 5	Ветвь 6	Ветвь 7	Ветвь 8	Ветвь 9
0	0	0	0,73	0,36	0	0	0,12	0
0	0	0	0,27	0,48	0	0	0,11	0
0	0	0	0,2	0,17	0	0	0,014	0



■ ***Рис. 4***. *Результат решения при использовании операции пересечения нечетких оценок*



■ ***Рис. 5***. *Геометрическое представление операции «тень нечеткого множества»*

ционной прямой изменяется «тень» одного нечеткого множества, накладываемая на другое нечеткое множество. Этим будет представляться степень взаимодействия оценок понятий, представляемых нечеткими множествами.

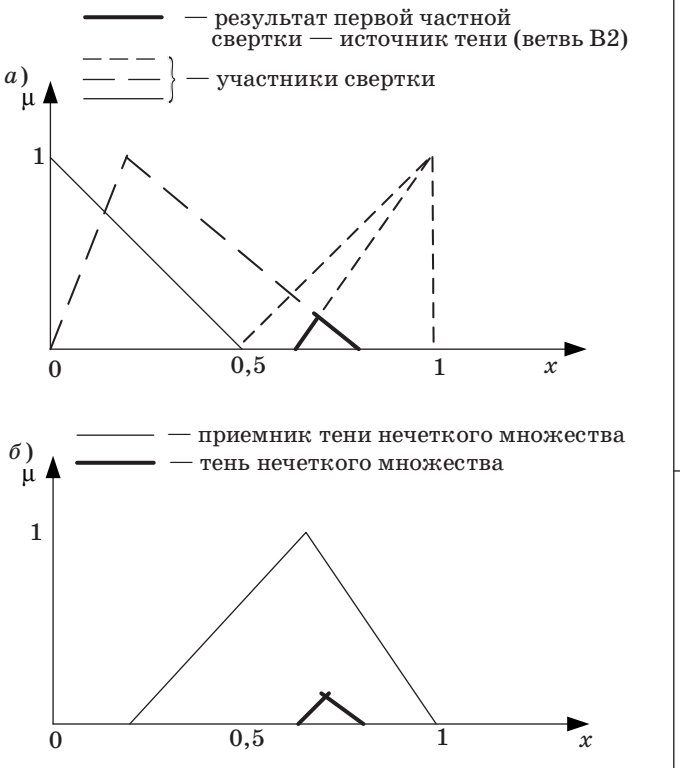
Нечеткое множество $\tilde{A}$, которое проецируется на другое нечеткое множество ($\tilde{B}$), назовем источником тени. Нечеткое множество $\tilde{B}$, на которое проецируется тень нечеткого множества $\tilde{A}$, назовем приемником тени.

Дальнейшие преобразования выполняются в следующей последовательности. Для каждого нечеткого множества, представляемого функцией принадлежности μ_{f_i}, строится тень на нечеткое множество, представляющее искомое решение. В результате для каждой μ_{f_i} получим соответствующую тень Sh_{f_i}, которая по определению будет нечетким множеством с соответствующей функцией принадлежности $\mu_{Sh_{f_i}}$.

Интегральное решение для ветви с номером f определим как пересечение

$$M_{Sh_f} = \bigcap_i \mu_{Sh_{f_i}}.$$

Аналогичные преобразования выполняются и для других ветвей дерева. Если рассматривать $\mu_{Sh_{f_i}}$ как функцию распределения истинности, наилучшее решение определим как $\max\limits_f M_{Sh_f}$.

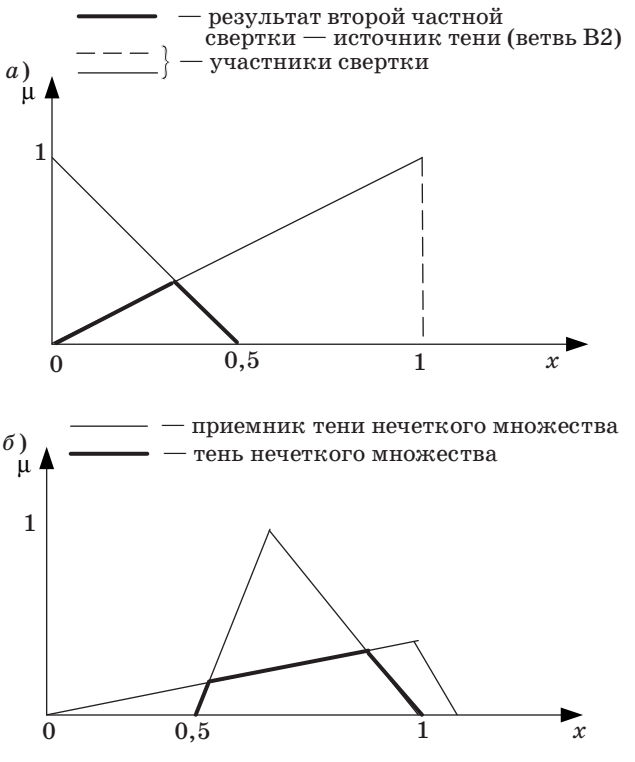


■ ***Рис. 6.*** *Построение первой частной свертки (а) и ее тени (б) для ветви B2 дерева решения*

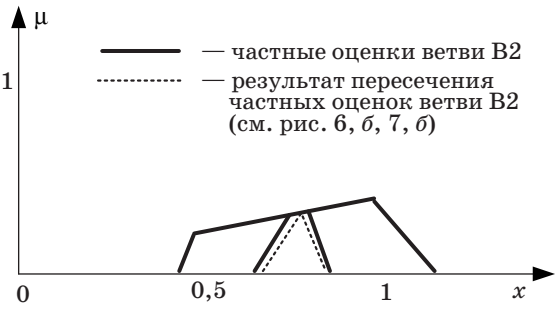
Если однозначное определение максимума не получается, то определяется значение, соответствующее координате центра тяжести CG функции принадлежности M_{Sh_f} и затем находится максимальное значение

$$\max_f M_{Sh_f}(CG).$$

Результаты выполнения описанных выше операций для ветви B2 дерева (см. рис. 2) получены с помощью электронной таблицы Fuzzy Calc [8]. Результаты вычисления частных сверток (рис. 6, *а*, 7, *а*) и их теней (рис. 6, *б*, 7, *б*) можно интерпретировать следующим образом. Частные свертки — это результат слияния нечетких множеств, представляющих частные состояния природы и дающих ненулевые пересечения. Тень этих нечетких оценок



■ ***Рис. 7.*** *Построение второй частной свертки (а) и ее тени (б) для ветви B2 дерева решения*



■ ***Рис. 8.*** *Итоговая оценка*

■ ***Таблица 2.*** *Ранжировка решений (используется «тень» нечетких сигналов)*

Номер ветви дерева	Оценка истинности решения	Номер ветви дерева	Оценка истинности решения
B1	0,559	B9	0,155
B5	0,476	B2	0,132
B4	0,267	B7	0,132
B6	0,187	B8	0,123
B3	0,164		

на нечетком множестве, представляющем альтернативное решение, будет определять степень их соответствия этому решению. Затем нужно будет получить интегральную оценку путем слияния частных сверток с помощью операции пересечения.

Результат вычисления пересечений функций принадлежности, представляющих нечеткие множества частных оценок для фрагментов ветви B2 (см. рис. 6, *б*, 7, *б*), показан на рис. 8.

Остальные ветви обрабатываются аналогичным образом. В табл. 2 приведены результаты обработки по рассмотренной методике дерева решений (см. рис. 2) и нечетких оценок (см. рис. 3).

Заключение

Таким образом, предложенный в работе подход позволяет:

1) отразить при решении задачи альтернативного выбора решения на основе нечеткого дерева концепцию ожидаемого значения в оценках решений и состояний природы;

2) использовать как числовые, так и качественные представления оценок решений и состояний природы в форме лингвистических высказываний, отражающих субъективные предпочтения лица, принимающего решения;

3) решать задачу выбора альтернативных технических и экономических проектов в условиях нестатической неопределенности.

Литература

1. **Козелецкий Ю.** Психологическая теория решений. — М.: Прогресс, 1979. — 504 с.
2. **Kickert W. J. M.** Fuzzy theories on decision-making. — Leiden: Martinus Nijhoff, 1978. — 182 p.
3. **Нечеткие** множества в моделях управления и искусственного интеллекта / Под. ред. Д. А. Поспелова. — М.: Наука, 1986. — 312 с.
4. **Чернов В. Г.** Модели поддержки принятия решений в инвестиционной деятельности на основе аппарата нечетких множеств. — М.: Горячая линия-Телеком, 2007. — 312 с.
5. **Bellman R., Zadeh L. A.** Decision-making in a fuzzy environment // Management Science. 1979. Vol. 17. P. 141–162.
6. **Вопросы** анализа и процедуры принятия решений / Сост. В. С. Шилейко. — М.: Мир, 1976. — 230 с.
7. **Чернов В. Г.** Решение задач многокритериального альтернативного выбора на основе геометрической проекции нечетких множеств // Информационно-управляющие системы. 2007. № 1 (26). С. 46–52.
8. **Чернов В. Г. и др.** Решение бизнес-задач средствами нечеткой алгебры. Кн. 2. Электронная таблица Fuzzy Calc. — М.: Тора-Центр, 1998. — 70 с.

УДК 517.977.56, 519.876.5

АДАПТИВНОЕ УПРАВЛЕНИЕ ДОРОЖНЫМ ДВИЖЕНИЕМ НА БАЗЕ СИСТЕМЫ МИКРОСКОПИЧЕСКОГО МОДЕЛИРОВАНИЯ ТРАНСПОРТНЫХ ПОТОКОВ

А. С. Голубков,
инженер, младший научный сотрудник
В. А. Царев,
канд. техн. наук, доцент
Институт менеджмента и информационных технологий
Череповецкий филиал Санкт-Петербургского государственного политехнического университета

Описаны состав и особенности функционирования современных автоматизированных систем управления дорожным движением. Предложен способ адаптивного управления дорожным движением на основе предсказания транспортных потоков и быстрых моделей оптимизации перекрестков. Представлены характеристики системы микроскопического моделирования транспортных потоков, применяемой в системе адаптивного управления дорожным движением.

Ключевые слова *— адаптивное управление дорожным движением, оптимизация управления дорожным движением, моделирование транспортных потоков, микроскопическое моделирование.*

Введение

В настоящее время во многих крупных городах весьма остро стоит проблема транспортных заторов. При этом исследования [1] показывают, что потенциал существующих улично-дорожных сетей (УДС) используется далеко не полностью. Повышение пропускной способности УДС может быть достигнуто за счет внедрения автоматизированных систем управления дорожным движением (АСУДД). При внедрении АСУДД достигается улучшение следующих показателей [2, 3]: время в пути транспортных средств (ТС) снижается на 10–15 %; количество общих транспортных остановок сокращается на 20–40 %; расход топлива снижается на 5–15 %, количество вредных выбросов в атмосферу сокращается на 5–15 %; повышается безопасность дорожного движения.

Современные АСУДД

Основными компонентами современных АСУДД [4] помимо светофоров и светофорных контроллеров являются:

1) детекторы транспорта (ДТ), обеспечивающие обнаружение ТС и подсчет их числа при движении по полосам;

2) одна или несколько ЭВМ для обработки данных с ДТ и расчета оптимальных управляющих сигналов;

3) совокупность программных средств, реализующих алгоритмы детектирования транспорта и оптимизации управления транспортными потоками;

4) средства информирования водителей ТС (различные информационные табло);

5) средства связи и телекоммуникации, используемые для объединения программно-аппаратных средств АСУДД в единую систему.

В современных АСУДД применяются различные типы детекторов транспорта: петлевые (индукционные); инфракрасные активные и пассивные; магнитные; акустические; радарные; видеодетекторы; комбинированные (в различных комбинациях ультразвуковые, радарные, инфракрасные и видеодетекторы). Все ДТ обладают различной эффективностью в различных условиях эксплуатации [5]. Однако в связи с достигнутым высоким уровнем развития вычислительной и телевизионной техники во многих случаях наиболее предпочтительными являются видеодетекторы на основе технологий обработки и анализа изображений, а также комбинации видеодетекторов с детекторами других типов.

В существующих АСУДД тех или иных производителей используются в различных комбинациях три основных способа адаптивного управления транспортными потоками [2].

1. Метод управления с использованием библиотек, характеризуемый предварительным расчетом множества планов координации и переключением их на основании текущих усредненных показаний стратегических ДТ путем выбора из библиотеки соответствующего подходящего плана.

2. Метод актуального управления, характеризуемый предварительным расчетом планов координации светофоров, переключением их по календарному графику и реализацией изменений в этих планах в соответствии с транспортными запросами, фиксируемыми локальными детекторами на отдельных направлениях.

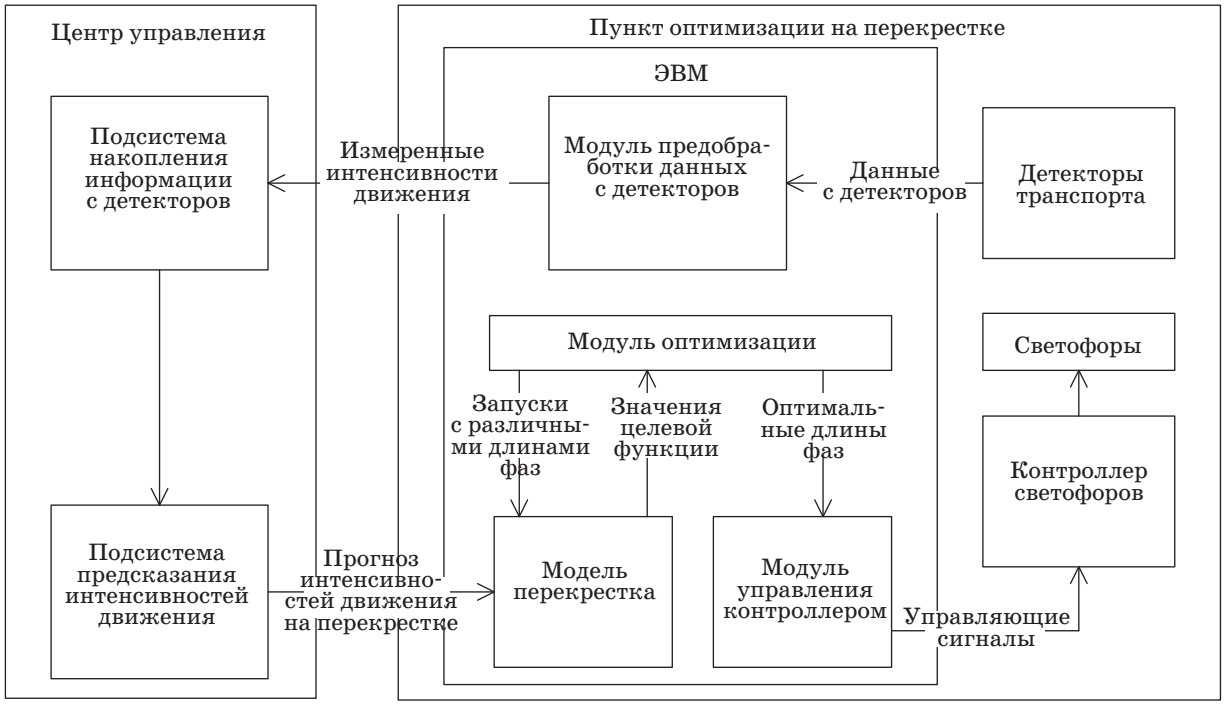
3. Метод адаптивного управления, характеризуемый постоянным пересчетом планов координации и календарных режимов на основании информации, получаемой с локальных и стратегических (путевых) детекторов в режиме реального времени.

Оптимизация управления транспортными потоками в современных АСУДД производится различными методами. В системе **Balance (Германия)** [6] применяются генетические алгоритмы оптимизации. В системе Utopia (Нидерланды) [7] производится расчет на основе ценовой функции, учитывающей время задержки, число остановок, специфические приоритетные требования, взаимное расположение перекрестков. В системе «Спектр» (Санкт-Петербург, Россия) [8] используются следующие алгоритмы: поиск разрывов потока транспорта; расчет по формуле Вебстера; переключение программ по интенсивностям. В АСУДД производства ОАО «Электромеханика» (Пенза, Россия) [9] используется следующее алгоритмическое обеспечение: алгоритм поиска разрыва потоков транспорта; поиск разрыва с сохранением общей длительности цикла координации; алгоритм переключения заранее рассчитанных режимов по контрольным точкам интенсивности движения транспорта; алгоритм динамического перерасчета параметров цикла на основе формулы Вебстера. В АСУДД «Агат» (Минск, Белоруссия) [3] используются следующие эвристические алгоритмы управления: выбор плана координации по карте времени; выбор фазы, режима по плану координации; выбор плана координации по параметрам движения в характерных точках и др.

Авторами данной статьи предлагается алгоритм адаптивного управления транспортными потоками на основе краткосрочного предсказания интенсивностей транспортных потоков и моделей оптимизации перекрестков.

Адаптивное управление транспортными потоками на основе моделей оптимизации перекрестков

Разрабатываемая система управления дорожным движением (рисунок) состоит из одного центрального пункта и множества локальных пун-



■ *Схема системы адаптивного управления дорожным движением*

ктов управления, число которых соответствует числу управляемых перекрестков в системе. Все локальные пункты имеют соединение по каналам связи с центральным пунктом управления.

Центральный пункт управления выполняет функции сбора и обработки информации об интенсивностях движения транспортных средств в УДС. Обработка информации представляет собой предсказание величин транспортных потоков на основе следующих данных:

— текущих интенсивностей транспортных потоков;

— скоростей движения ТС;

— расстояний между смежными управляемыми перекрестками в системе;

— предсказания маршрутов движения ТС на основе статистики для текущего дня недели и времени суток;

— текущих длин фаз светофорных объектов на перекрестках УДС.

Локальные пункты в системе выполняют непосредственно оптимизацию управления транспортными потоками на соответствующих перекрестках. В состав каждого локального пункта управления входят:

— детекторы транспорта;

— ЭВМ, выполняющая предобработку данных с ДТ, если это необходимо, и оптимизацию управления транспортными потоками;

— контроллер светофоров, допускающий внешнее задание длин фаз светофорного объекта;

— светофоры.

В качестве ДТ предлагается использовать видеодетекторы. В этом случае сигнал с видеокамер поступает в ЭВМ локального пункта управления, где программный модуль предобработки выполняет анализ видеоизображений и оценку интенсивностей транспортных потоков на всех контролируемых полосах. Далее интенсивности транспортных потоков передаются в центральный пункт управления.

Оптимизация управления транспортными потоками производится следующим образом. В ЭВМ имеется точная программная микроскопическая модель перекрестка. При расчете оптимальных длин фаз для следующего фазового цикла управления светофорным объектом (длительность фазового цикла составляет, как правило, 2–5 мин) выполняются следующие действия.

• В модели задаются входные интенсивности транспортных потоков на следующие 5 мин (прогноз интенсивностей от центрального пункта управления) с точностью до отдельного ТС.

• Модуль оптимизации запускает прогоны модели перекрестка длительностью 5 мин модельного времени, для каждого прогона задает новые длины фаз модельного светофорного объекта и рассчитывает по результатам каждого прогона значение целевой функции.

• В результате цикла оптимизации, состоящего из нескольких прогонов модели, модуль оптимизации находит оптимальные длины фаз модельного светофорного объекта, соответствующие экстремуму целевой функции поиска.

Длины фаз светофорного объекта представляют собой вектор параметров оптимизации $\boldsymbol{\varphi} = (\varphi_1, \varphi_2, \varphi_3, \varphi_4)$ (на крестообразном перекрестке обычно задается не более четырех фаз). В качестве целевой функции $F(\boldsymbol{\varphi})$ может служить среднее время ожидания проезда перекрестка ТС. Критерием оптимизации в этом случае будет минимум среднего времени ожидания проезда

$$\min_{\boldsymbol{\varphi}\in\Phi} F(\boldsymbol{\varphi}) = F(\boldsymbol{\varphi}^*),$$

где Ф — допустимое множество значений координат вектора длин фаз; $\boldsymbol{\varphi}^*$ — вектор оптимальных значений длин фаз. Допустимое множество значений координат вектора длин фаз имеет следующий вид:

$$\Phi = \{\boldsymbol{\varphi} \,|\, T_{\min} \le \boldsymbol{\varphi}_i \le T_{\max}, i = 1, \ldots, 4\} \subset R^4,$$

где $T_{\min}$ и $T_{\max}$ — соответственно минимальное и максимальное значения длины фазы.

Расчет производных целевой функции на модели является невозможным, поэтому в качестве методов оптимизации могут быть использованы только прямые методы. Предложено применение поочередного циклического варьирования длин фаз светофорного объекта от прогона к прогону с постоянным шагом по длине фазы. Длина шага варьирования длин фаз может быть задана равной 2–3 с.

Необходимым условием возможности реализации описанной системы адаптивного управления дорожным движением является наличие системы микроскопического моделирования транспортных потоков, скорость работы которой была бы достаточной для выполнения оптимизации длин фаз светофорного объекта за время одного фазового цикла.

Система микроскопического моделирования транспортных потоков

Авторами статьи разработана система микроскопического моделирования транспортных потоков в УДС, которая может быть использована для оптимизации управления транспортными потоками в составе системы адаптивного управления дорожным движением. Главной особенностью системы моделирования является применение дискретно-событийного подхода в моделиро-

вании [10, 11], благодаря чему система имеет высокое быстродействие.

Быстродействие системы оценено в серии экспериментов с моделями отдельных типовых перекрестков. Эксперименты выполнены на компьютере с процессором Intel Core 2 Quad Q6600 с частотой каждого ядра 2,4 ГГц (реально в экспериментах использовалось только одно ядро, так как моделирование выполняется в один программный поток). В результате моделирование транспортных потоков через единичный перекресток в течение 45 сут (3 888 000 с) заняло 2864 с процессорного времени. Таким образом, превышение скорости моделирования над скоростью течения реального времени составило 3 888 000/2864 ≈ ≈ 1358 раз, т. е. за время реального фазового цикла на перекрестке модуль оптимизации способен выполнить более 1300 прогонов оптимизационного эксперимента.

Особенностью дискретно-событийного подхода в моделировании является независимость результатов моделирования от скорости выполнения модели, т. е. даже в режиме полной загрузки процессора моделирование покажет совершенно идентичные результаты результатам выполнения, например, в режиме реального времени.

Напротив, в системно-динамическом подходе при ускорении моделирования посредством увеличения шага дискретизации по времени точность моделирования падает. Системно-динамический подход [12] реализует подавляющее большинство современных систем микроскопического моделирования транспортных потоков: Aimsun (Испания) [13], Paramics Modeler (Шотландия) [14], DRACULA (Великобритания) [15], TransModeler (США) [16], VISSIM (Германия) [17]. Во всех перечисленных системах моделирования используется шаг дискретизации по времени 0,1–1,0 с.

В системно-динамической дорожно-транспортной модели шаг моделирования по времени, равный 1 с, вполне способен лишить модель адекватности. Так, ТС на скорости 60 км/ч за 1 с преодолевает более 16 м пути, т. е. на типовых скоростях движения модельное ТС позиционируется лишь с точностью порядка 10 м.

В предложенной дискретно-событийной модели точность позиционирования модельных объектов остается постоянной практически при любой скорости и зависит от разрядности используемых переменных и типа выполняемых над ними арифметических операций. При использовании чисел с плавающей запятой двойной точности (64 бита, 15 значащих десятичных цифр мантиссы) точность позиционирования модельных ТС в дискретно-событийной модели в любой момент времени составит не более 1 см.

Заключение

Предложенная система адаптивного управления дорожным движением способна продемонстрировать высокую эффективность благодаря исчерпывающей оптимизации каждого отдельного перекрестка и учету транспортных потоков между соседними перекрестками с точностью до отдельных ТС. При наличии в УДС по какому-либо направлению транспортного потока высокой плотности происходит автоматическая подстройка управления на всех смежных перекрестках с организацией на данном направлении зеленой волны. При этом оптимизации подвергаются и все прочие направления с транспортными потоками меньшей плотности.

Оптимизация управления каждым отдельным перекрестком в реальном времени является возможной благодаря использованию системы микроскопического дискретно-событийного моделирования транспортных потоков в УДС, разработанной авторами статьи. Данная система моделирования вследствие применения дискретно-событийного подхода обладает высокой производительностью и точностью. В ближайшее время на сайте разработчиков [18] будет доступна ознакомительная версия системы моделирования.

Качество оптимизации управления транспортными потоками в высокой степени зависит от точности предсказания плотности потоков транспорта. При этом точность предсказания тем выше, чем меньше временной интервал предсказания. При использовании на локальных перекрестках аппаратного обеспечения достаточной производительности пересчет оптимальных длин фаз цикла регулирования светофорного объекта может производиться с началом каждой следующей фазы. В этом случае реально используемый временной интервал предсказания сократится до длительности одной фазы, т. е. до 15–100 с, в результате чего повысится эффективность оптимизации.

Литература

1. **Бродский Г. С., Айвазов А. Р.** Автоматизированное управление дорожным движением в городской среде // Мир дорог. 2007. № 26. С. 2–3.
2. **Бродский Г. С., Рыкунов В. В.** Поехали! АСУДД — мировой опыт и экономический смысл // Мир дорог. 2008. № 32. С. 36–39.

3. **ГНПО** АГАТ. http://www.agat.by (дата обращения: 16.06.2010).
4. **Crowdhury M. A., Sadek A.** Fundamentals of Intelligent Transportation System planning. — Boston — London: Artech House, 2005. — 190 p.
5. **Кременец Ю. А., Печерский М. П., Афанасьев М. Б.** Технические средства организации дорожного движения. — М.: Академкнига, 2005. — 279 с.
6. **GEVAS** software: Traffic Control. http://www.gevas.eu/index.php?id=149&L=1 (дата обращения: 16.06.2010).
7. **UTOPIA** — Peek Traffic. http://www.peektraffic.nl/page/484 (дата обращения: 16.06.2010).
8. **ЗАО** «РИПАС»: Разработка и производство автоматизированных систем. http://www.ripas.ru (дата обращения: 16.06.2010).
9. **АСУДД** — ОАО «Электромеханика». http://www.elmeh.ru/catalog/3/asud (дата обращения: 16.06.2010).
10. **Карпов Ю. Г.** Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. — СПб.: БХВ-Петербург, 2006. — 400 с.
11. **Советов Б. Я., Яковлев С. А.** Моделирование систем. — М.: Высш. шк., 2001. — 343 с.
12. **Nagel K.** High-speed microsimulations of traffic flow. Thesis: University Cologne, 1995. — 202 p.
13. **Aimsun**. The integrated transport modeling software. http://www.aimsun.com (дата обращения: 20.05.2010).
14. **Quadstone** Paramics. Traffic Simulation Solutions. http://www.paramics-online.com (дата обращения: 20.05.2010).
15. **SATURN** Software Web Site. https://saturnsoftware.co.uk (дата обращения: 20.05.2010).
16. **TransModeler** Traffic Simulation Software. http://www.caliper.com/transmodeler/ (дата обращения: 20.05.2010).
17. **PTV** Vision — транспортное планирование. http://www.ptv-vision.ru (дата обращения: 20.05.2010).
18. **Компания** «Малленом». http://www.mallenom.ru (дата обращения: 20.05.2010).

УВАЖАЕМЫЕ АВТОРЫ!

Каждому из Вас необходимо зарегистрироваться на сайте РУНЭБ (http://www.elibrary.ru) с тем, чтобы Вам присвоили индивидуальный цифровой код (при регистрации код присваивается автоматически), что обязательно для создания корректной базы данных РУНЭБ, объективно отражающей информацию о Вашей научной активности, а также для подсчета Вашего индекса цитирования (РИНЦ).

УДК 004.273

АРХИТЕКТУРА WEB-ОРИЕНТИРОВАННЫХ САПР

Г. Д. Дмитревич,
доктор техн. наук, профессор
А. А. Мохсен,
аспирант
А. И. Ларистов,
канд. техн. наук, доцент
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

Показана архитектура современных Web-приложений, которые представляют собой коллекцию элементов Web-узла, программно выполняющих какие-либо действия и являющихся основой Web-ориентированных САПР. Web-приложения (Web-САПР) создаются таким образом, чтобы они выполнялись на Web-серверах и использовали в качестве пользовательского интерфейса Web-браузеры. Обычно Web-приложения создаются как приложения в архитектуре «клиент—сервер», но серверная часть может иметь различные архитектурные решения. Приведем пример архитектуры конкретной Web-ориентированной САПР.

Ключевые слова *— Web-приложение, Web-ориентированная САПР, архитектура Web-приложения, архитектура Web-ориентированной САПР, архитектура клиент—сервер, образовательные порталы.*

Одним из направлений развития систем автоматизированного проектирования в настоящее время является разработка распределенных архитектур и построение на их основе Web-ориентированных САПР. Реализация подобной распределенной архитектуры САПР требует создания специального Web-приложения, обеспечивающего запуск и синхронизацию подсистем на стороне клиента и на стороне сервера, а также пересылку данных между клиентскими и серверными подсистемами. Рассмотрим основные методы построения Web-приложений.

Изначально World Wide Web (WWW) создавалась как «пространство для обмена информацией, в котором люди и компьютеры могут общаться между собой» [1]. Поэтому первые Web-приложения представляли собой примитивные файловые серверы, которые возвращали статические HTML-страницы запросившим их клиентам. Таким образом, Web начиналась как документо-ориентированная сеть.

Следующим этапом развития Web стало появление понятия приложений, которые базировались на таких интерфейсах, как CGI (или FastCGI), а в дальнейшем — на ISAPI. Common Gateway Interface (CGI) — это стандартный интерфейс с серверами, позволяющий выполнять серверные приложения, вызываемые через URL. Входной информацией для таких приложений служило содержимое HTTP-заголовка (и тело запроса при использовании протокола POST). CGI-приложения генерировали HTML-код, который возвращался браузеру. Основной проблемой CGI-приложений было то, что при каждом клиентском запросе сервер выполнял CGI-программу в реальном времени, загружая ее в отдельное адресное пространство.

Появление Internet Server API (ISAPI) позволило не только решить проблемы производительности, которые возникали с CGI-приложениями, но и предоставить в распоряжение разработчиков более богатый программный интерфейс. ISAPI DLL можно было ассоциировать с расширениями имен файлов через специальную мета-базу. Эти два механизма (CGI и ISAPI) послужили основой создания первого типа Web-приложений, в которых в зависимости от каких-либо клиентских действий выполнялся серверный код. Таким образом, стала возможной динамическая генерация содержимого Web-страниц и наполнение Web перестало быть чисто статическим.

Интерфейс ISAPI — это особенность Microsoft Internet Information Server. ISAPI-приложения представляют собой динамические загружаемые библиотеки (DLL), которые выполняются в адресном пространстве Web-сервера. У других Web-серверов через некоторое время также появилась возможность выполнять приложения, реализо-

ванные в виде библиотек [2]. В случае Web-серверов Netscape этот программный интерфейс назывался NSAPI (Netscape Server API). У довольно популярного Web-сервера Apache также имеется возможность выполнять Web-приложения, реализованные в виде библиотек; такие библиотеки называются Apache DSO (Dynamic Shared Objects).

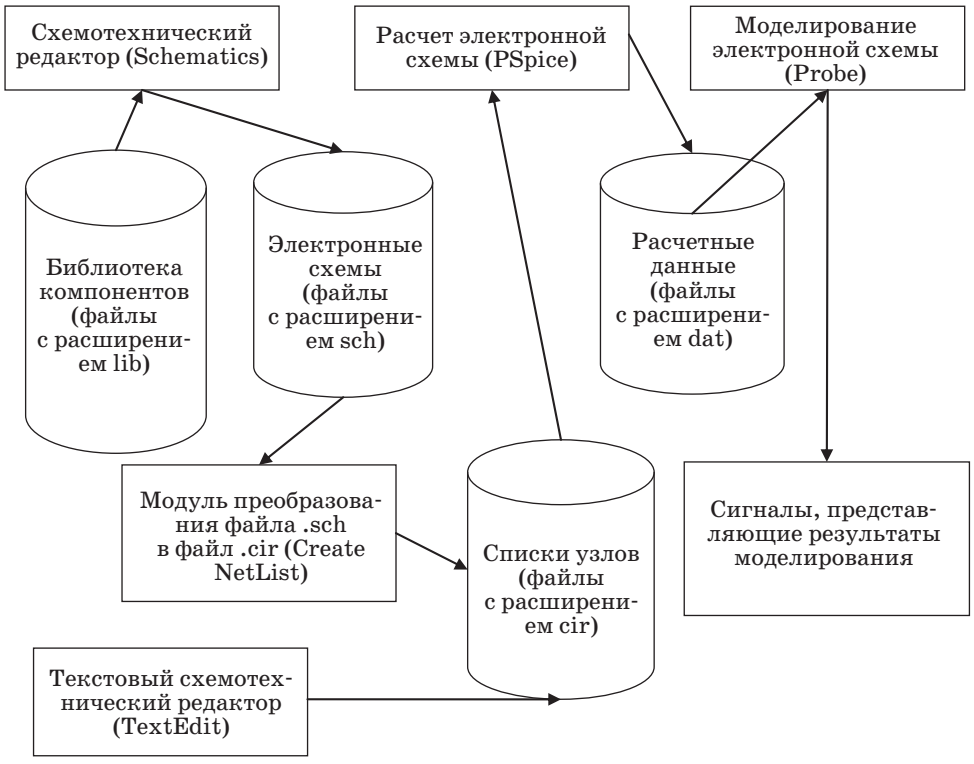
Отметим, что с ростом объема используемых данных и числа посетителей Web-сайтов возрастают и требования к надежности, производительности и масштабируемости Web-приложений. Следующим этапом эволюции подобных приложений стало отделение бизнес-логики, реализованной в Web-приложении, а нередко и сервисов обработки данных и реализации транзакций от его интерфейса. В этом случае в самом Web-приложении обычно остается так называемая презентационная часть, а бизнес-логика, обработка данных и реализация транзакций переносятся в сервер приложений в виде бизнес-объектов.

Следующим шагом эволюции Web-приложений, помимо доступа к корпоративным данным и данным партнеров, стало получение доступа к корпоративным приложениям. Для интеграции Web-приложений с внутренними информационными системами предприятий и приложениями, обеспечивающими взаимодействие с клиентами и партнерами, используются специальные решения, называемые корпоративными порталами. Нередко в содержимое портала включают средства управления информационным наполнением Web-сайта, поскольку объем данных, доступных пользователям с помощью сайтов крупных компаний и порталов, сейчас таков, что управление этими данными «вручную» не представляется возможным [3].

Решение многих описанных выше задач, возникающих при создании современных Web-приложений, теперь начинает возлагаться на Web-сервисы — не зависящие от платформы, объектной модели и клиента программные компоненты, которые можно вызывать из клиентских Web-приложений (а также из самих Web-сервисов) через основанный на протоколе HTTP и языке XML протокол SOAP. Для описания Web-сервисов используется XML-подобный язык WSDL, а для организации реестров Web-сервисов, в которых разработчики и компании могут искать необходимые им сервисы, а также публиковать данные о своих сервисах, — интерфейс UDDI.

Обзор возможных архитектур построения Web-приложений позволяет сделать вывод о реализации распределенных САПР на основе технологии ASP.NET, предложенной фирмой Microsoft в среде Visual Studio. Рассмотрим применение данной технологии при разработке Web-ориентированной САПР электронных схем.

Традиционная архитектура САПР (рис. 1) включает отдельные подсистемы, написанные на



■ ***Рис. 1.*** *Традиционная архитектура САПР*

процедурных или объектно-ориентированных языках программирования, имеющих мощные средства для работы с различными структурами данных, таких как списки, очереди, стеки, деревья, графы и т. д. Обмен данными между подсистемами обеспечивается через общую область памяти, через систему двоичных файлов, посредством специализированных языков описания (текстовые файлы).

На настоящий момент актуальными являются разработка и адаптация специальных Internet-ориентированных версий настольных САПР, позволяющих обеспечить автоматизацию процесса проектирования объектов различной физической природы в масштабах предприятия.

Рассмотрим архитектуру Internet-версии подобной САПР на примере САПР электронных схем Design Lab корпорации MicroSim, получившую наибольшее распространение среди разработчиков радиоэлектронной аппаратуры.

Основу системы Design Lab составляет программа PSpice, которая является наиболее известной модификацией программы схемотехнического моделирования SPICE (Simulation Program with Integrated Circuit Emphasis), разработанной в начале 70-х гг. в Калифорнийском университете г. Беркли. Она оказалась очень удачной, интенсивно развивается и де-факто стала эталонной программой моделирования аналоговых устройств.

В состав САПР Design Lab входят следующие подсистемы:

Schematics — интегрированная управляющая подсистема и графический редактор электронных схем;

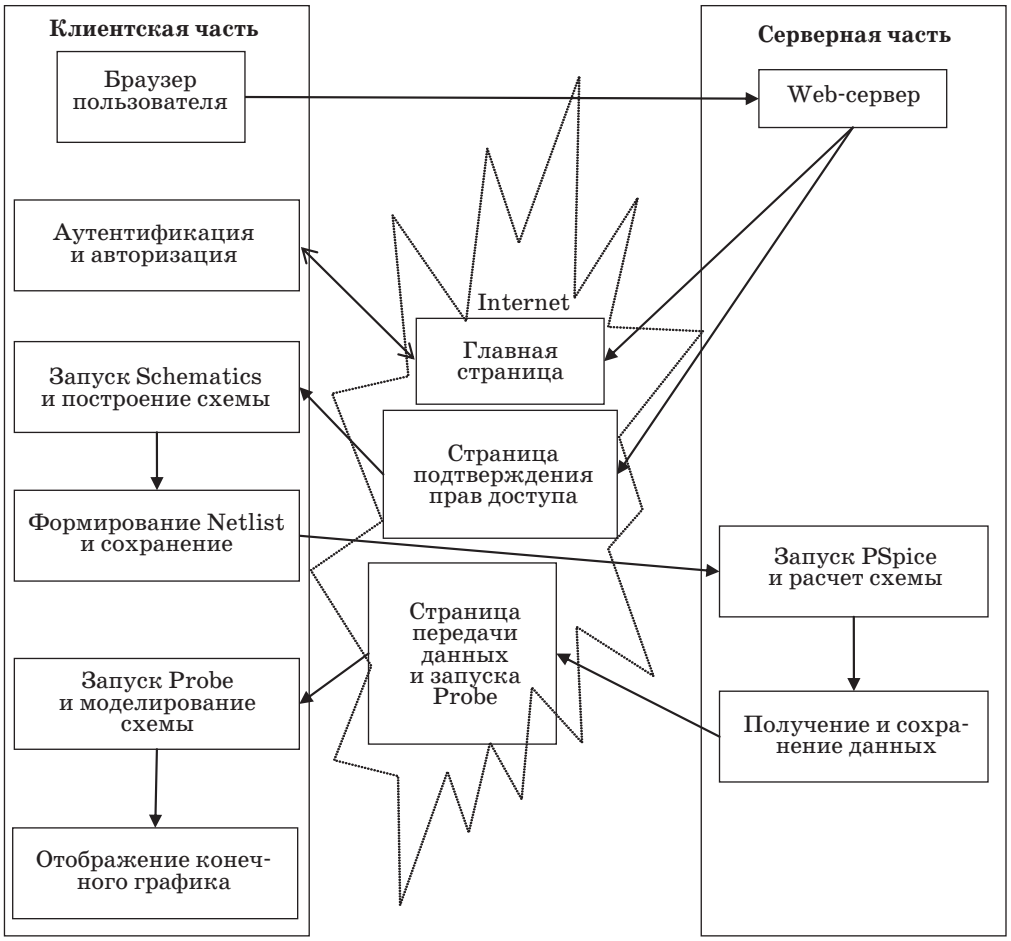
PSpice — проектирующая подсистема, выполняющая построение математической модели и численный анализ схемы;

Probe — графический постпроцессор, ориентированный на отображение в виде графиков результатов моделирования;

StmEd — редактор входных сигналов, позволяющий задавать произвольную форму возмущающих воздействий [4];

Parts — подсистема расчета параметров моделей схемных компонентов.

Для передачи данных между подсистемами Schematics и PSpice используется текстовый файл с описанием схемы на входном языке системы.



■ ***Рис. 2.*** *Архитектура Web-ориентированной САПР электронных схем*

Передача данных моделирования из PSpice в Probe осуществляется с помощью двоичного файла.

В процессе работы САПР активно используются библиотеки параметров схемных компонентов, представляющие собой текстовые файлы на языке задания описания моделей и параметров компонентов схем.

Для адаптации работы системы Design Lab в сети Internet необходима разработка новой распределенной архитектуры САПР с разделением функций между клиентом и сервером, чтобы добиться оптимальной производительности в условиях низкоскоростных каналов Internet и лимитированных ресурсов Web-серверов [5]. Так, предварительную обработку и ввод данных, отправляемых серверу, имеет смысл выполнять на стороне клиента. Это позволит исключить, например, повторные передачи неправильно составленных схем. Графическое представление результатов запроса также стоит выполнять на стороне клиента, что существенно сократит объем данных, передаваемых по сети. Моделирование схемы и доступ к библиотекам параметров схемных компонентов целесообразно обеспечить за счет ресурсов сервера. Отметим, что сеть Intranet — отличная платформа для работы с информацией внутри предприятия. Современный Web-браузер доступен для любой клиентской системы.

Архитектура Web-ориентированной САПР электронных схем представлена на рис. 2.

В настоящее время стали появляться Web-ориентированные САПР. Они представляют собой приложения, разделенные на 3 модуля — клиентский, обслуживающий и проектирующий. Клиентский модуль, выполняемый на компьютере пользователя, осуществляет формирование данных для их передачи на сервер. Обслуживающий модуль через Web-страницы запускает части клиентского модуля. Проектирующий модуль, выполняемый на сервере, производит обработку поступивших данных (см. рис. 2). Intranet ориентирован, как правило, на применение в рамках одного компактного или распределенного предприятия и отличается высокой безопасностью и скоростью работы. Используется для решения задач по автоматизации документооборота, информационному сопровождению бизнес-процессов, поиска и совместного доступа к данным и документам организации и имеет шлюзы для подключения в Internet.

В данной статье мы изучили эволюцию архитектуры Web-решений, начиная от простейших хранилищ HTML-страниц и заканчивая современными корпоративными решениями, интегрированными с корпоративными информационными системами и информационными системами партнеров. Обсудили задачи, возникающие на каждом этапе развития Web-приложений и технологии, их решающие, включая CGI, ISAPI; взаимодействие с серверами приложений и с базами данных; создание и применение Web-сервисов, основанных на XML. Предложили архитектуру Web-ориетированной САПР на основе популярной САПР электронных схем Design Lab корпорации MicroSim.

Литература

1. **Berners-Lee T.** WWW: Past, present, and future // Computer. Oct. 1996. Vol. 29. N 10. P. 69–77.
2. **http:**//supportline.microfocus.com/Documentation/books/nx30books/piisapcn.htm (дата обращения: 10.11.2009).
3. **http:**//zone.ni.com/devzone/cda/epd/p/id/6363 (дата обращения: 02.12.2009).
4. **Разевиг В. Д.** Система схемотехнического моделирования и проектирования печатных плат Design Center (PSpice). — М.: СК Пресс, 1996. — 272 с.
5. **Разевиг В. Д.** Система схемотехнического моделирования Micro-CAP V. — М.: СОЛОН, 1997. — 273 с.

УДК 004.4'244

МЕТОДИКА СИНТЕЗА ТЕСТОВ АППАРАТУРЫ ПО СПЕЦИФИКАЦИЯМ НА ЯЗЫКЕ UML

А. В. Березкин,
магистр техники и технологии, аспирант
А. С. Филиппов,
канд. техн. наук, доцент
Санкт-Петербургский государственный политехнический университет

Язык UML рассматривается как язык описания спецификаций аппаратуры, из которых могут быть получены ее поведенческие тесты. Предлагается использовать данный язык в начале и в середине маршрута проектирования цифровых устройств, когда определяется их структура и функциональность на уровне последовательности управляющих воздействий. Эти спецификации являются документами, по которым создается RTL-описание устройств, а разработанная методика служит для проверки соответствия RTL-описаний UML-спецификациям. Данная проверка осуществляется путем генерации тестов устройств на основании UML-спецификаций.

Ключевые слова *— UML, моделирование аппаратуры, верификация, тестирование.*

Введение

Для задания спецификации на встраиваемые системы удобно использовать языки описания высокого уровня. Наиболее часто используется язык UML (Unified Modeling Language 2.0) [1], специально предназначенный для решения подобных задач и поддержанный инструментальными средствами. Существующие работы (см. далее) представляют методики, позволяющие выполнять формальную верификацию спецификаций на языке UML. Разработанные по данным спецификациям системы тоже верифицируются, однако этот процесс требует введения дополнительных информационных сущностей в процесс разработки. Иными словами, на данный момент нет средств, позволяющих выполнять верификацию систем на основе непосредственно их спецификаций.

Одним из важнейших методов верификации является тестирование — проверка систем в ходе их нормальной работы в рамках заранее подготовленных сценариев, в соответствии с которыми и разрабатывают тесты.

Актуально синтезировать функциональные тесты непосредственно на основе спецификации системы. Целью данной работы является разработка методики синтеза таких тестов. Проблема рассматривается, в основном, для аппаратуры, однако принципы, заложенные в методику, могут быть применены и к программному обеспечению.

Верификация в маршруте проектирования вычислительных систем

Синтез функциональных тестов выполняется исходя из спецификации системы, которая создается в ходе ее проектирования. Следовательно, этот процесс является частью всего маршрута проектирования вычислительных систем. Определим точки маршрута, в которых следует синтезировать и использовать такие тесты. Рассмотрим для этого упрощенную схему маршрута проектирования цифровых систем (рис. 1).

В ходе проектирования системы разработчики создают различные информационные сущности (артефакты), описывающие систему или отдельные ее свойства. На начальных этапах проектирования эти сущности абстрактны (модели, неформальные описания), на более поздних — конкретны (код, образы памяти, используемые компоненты). Создаваемые артефакты анализируются различными методиками верификации. К примеру, техническое задание может подвергаться экспертизе; спецификация системы — проверке моделей; программный код — статическому анализу. В любом случае верификация предполагает проверку соответствия друг другу двух артефактов, даже если один из них не является прямым продуктом процесса разработки конкретной системы (например, в случае статического анализа программный код проверяется

■ ***Рис. 1.*** *Этапы и артефакты маршрута проектирования*

с помощью такого «внешнего» артефакта, как набор обобщенных правил написания корректного кода).

Существенно отметить, что при современных технологиях проектирования цифровых устройств и систем для описания аппаратуры используются языки высокого уровня HDL. Эти формализованные описания представляют собой «аппаратные коды», которые могут быть использованы для автоматизированной компиляции схем аппаратуры. Таким образом, проектирование аппаратного обеспечения систем во многом стало походить на проектирование программного обеспечения.

Тестирование проверяет соответствие разработанной системы (т. е. программного и аппаратного кода) набору требований к работе системы. Если данный набор требований выражен достаточно строго в виде спецификации системы, он может быть использован для генерации тестов.

Предлагаемый подход представлен на рис. 2. На первых этапах проектирования создается набор строгих спецификаций на некотором универсальном языке. Далее на основании этих спецификаций разработчики пишут исходный код компонентов системы. После компиляции системы появляется возможность ее тестирования,

■ ***Рис. 2.*** *Включение автоматизированного тестирования в маршрут проектирования*

и для этого предлагается использовать тесты, сгенерированные из спецификаций, созданных на начальных этапах разработки. Таким образом, предлагаемый подход позволяет установить, насколько разработанная система соответствует требованиям спецификаций, сформулированным ранее. Полнота сгенерированных тестов целиком зависит от того, насколько полная спецификация представлена генератору тестов.

Для того чтобы предложенный подход было удобно применять, язык описания спецификаций системы должен удовлетворять некоторым требованиям. Этот язык должен быть распространенным и доступным, универсальным, способным представлять как абстрактные сущности, так и конкретные. Немаловажна также инструментальная поддержка языка. Всеми перечисленными особенностями обладает UML, что и служит причиной его выбора.

Другая причина выбора языка — огромное количество исследований, посвященных языку и его применениям. В настоящий момент UML рассматривается разработчиками и исследователями не только как язык моделирования программного обеспечения, но и как универсальный язык проектирования любых систем и процессов с применением объектно-ориентированного подхода (см., например, работы [2–5]).

В настоящий момент известно также большое количество программных средств для работы с UML. Обширные списки таких средств можно легко найти на открытых интернет-ресурсах, например на странице [6]. Их анализ показывает, что генерация кода из UML-описаний — хорошо проработанная задача. Как видно, имеется целый ряд средств, которые могут генерировать код на таких языках, как C/C++, C#, Java. Однако весьма затруднительно найти средство, генерирующее аппаратный код из UML-описания. В то же время имеется множество литературы, посвященной методикам генерации аппаратного кода из UML-описания. В качестве примеров можно привести работы [4, 5]. Спецификации, рассматриваемые в данных работах, носят как структурный, так и поведенческий характер. В них предлагается устройства и другие элементы аппаратуры (в том числе их входы и выходы) описывать с помощью классов, а их соединения — с помощью стереотипов и ассоциаций. Предлагается также использовать диаграмму состояний для задания конечных автоматов.

Рассмотренные источники представляют существенный интерес для нашей темы — генерации аппаратных тестов, поскольку тест сам является определенным образом организованным алгоритмом, а при практическом подходе — программным или аппаратным кодом. В то же время генерация теста является отдельной задачей, хотя и связанной с генерацией кода. Например, в работе [7] рассматривается методика генерации простых тестов объектно-ориентированного дизайна программной системы. Но основная масса работ, посвященных генерации тестов из UML-описания, носит теоретический, а не практический характер, как, например, работы [8, 9], в которых описываются алгоритмы генерации тестовых примеров (test case) из диаграмм классов, состояний, коммуникации и деятельности. Работ, в которых анализируется практический подход, гораздо меньше, что открывает большие перспективы для исследований.

Учитывая все вышесказанное, можно отметить, что наибольший интерес представляет задача генерации аппаратных тестов из UML, которая и рассматривается в статье.

Использование UML для проектирования аппаратных систем

Генерация аппаратных тестов из спецификации системы — одна из разновидностей задач, связанных с UML. Следовательно, не все элементы языка UML целесообразно использовать для решения этой задачи. Рассмотрим диаграммы UML, которые отражают структуру и функциональность системы, достаточную для генерации кода и, в частности, тестов. В источниках, упомянутых в предыдущем разделе, используются следующие шесть диаграмм UML: диаграмма классов, диаграмма состояний, диаграмма деятельности, диаграмма коммуникации, диаграмма последовательности и диаграмма синхронизации. Поэтому можно предположить, что и для нашей задачи понадобятся диаграммы из этого набора.

Для выделения набора диаграмм, необходимых для генерации аппаратных тестов, определим, какие данные требуются для создания таких тестов.

• Информация об интерфейсе системы (входы, выходы, параметры), поскольку тест будет воздействовать на систему через ее интерфейсную часть. Для задания такой информации лучше всего подходит диаграмма классов, поскольку она создана для описания статических объектов.

• Информация о последовательности тестовых воздействий на устройство, т. е. об алгоритме тестирования. Алгоритмы описываются в UML с помощью диаграмм деятельности.

• Оценка реакции системы на внешние воздействия во время тестирования. Для этой цели необходимо описать правильное поведение устройства с точностью до внутренних сигналов. Это можно сделать с помощью диаграммы последовательности, которая позволяет устанавливать в произ-

вольной форме причинно-следственные связи между различными событиями.

Таким образом, в нашем случае достаточно применения трех видов диаграмм: диаграммы классов, диаграммы деятельности и диаграммы последовательности. Рассмотрим кратко методы использования этих диаграмм в описании аппаратных систем.

Два варианта использования диаграммы классов для описания интерфейса простого устройства (счетчика с параллельной загрузкой) показаны на рис. 3.

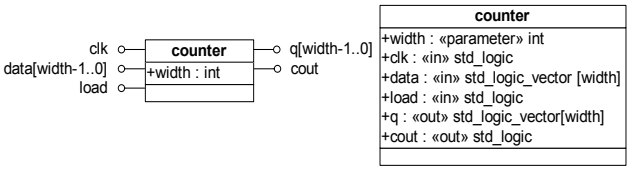
На примере слева для отображения входов и выходов счетчика используются объекты типа «интерфейс», а параметр устройства моделируется с помощью атрибута объекта. На примере справа входы, выходы, а также параметры устройства моделируются с помощью атрибутов с разными нестандартными для **UML типами и стереотипами**. Предлагается использовать следующие стереотипы: «parameter» для параметров устройства, «in» для входов устройства и «out» для его выходов. Для атрибутов со стереотипами «in» и «out» возможно также использование нестандартных типов — std_logic и std_logic_vector. Данные типы полностью соответствуют одноименным типам из библиотеки IEEE std_logic_1164 для VHDL [10], включаемой во все средства проектирования на VHDL.

Первый способ более универсальный: его можно использовать для формирования структурных спецификаций устройства, соединяя входы и выходы устройств линиями ассоциации. Второй способ проще для генерации кода: в самом деле, все интерфейсные характеристики устройства находятся внутри одного объекта.

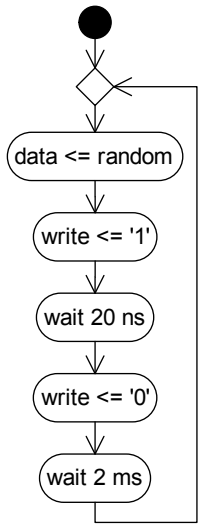
Пример использования диаграммы деятельности для описания воздействий, которые подаются на некоторое устройство в процессе его нормальной работы, показан на рис. 4. Диаграмма деятельности позволяет синтезировать генерирующую часть теста, которая является сценарием тестирования.

Пример использования диаграммы последовательности для описания работы некоторого абстрактного передатчика показан на рис. 5. Диаграмма последовательности используется для отображения внутреннего поведения устройства, для пояснения взаимосвязи между возникающими в нем управляющими сигналами.

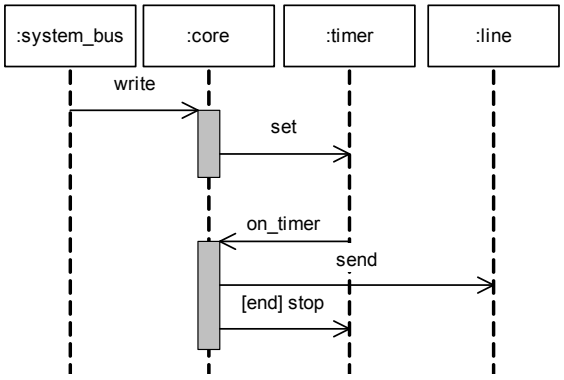
Прямоугольники в верхней части диаграммы отображают компоненты устройства. Вертикальные пунктирные линии — «линии жизни». Стрелками обозначаются сообщения, которыми обмениваются компоненты устройства. Сообщение трактуется как переход одного управляющего сигнала из неактивного состояния в активное (например, из «0» в «1»). Вертикально вытянутые



■ ***Рис. 3.*** *Использование диаграммы классов UML для описания интерфейса счетчика*



■ ***Рис. 4.*** *Использование диаграммы деятельности для описания воздействий на устройство*



■ ***Рис. 5.*** *Использование диаграммы последовательности для описания поведения устройства*

прямоугольники — спецификации исполнения (execution specification). **В нотации UML 2.0 данными** элементами обозначаются части линии жизни объекта, представляющие целостный период его работы. В нашем случае они трактуются как элементы, объединяющие сообщения в причинно-следственные связи. Такая трактовка почти не отступает от стандартной трактовки спецификации исполнения, однако не предусматривает задания в явном виде некоторых ограничений. Пример из рассматриваемого рисунка: после сиг-

нала write, направляемого в core, указанный узел отправляет сигнал set таймеру.

Диаграмма взаимодействия может дополняться разнообразными условиями, которые могут по-разному трактоваться (в зависимости от реализации). В рассмотренном рисунке условие сообщения stop записано в квадратных скобках, что является отступлением от нотации UML 2.0 (но допустимо в UML 1.x). Для обеспечения совместимости такой записи с UML 2.0 условия можно задавать как часть имени сообщения. Также могут добавляться временные требования, ограничения, счетчики и т. д.

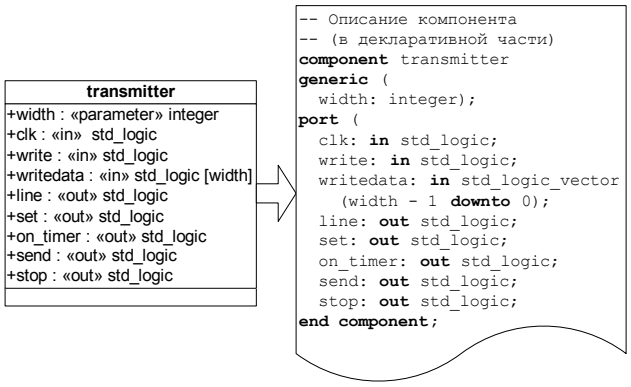
Использование диаграммы взаимодействия дает возможность генерировать наблюдающую часть теста — собственно верификатор, который проверяет последовательность управляющих сигналов, сравнивая ее с эталонной. Иными словами, представленный подход позволяет тестировать поток управления цифрового устройства. Для тестирования других аспектов системы (потока данных, быстродействия, пропускной способности и т. д.) необходимо использовать другие специализированные тесты.

Генерация тестов аппаратуры из диаграмм UML

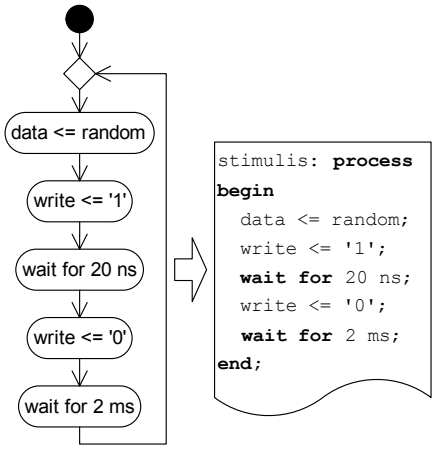
Один из наиболее распространенных языков для создания тестов аппаратуры — VHDL Testbench. Тестирование с помощью данного языка поддерживается многими средствами проектирования и моделирования аппаратуры. В дальнейшем изложении мы будем опираться на этот стандарт как на целевой.

Как говорилось ранее, для генерации тестов используется три вида диаграмм UML: диаграмма классов для извлечения интерфейсной информации, диаграмма деятельности для извлечения последовательности тестовых воздействий и диаграмма последовательности для извлечения поведенческой информации. Рассмотрим использование этих видов диаграмм в задаче генерации тестов.

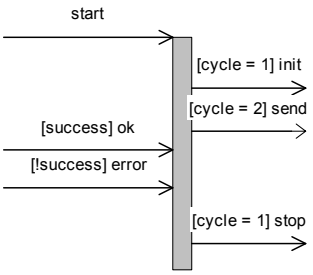
Применение диаграммы классов в генерировании интерфейсной части testbench (описание интерфейса тестируемого устройства) демонстрирует рис. 6. Тестируемое устройство — передатчик из рис. 5. Необходимо отметить, что, поскольку в testbench тестируемое устройство рассматривается как цельный блок, его внутренние сигналы, фигурирующие в диаграмме последовательности, недоступны. Поэтому, чтобы сделать их наблюдаемыми, необходимо вынести их на интерфейсный уровень, что должно найти отражение как на диаграмме классов, так и в коде устройства. Например, на рассматриваемом рисунке



■ ***Рис. 6.*** *Генерация интерфейсной части testbench из диаграммы классов UML*



■ ***Рис. 7.*** *Пример использования диаграммы деятельности для синтеза генерирующей части теста*




```
if ((state = idle) and (start = '1')) then
  cycle := 0;
  state <= after_cluster_0;
elsif ((state = after_cluster_0)
    and ((ok = '1') or (error = '1'))) then
  cycle := 0;
  state <= after_cluster_2;
elsif ((state = after_cluster_2)
    and (stop = '1')) then
  state <= idle;
else
  cycle <= cycle + 1;
end if;
```

■ ***Рис. 8.*** *Генерация верификатора из одной спецификации исполнения*

сигналы set, on_timer и др. являются внутренними, вынесенными на интерфейсный (внешний) уровень.

Рассмотрим использование диаграмм деятельности для синтеза генерирующей части теста. Поскольку диаграммы деятельности представляют собой схемы алгоритмов, их трансляция в код не представляет сложностей (рис. 7).

Рассмотрим использование диаграмм последовательности для генерации наблюдающей (верифицирующей) части теста. Генератор этой части теста оперирует со спецификациями исполнения, которые объединяют сообщения в причинно-следственные цепочки. Каждая такая цепочка транслируется в конечный автомат. «Входящие» в спецификацию исполнения сообщения управляют конечным автоматом, т. е. приводят к изменению его состояния. Входящие сообщения считаются причинами. «Исходящие» сообщения считаются следствиями, их наличие проверяется конечным автоматом, когда он находится в определенных состояниях. Поясним сказанное примером.

На рис. 8 показана часть некоторой диаграммы последовательности — спецификация исполнения, расположенная на линии жизни некоторого гипотетического управляющего устройства. Ниже приведен соответствующий ему код, управляющий состоянием верификатора.

Как видно из рисунка, вся спецификация исполнения делится на «кластеры», каждый из которых — это группа смежных входящих сообщений. Каждому кластеру ставится в соответствие одно состояние управляющего конечного автомата. Переход от одного состояния (кластера) к другому происходит под воздействием входящих сообщений за исключением самого последнего сообщения, которое используется для остановки верификатора независимо от того, какое это сообщение.

Верификатор может использовать ряд внутренних счетчиков-переменных. В рассматриваемом примере используется счетчик cycle, значение которого равно количеству тактов после последнего входящего сообщения. Также используется объект success, который может быть сигналом, счетчиком, внутренней переменной.

Каждое исходящее сообщение при генерации порождает одно проверяющее утверждение (assert ... report ... severity). Например, второе исходящее сообщение ([cycle = 2] send) порождает утверждение вида

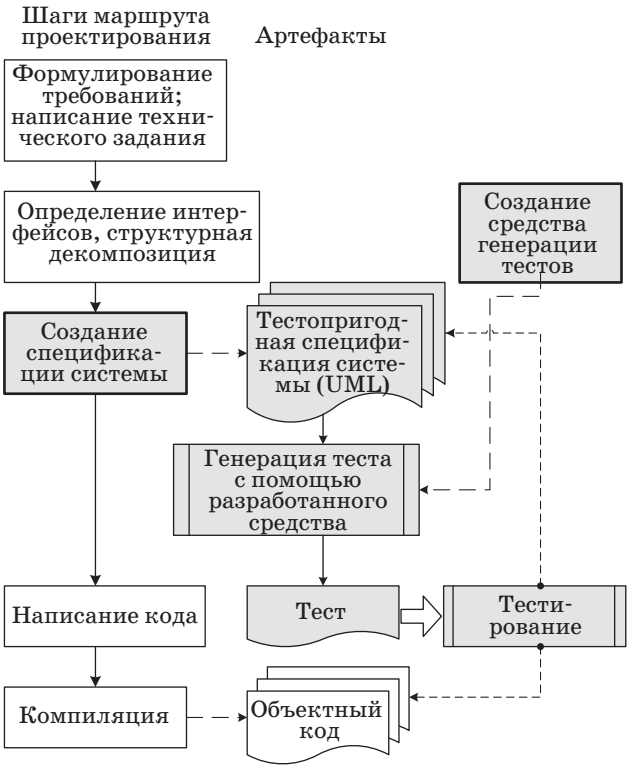
```
if ((state = after_cluster_0) and (cycle = 2)) then
    assert (send = '1')
    report ("send error")
    severity error;
end if;
```

Таким образом, верификатор проверяет, что в заданное время при заданных условиях сообщение, которое должно появиться согласно спецификации, действительно появляется в устройстве.

Применение методики

Применение методики включает в себя два этапа, на каждом из которых данная методика применяется полностью, но с разной целью. Первый раз — для создания спецификации системы, которая может быть использована для генерации теста («тестопригодная» спецификация); второй раз — для создания средства преобразования этой спецификации в аппаратный тест. Применение методики на первом этапе полностью вписывается в маршрут проектирования цифровой системы и соответствует схеме, представленной на рис. 2. Реализация второго этапа выполняется независимо, один раз. Первый этап выполняется на основании материала, изложенного в разд. «Использование UML для проектирования аппаратных систем»; второй этап — с использованием материала из разд. «Генерация тестов аппаратуры из диаграмм UML».

Полная схема применения методики (оба этапа) представлена на рис. 9. Используются следующие обозначения:



■ ***Рис. 9.*** *Схема применения методики*

• этапы применения методики подсвечены серым и обведены утолщенной рамкой;

• процессы, которые выполняются автоматизированно благодаря применению методики, также подсвечены и выделены курсивом;

• остальные подсвеченные элементы — это артефакты, получаемые в результате применения методики.

Выбор технических средств для создания генератора тестов

Для создания генератора тестов подходит любое средство проектирования UML, которое поддерживает генерацию кода на произвольном языке. На данный момент имеется большое количество таких средств, как платных, так и свободных. Многие средства поддерживают импорт и экспорт моделей через формат XMI, который является разновидностью XML для обмена метамоделями [11], и это позволяет отделить процесс разработки UML-диаграммы от ее преобразования в тест. В ряде работ (например, в [5]) предлагается использовать XSLT для преобразования XMI-описания в произвольный код.

Заключение

В данной статье была предложена методика создания синтеза поведенческих функциональных тестов на основе высокоуровневой спецификации UML. Главный акцент сделан на генерации тестов для аппаратуры, однако принципы, представленные в работе, могут быть применены и к программному обеспечению. Применение методики позволит создавать тесты систем, проверяющие их соответствие спецификациям, созданным ранее.

Литература

1. **UML 2.0** // Object Management Group. http://www.omg.org/spec/UML/2.0/ (дата обращения: 6.04.2010).
2. **Шопырин Д. Г., Шалыто А. А.** Объектно-ориентированный подход к автоматному программированию // Информационно-управляющие системы. 2003. № 5. С. 29–39.
3. **Леонтьев А. Е.** Применение UML при проектировании встраиваемых систем цифровой обработки сигналов // Информационно-управляющие системы. 2004. № 2. С. 38–44.
4. **Coyle F. P., Thornton M. A.** From UML to HDL: a Model Driven Architectural Approach to Hardware-Software Co-Design // Information Systems: New Generations Conf., 4–6 April 2005. P. 88–93.
5. **Rieder M. et. al.** Synthesized UML, a Practical Approach to Map UML to VHDL // Rapid Integration of Software Engineering: Lecture Notes in Computer Science. — Berlin, Germany: Springer, 2006. Vol. 3943. P. 203–217.
6. **Unified** Modeling Language: Tools // DMOZ: open directory project. http://www.dmoz.org/Computers/Programming/Methodologies/Modeling_Languages/Unified_Modeling_Language/Tools/ (дата обращения: 6.04.2010).
7. **Pires W., Brunet J., Ramalho F.** UML-based Design Test Generation // Proc. of the 2008 ACM Symp. on Applied Computing, Fortaleza, Ceara, Brazil, 16–20 Mar. 2008 / Association for Computer Machinery (ACM). — N. Y., USA, 2008. P. 735–740.
8. **Chen M., Mishra P.** Coverage-driven Automatic Test Generation for UML Activity Diagrams // Proc. of the 18th ACM Great Lakes Symp. on VLSI, Orlando, Florida, USA, 4–6 May 2008 / Association for Computer Machinery (ACM). N. Y., USA, 2008. P. 139–142.
9. **Gnesi S., Latella D., Massink M.** Formal Test-case Generation for UML Statecharts // Proc. of the Ninth IEEE Intern. Conf. on Engineering Complex Computer Systems Navigating Complexity in the e-Engineering Age, 14–16 April 2004 / IEEE Computer Society. Washington, D. C., USA, 2004. P. 75–84.
10. **std**_logic_1164 multi-value logic system. http://standards.ieee.org/downloads/1076/1076.2-1996/std_logic_1164.vhdl (дата обращения: 6.04.2010).
11. **XML** Metadata Interchange. http://www.omg.org/technology/documents/formal/xmi.htm (дата обращения: 6.04.2010).

УДК 004.4′242

МЕТОД ПОСТРОЕНИЯ УПРАВЛЯЮЩИХ КОНЕЧНЫХ АВТОМАТОВ НА ОСНОВЕ ТЕСТОВЫХ ПРИМЕРОВ С ПОМОЩЬЮ ГЕНЕТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

***Ф. Н. Царев*[1],**
аспирант
Санкт-Петербургский государственный университет информационных технологий, механики и оптики

Предлагается метод построения управляющих конечных автоматов на основе тестовых примеров с помощью генетического программирования.

Приводятся описания представления автоматов в виде особей алгоритма генетического программирования, операций мутации и скрещивания, а также генетического алгоритма. Применение метода иллюстрируется на примере построения автомата управления часами с будильником.

Ключевые слова *— генетическое программирование, автоматное программирование, машинное обучение.*

Введение

Автоматное программирование — парадигма программирования, в рамках которой программные системы предлагается строить в виде набора взаимодействующих автоматизированных объектов управления [1]. Автоматизированный объект управления состоит из управляющего конечного автомата и объекта управления. Таким образом, поведение каждого автоматизированного объекта управления во многом описывается детерминированным конечным автоматом.

Для большинства задач автоматы удается строить эвристически вручную. Однако в ряде случаев такое построение слишком трудоемко или приводит к неоптимальным результатам. К таким задачам относятся, например, задачи «Умный муравей» [2, 3], «Умный муравей-3» [4] и задача об управлении моделью беспилотного летательного аппарата [5]. Для построения автоматов в таких задачах можно применять генетические алгоритмы (ГА) [6–8].

Традиционный метод построения конечных автоматов с помощью ГА [3, 9–11] использует вычисление функции приспособленности на основе моделирования работы системы со сложным поведением в некоторой внешней среде. Главным недостатком этого метода является то, что при его применении функцию приспособленности необходимо «с нуля» реализовывать для каждой задачи. Кроме того, такой подход к вычислению функции приспособленности связан с большими затратами вычислительных ресурсов.

Целью настоящей работы является разработка метода построения конечных автоматов на основе генетического программирования, в котором устранены указанные недостатки. Для достижения этой цели предлагается осуществлять построение конечных автоматов на основе тестовых примеров.

Постановка задачи

При применении парадигмы автоматного программирования для реализации сущности со сложным поведением выделяется система управления и объект управления. На начальном этапе проектирования программы выделяются события (e1, e2, ...), входные переменные (x1, x2, ...) и выходные воздействия (z1, z2, ...). После этого проектирование программы может идти разными путями. Один из них состоит в написании сценария работы программы, по которому далее эвристически строится автомат. Пример построения автомата таким способом приведен в работе [12].

[1] Научный руководитель — доктор технических наук, профессор, заведующий кафедрой технологий программирования Санкт-Петербургского государственного университета информационных технологий, механики и оптики *А. А. Шалыто*.

Другой подход, который практически не применяется для построения автоматных программ, но достаточно широко распространен при традиционной разработке программ, называется «разработкой на основе тестов» (*test-driven development*) [13]. При его использовании процесс написания кода на языке программирования идет параллельно с написанием тестов для программы, а добавление функциональности в программу осуществляется только после того, как создан тест для проверки этой функциональности. Таким образом, функциональность программы описывается набором тестов для нее.

В случае применения автоматного программирования в качестве тестов для управляющего конечного автомата естественно рассматривать пары последовательностей, одна из которых описывает события и входные переменные, поступающие на вход автомату, а вторая — выходные воздействия, которые должен вырабатывать автомат при обработке входных воздействий. Таким образом, задача построения управляющего конечного автомата становится похожей на задачу построения конечного преобразователя, для решения которой успешно используются ГА [14].

Описание предлагаемого метода

Исходными данными для построения конечного автомата управления системой со сложным поведением являются:

— список событий;

— список входных переменных;

— список выходных воздействий;

— набор тестов Tests, каждый из которых содержит последовательность Input[*i*] событий, поступающих на вход конечному автомату, и соответствующую ей эталонную последовательность Answer[*i*] выходных воздействий.

Отметим, что при использовании описываемого метода входные переменные явным образом не задаются. Для учета входных переменных необходимо добавить в список событий новые события, объединяющие исходные события и логические формулы, содержащие входные переменные. Например, если в списке событий присутствует событие e1, а в списке входных переменных — x1, то новым событием может быть e1 [x1] (произошло событие e1 и переменная x1 истинна).

Отметим также, что для тестов, которые задаются для построения конечного автомата, справедливо свойство, которое можно сформулировать следующим образом: «префиксы тестов являются тестами» — если из входной последовательности событий удалить часть событий, находящихся в ее конце, то результат обработки автоматом этой последовательности будет префиксом исходной выходной последовательности.

Представление конечного автомата в виде хромосомы ГА. Конечный автомат в алгоритме генетического программирования представляется в виде объекта, который содержит описания переходов для каждого состояния и номер начального состояния. Для каждого состояния хранится список переходов. Каждый переход описывается событием, при поступлении которого этот переход выполняется, и числом выходных воздействий, которые должны быть сгенерированы при выборе этого перехода.

Таким образом, в особи кодируется только «скелет» управляющего конечного автомата, а конкретные выходные воздействия, вырабатываемые на переходах, определяются с помощью алгоритма расстановки пометок, который аналогичен предложенному в работе [15].

Выбор представления графа переходов автомата с помощью списков ребер (в отличие от работы [15], в которой применялись полные таблицы переходов) обоснован тем, что, как правило, в автоматах управления системами со сложным поведением не в каждом состоянии определена реакция на каждое событие.

Алгоритм расстановки пометок. Опишем алгоритм расстановки пометок на переходах, применяемый в настоящей работе. Как было сказано выше, для каждого перехода в особи ГА записано, сколько выходных воздействий должно вырабатываться при его выборе. Подадим на вход конечному автомату последовательность событий, соответствующую одному из тестов, и будем наблюдать за тем, какие переходы выполняет автомат. Зная эти переходы и информацию о том, сколько выходных воздействий должно быть сгенерировано на каждом переходе, можно определить, какие выходные воздействия должны вырабатываться на переходах, использовавшихся при обработке входной последовательности.

Для каждого перехода T и каждой последовательности выходных воздействий zs вычисляется величина C[T][zs] — число случаев, в которых при обработке входной последовательности, соответствующей одному из тестов, на переходе T должны быть выработаны выходные воздействия, образующие последовательность **zs. Далее**, каждый переход помечается той последовательностью zs_0, для которой величина $C[T][zs_0]$ максимальна.

Функция приспособленности. Функция приспособленности основана на редакционном расстоянии (расстоянии Левенштейна) [16]. Редакционным расстоянием между двумя последовательностями символов называется минимальное число операций замены символа, вставки симво-

ла и удаления символа, которые необходимо выполнить над первой последовательностью для того, чтобы она совпала со второй.

Для вычисления функции приспособленности выполняются следующие действия: на вход автомату подается каждая из последовательностей Input[i]. Обозначим последовательность выходных воздействий, которую сгенерировал автомат при входе Input[i], как Output[i]. После этого вычисляется величина

$$FF_1 = \frac{\sum_{i=1}^{n}\left(1 - \frac{ED(\text{Output}[i], \text{Answer}[i])}{\max(|\text{Output}[i]|, |\text{Answer}[i]|)}\right)}{n},$$

где $ED(A, B)$ — редакционное расстояние между строками A и B. Отметим, что значения функции FF_1 лежат в пределах от 0 до 1. При этом, чем «лучше» автомат соответствует тестам, тем больше значение функции приспособленности.

Функция приспособленности зависит не только от того, насколько «хорошо» автомат работает на тестах, но и от числа переходов, которые он содержит. Эта функция вычисляется следующим образом:

$$FF_2 = \begin{cases} 0{,}5 \cdot T \cdot FF_1 + \frac{1}{M}(\text{M} - cnt), FF_1 < 1 \\ T + \frac{1}{M}(M - cnt), FF_1 = 1 \end{cases},$$

где T — «стоимость» прохождения всех тестов; M — произвольное целое число, большее максимального числа переходов в автомате; cnt — число переходов в автомате. При проведении вычислительных экспериментов были выбраны следующие значения: $T = 20$, $M = 100$.

Эта функция приспособленности устроена таким образом, что при одинаковом значении функции FF_1, отражающей «прохождение» тестов автоматом, преимущество имеет автомат, содержащий меньшее число переходов. Кроме этого, автомат, который «идеально» проходит все тесты, оценивается выше, чем автомат, проходящий тесты не идеально.

Операция мутации. При выполнении операции мутации с заданной вероятностью (по умолчанию, она равна 0,05) выполняется каждое из действий:

— изменение начального состояния;

— изменение описания каждого из переходов;

— удаление или добавление перехода для каждого из состояний.

После выполнения операции мутации может возникнуть ситуация, когда в автомате из одного состояния присутствуют два перехода по одному и тому же событию. Для устранения таких переходов применяется операция удаления дублирующихся переходов.

Операция удаления дублирующихся переходов. В целях удаления дублирующихся переходов для каждого состояния выполняются следующие операции: последовательно просматривается список переходов из этого состояния, при этом запоминаются события, переходы по которым определены для этого состояния. Если очередной переход Tr происходит по событию, для которого в списке уже есть переход, то переход Tr удаляется из списка.

Операция скрещивания. Скрещивание описаний автоматов производится следующим образом. Обозначим как P1 и P2 — «родительские» особи, а **S1 и S2 — особи-«потомки». Для начальных** состояний S1.is и S2.is автоматов S1 и S2 будет верно одно из двух соотношений:

S1.is = P1.is и S2.is = P2.is;

S1.is = P2.is и S2.is = P1.is.

Опишем, как устроены переходы автоматов S1 и **S2. Скрещивание описаний автоматов производ**ится отдельно для каждого состояния. Обозначим список переходов из состояния номер i автомата P1 как P1.T[i], а список переходов из состояния номер i автомата P2 как P2.T[i]. Для выполнения «скрещивания переходов» с равной вероятностью может быть выбран один из двух методов.

При использовании *традиционного метода скрещивания* списки переходов S1.T[i] и S2.T[i] строятся следующим образом.

1. Строится общий список переходов, в который помещаются переходы, входящие как в P1.T[i], так и в P2.T[i].

2. К полученному списку применяется случайная перестановка.

3. Далее возможны два равновероятных варианта:

• либо в S1.T[i] помещаются первые $|$P1.T[i]$|$ переходов из полученного списка, а в S2.T[i] — оставшиеся переходы;

• либо в S1.T[i] помещаются первые $|$P2.T[i]$|$ переходов из полученного списка, а в S2.T[i] — оставшиеся переходы.

При использовании *метода скрещивания с учетом тестов* списки переходов S1.T[i] и S2.T[i] строятся следующим образом.

1. Составляется список всех используемых тестов, упорядоченный по возрастанию нормированного редакционного расстояния между «правильным ответом» Answer и последовательностью Output выходных воздействий, генерируемой автоматом, — значения выражения $\frac{\text{ED}(\text{Output}[i], \text{Answer}[i])}{\max(|\text{Output}[i]|, |\text{Answer}[i]|)}$. В автоматах P1 и P2 помечаются те переходы, которые используются при обработке первых 10 % тестов из полученного упорядоченного списка.

2. *Помеченные* переходы копируются в S1.*T*[*i*] и S2.*T*[*i*] напрямую.

3. Строится общий список переходов, в который помещаются *непомеченные* переходы, входящие как в P1.*T*[*i*], так и в P2.*T*[*i*].

4. К полученному списку *L* применяется случайная перестановка.

5. Список S1.*T*[*i*] дополняется первыми переходами из списка *L* до размера |P1.*T*[*i*]|, а список S2.*T*[*i*] дополняется оставшимися переходами.

В обоих случаях к получившимся в результате скрещивания автоматам S1 и S2 применяется операция удаления дублирующихся переходов.

Пример применения предлагаемого метода

Применение предлагаемого метода иллюстрируется на примере построения автомата управления часами с будильником [1]. Эти часы имеют три кнопки (помеченные буквами «А», «Н», «М»), которые предназначены для изменения режима их работы и для настройки текущего времени или времени срабатывания будильника. Если будильник выключен, то кнопки «Н» и «М» служат для установки текущего времени, а кнопка «А» переводит часы в режим «Настройка будильника», в котором кнопки **«Н» и «М» устанавливают** не текущее время, а время срабатывания будильника. Повторное нажатие кнопки **«А» включает** будильник. После этого если текущее время совпадает со временем срабатывания будильника, то включается звонок, который отключается либо нажатием кнопки **«А», либо само**произвольно через минуту. Кроме этого, нажатие кнопки «А» приводит к выключению будильника.

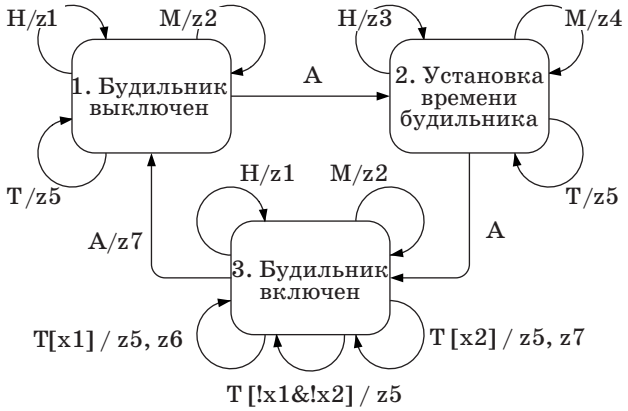
Рассматриваемые часы с будильником являются системой со сложным поведением, так как в ответ на одни и те же входные события (нажатия кнопок) в зависимости от режима работы генерируются различные выходные воздействия. Поведение этих часов может быть описано с помощью конечного автомата [1], который содержит три состояния (рис. 1).

Система управления часами с будильником имеет четыре события:

- H — нажата кнопка «Н»;
- M — нажата кнопка «М»;
- A — нажата кнопка «А»;
- T — генерируется таймером каждую секунду.

Кроме этого, она содержит две входные переменные:

- x1 — верно ли, что текущее время совпадает со временем срабатывания будильника;
- x2 — верно ли, что текущее время на минуту больше времени срабатывания будильника?



■ ***Рис. 1.*** *Граф переходов автомата управления часами с будильником*

Число выходных воздействий равно семи:

- z1 — увеличить на единицу число часов в текущем времени;
- z2 — увеличить на единицу число минут в текущем времени;
- z3 — увеличить на единицу число часов во времени срабатывания будильника;
- z4 — увеличить на единицу число минут во времени срабатывания будильника;
- z5 — прибавить минуту к текущему времени;
- z6 — включить звонок будильника;
- z7 — выключить звонок будильника.

Система тестов для построения автомата управления часами с будильником состояла из 38 тестов, описывающих его работу во всех трех режимах. В качестве примера приведем тесты для состояния «Будильник выключен» (таблица).

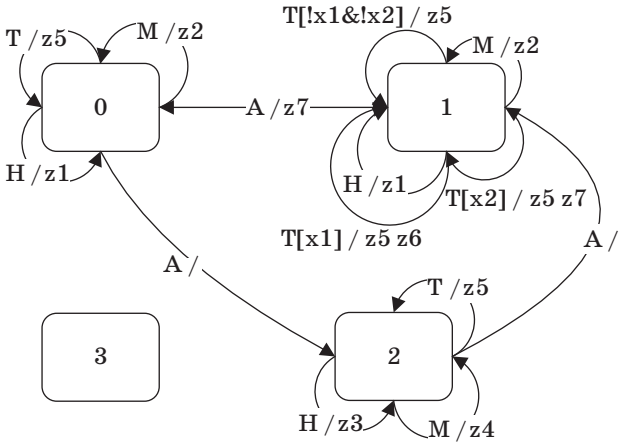
Построение конечного автомата управления часами с будильником проводилось при следующих параметрах алгоритма генетического программирования:

- размер поколения — 2000 особей;
- доля «элиты» — наиболее приспособленных особей, напрямую переходящих в следующее поколение, — 10 %;
- число поколений до малой «мутации поколения» — 100 поколений;
- число поколений до большой «мутации поколения» — 150 поколений;
- размер автоматов в начальном поколении — четыре состояния.

Было проведено 1000 запусков алгоритма с указанными параметрами. Цель в каждом из них состояла в том, чтобы построить автомат, содержащий 14 переходов и соответствующий всем тестам (значение функции приспособленности, соответствующее такому автомату, — 20.86). На каждом из запусков алгоритма генетического программирования был построен автомат (рис. 2), в котором из начального (нулевого) состояния до-

■ *Тесты для состояния «Будильник выключен»*

	Входная последовательность	Выходная последовательность	Комментарий
1	*T, T, T, T*	z5, z5, z5, z5	Описывает обработку часами события «Сработал таймер». При возникновении этого события текущее время должно быть увеличено на минуту
2	*H, H, H, H*	z1, z1, z1, z1	Описывает обработку часами нажатия кнопки «H». При нажатии на эту кнопку число часов в текущем времени должно быть увеличено на единицу
3	*M, M, M, M*	z2, z2, z2, z2	Описывает обработку часами нажатия кнопки «M». При нажатии на эту кнопку число минут в текущем времени должно быть увеличено на единицу
4	*T, M, H, T, T, T, M, T, H, H, T, M*	z5, z2, z1, z5, z5, z5, z2, z5, z1, z1, z5, z2	Описывает обработку событий H, M и T в состоянии «Будильник выключен»
5	*A, A, A, T, T, T, T*	z7, z5, z5, z5, z5	После трех нажатий кнопки **«A» часы должны находиться в состоянии** «Будильник выключен». Аналог первого теста
6	*A, A, A, H, H, H, H*	z7, z1, z1, z1, z1	После трех нажатий кнопки **«A» часы должны находиться в состоянии** «Будильник выключен». Аналог второго теста
7	*A, A, A, M, M, M, M*	z7, z2, z2, z2, z2	После трех нажатий кнопки **«A» часы должны находиться в состоянии** «Будильник выключен». Аналог третьего теста

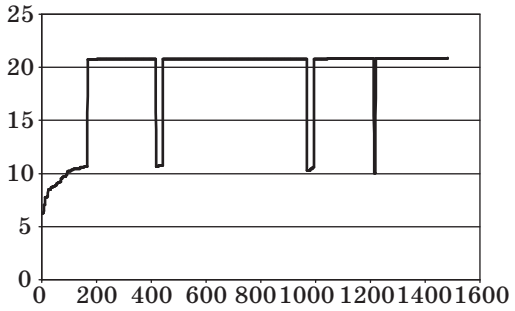


■ ***Рис. 2.*** *Граф переходов автомата, построенного с помощью алгоритма генетического программирования*

стижимы только три состояния из четырех. Если удалить недостижимое состояние, то этот граф переходов будет изоморфен графу переходов, построенному вручную.

График зависимости максимального значения функции приспособленности от номера поколения представлен на рис. 3 при одном из запусков алгоритма.

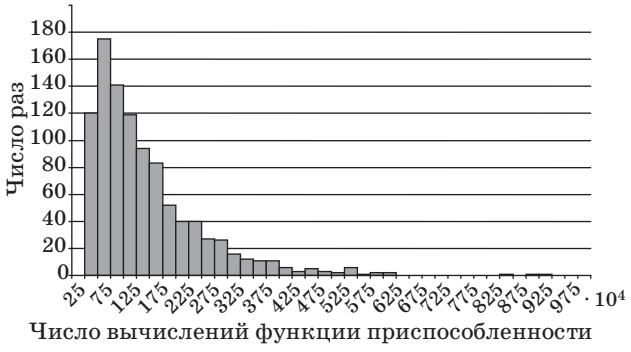
Как видно из графика, автоматы, входящие в начальное поколение, проходят тесты примерно наполовину. Примерно к двухсотому поколению был построен автомат, полностью проходящий все тесты, однако содержащий достаточно большое число переходов. Далее шел процесс уменьшения числа переходов, во время которого три раза (в районе 400-, 1000- и 1200-го поколений) к популяции применялась операция «большой мутации», в результате чего значение функции



■ ***Рис. 3.*** *Зависимость максимального значения функции приспособленности от номера поколения*

приспособленности уменьшалось примерно до десяти. В итоге в 1482-м поколении был построен автомат, полностью проходящий все тесты и содержащий 14 переходов.

Для каждого из запусков запоминалось число вычислений функции приспособленности (которое равно числу просмотренных во время работы авто-



■ ***Рис. 4.*** *Распределение числа вычислений функции приспособленности в проведенных вычислительных экспериментах*

матов) в процессе построения автомата. На рис. 4 показано распределение этой величины, полученное в результате вычислительных экспериментов.

Минимальное значение числа вычислений функции приспособленности составило 256 063, максимальное — 9 239 523. В предположении, что эта величина распределена по логнормальному распределению, ее среднее значение составляет 1 443 351.20 (стандартное отклонение — 1 103 401.82).

Заключение

В работе предложен метод построения автоматов управления системами со сложным поведением с учетом тестов с помощью генетического программирования. При использовании разработанного метода для построения управляющих конечных автоматов необходимо задать только тестовые примеры, а вычисление функции приспособленности требует существенно меньше вычислительных ресурсов, чем при использовании моделирования для вычисления функции приспособленности. Рассмотрен пример применения разработанного метода.

Исследования проводятся по государственному контракту, выполняемому в рамках Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России на 2009–2013 годы».

Литература

1. **Поликарпова Н. И., Шалыто А. А.** Автоматное программирование. — СПб.: Питер, 2010. — 176 с.
2. **Jefferson D. et al.** The Genesys System: Evolution as a Theme in Artificial Life // Proc. of Second Conf. on Artificial Life. MA: Addison-Wesley, 1992. P. 549–578. www.cs.ucla.edu/~dyer/Papers/AlifeTracker/Alife91Jefferson.html (дата обращения: 19.03.2010).
3. **Царев Ф. Н., Шалыто А. А.** Применение генетического программирования для генерации автомата в задаче об «Умном муравье» // Интегрированные модели и мягкие вычисления в искусственном интеллекте: Сб. тр. IV Междунар. науч.-практ. конф. Т. 2. М.: Физматлит. 2007. С. 590–597. http://is.ifmo.ru/genalg/_ant_ga.pdf (дата обращения: 19.03.2010).
4. **Бедный Ю. Д., Шалыто А. А.** Применение генетических алгоритмов для построения автоматов в задаче «Умный муравей». http://is.ifmo.ru/works/ant (дата обращения: 19.03.2010).
5. **Параценко Д. А., Царев Ф. Н., Шалыто А. А.** Технология моделирования одного класса мультиагентных систем на основе автоматного программирования на примере игры «Соревнование летающих тарелок»: Проектная документация / СПбГУ ИТМО. 2006. http://is.ifmo.ru/unimod-projects/plates (дата обращения: 19.03.2010).
6. **Гладков Л. А., Курейчик В. В., Курейчик В. М.** Генетические алгоритмы. — М.: Физматлит, 2006. — 320 с.
7. **Рассел С., Норвиг П.** Искусственный интеллект: современный подход. — М.: Вильямс, 2006. — 1407 с.
8. **Koza J. R.** Genetic programming: on the programming of computers by means of natural selection. — MIT Press, 1992. — 835 с.
9. **Поликарпова Н. И., Точилин В. Н., Шалыто А. А.** Применение генетического программирования для генерации автоматов с большим числом входных переменных // Науч.-техн. вестник СПбГУ ИТМО. 2008. Вып. 53. Автоматное программирование. С. 24–42.
10. **Данилов В. Р.** Метод представления автоматов деревьями решений для использования в генетическом программировании // Науч.-техн. вестник СПбГУ ИТМО. 2008. Вып. 53. Автоматное программирование. С. 103–108.
11. **Давыдов А. А., Соколов Д. О., Царев Ф. Н.** Применение генетических алгоритмов для построения автоматов Мура и систем взаимодействующих автоматов Мили на примере задачи об «Умном муравье» // Науч.-техн. вестник СПбГУ ИТМО. 2008. Вып. 53. Автоматное программирование. С. 108–114.
12. **Мазин М. А., Парфенов В. Г., Шалыто А. А.** Разработка интерактивных приложений Macromedia Flash на базе автоматной технологии: Проектная документация / СПбГУ ИТМО. 2003. http://is.ifmo.ru/projects/flash/ (дата обращения: 19.03.2010).
13. **Бек К.** Экстремальное программирование: разработка через тестирование. — СПб.: Питер, 2003. — 224 с.
14. **Lucas S., Reynolds T.** Learning Finite State Transducers: Evolution versus Heuristic State Merging // IEEE Transactions on Evolutionary Computation. 2007. Vol. 11. Is. 3. P. 308–325.
15. **Lucas S., Reynolds T.** Learning Deterministic Finite Automata with a Smart State Labeling Evolutionary Algorithm // IEEE Transactions on Evolutionary Computation. 2005. Vol. 27. Is. 7. P. 1063 –1074.
16. **Левенштейн В. И.** Двоичные коды с исправлением выпадений, вставок и замещений символов // Докл. Академии наук СССР. 1963. № 4. С. 845–848.

УДК 004.43

МЕТОДЫ РАСШИРЕНИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ (Часть 2)

***В. Д. Михеева*[1],**
старший инженер по программному обеспечению,
Российское отделение компании «Интел»

Приводится обзор методов расширения современных языков программирования, определенных автором и использованных для построения классификации расширений по способам интеграции и исполнения кода расширений. Рассматривается метод расширения языков программирования новыми конструкциями, методы исполнения расширений, а также приводится пример предметно-ориентированного расширения языка общего назначения средствами таблично-ориентированного программирования, реализованного автором на основе средств программирования системы эфемеридных расчетов в астрономии.

Ключевые слова *— предметно-ориентированный язык программирования, расширение языка программирования, инструментальные средства программирования, таблично-ориентированное программирование.*

Метод 4. Расширение новыми языковыми конструкциями

Все три рассмотренных метода [1] внесения расширенных возможностей в БЯ не изменяют имеющихся языковых конструкций, а лишь отличаются способом интеграции основной программы с внешней реализацией расширений. Рассматриваемый здесь четвертый способ представляет собой наиболее тесную интеграцию выразительных средств БЯ с конструкциями расширений. А именно, имеется в виду буквальное совмещение программного кода на двух языках в одной программе, что удобнее пользователю (программисту) по ряду причин. Во-первых, появляется возможность выражать специализированную функциональность с помощью наиболее подходящих языковых конструкций и, во-вторых, облегчается поиск ошибок в процессе разработки приложений, поскольку синтаксический и семантический контроль конструкций расширений наряду с контролем конструкций БЯ может быть выполнен компилятором на этапе трансляции.

Обычно при таком совмещении текст на специализированном языке выделяется в программном коде на БЯ с помощью окружающих маркирующих конструкций, легко выделяемых лексически, иногда даже на этапе *препроцессирования* текста программы (этапе предварительного анализа текста программы до синтаксического разбора). В других случаях появление специализированных конструкций однозначно определяется из контекста и такого выделения не требуется — тогда в одной программе буквально появляется «смесь» языковых конструкций с внешне ничем не выделенными границами.

Совмещение языков
с явным выделением кода расширений.

Типичным примером совмещения разных языков с использованием маркирующих конструкций для явного выделения кода расширения можно считать ассемблерные вставки в код C/C++. Рассмотрим пример такого кода — программу на C для компилятора GNU (GNU C compiler — GCC) с использованием кода встроенной функции на ассемблере (пример 7) [2]. Эта программа вычисляет наибольший общий делитель двух заданных чисел с помощью алгоритма Евклида, реализованного в виде встроенной функции на ассемблере в целях оптимизации производительности исполняемого кода.

Пример 7. Программа на C со встроенным кодом на ассемблере.

```
1 #include <stdio.h>
2
3 int gcd( int a, int b ) {
4   int result ;
```

[1] Научный руководитель — кандидат физ.-мат. наук, заведующий лабораторией астрономического программирования Института прикладной астрономии РАН *Ф. А. Новиков*. Окончание. Начало в № 4.

```
5   /* Compute Greatest Common Divisor using Euclid's Algorithm */
6   asm volatile ( "movl %1, %%eax;"
7           "movl %2, %%ebx;"
8           "CONTD: cmpl $0, %%ebx;"
9           "je DONE;"
10          "xorl %%edx, %%edx;"
11          "idivl %%ebx;"
12          "movl %%ebx, %%eax;"
13          "movl %%edx, %%ebx;"
14          "jmp CONTD;"
15          "DONE: movl %%eax, %0;" : "=g" (result) : "g" (a), "g" (b)
16  );
17
18  return result ;
19 }
20
21 int main() {
22  int first, second ;
23  printf( "Enter two integers : " ) ;
24  scanf( "%d%d", &first, &second );
25
26  printf( "GCD of %d & %d is %d\n", first, second, gcd(first,
second) ) ;
27
28  return 0 ;
29 }
```

В примере 7 представлена основная C-функция main, вызывающая вспомогательную C-функцию gcd, в теле которой присутствует сегмент встроенного ассемблерного кода (строки 6–16), реализующий тело алгоритма Евклида на уровне инструкций микропроцессора. Маркером начала и конца сегмента данного языкового расширения является синтаксическая конструкция asm (...), внутри которой располагаются строки кода на языке ассемблер для микропроцессора семейства Intel x86. Заметим, что спецификатор volatile здесь не является маркером, а только указывает компилятору, что данный код не подлежит автоматической оптимизации. Существует множество стилей написания подобных ассемблерных вставок. В данном примере применена конструкция в стиле AT&T, соответствующая стандарту [3] и поддерживающаяся компилятором GCC. Компиляторами C/C++ компании Microsoft для таких расширений используется другая нотация — в стиле Intel.

Можно привести другой пример, в настоящее время еще экзотический, поскольку речь идет об инновационной разработке компании Intel — об архитектуре Intel, поддерживающей интеграцию с различными акселераторами. Это архитектура IA с технологией расширения Exoskeleton Sequencer (EXO) и специально разработанные средства программирования с поддержкой языковых расширений — C for Heterogeneous Integration (CHI) [4]. «Интегрированная среда программирования CHI предоставляет разработчикам приложений возможность встраивать специализированный ассемблерный код для акселераторов или код на предметно-ориентированном языке в исходный код на традиционном языке C/C++»[2] [4, p. 185]. В примере 8 [4] представлен фрагмент кода для CHI, содержащий встроенный код на предметно-ориентированном языке DPL (Datastream Programming Language), специально разработанном для программирования реконфигурируемого акселератора SCC-DPE [4, p. 190].

Пример 8. Программа на C со встроенным кодом на языке DPL.

```
1 float Vin[4];
2 float Vout[4];
3
4 void *in_desc = (void *)chi_alloc_buffer_desc
5       (DPE_INPUT_BUFFER, Vin, 4, 1);
6 void *out_desc = (void *)chi_alloc_buffer_desc
7       (DPE_OUTPUT_BUFFER, Vout, 4, 1);
8
9 #pragma omp parallel target(dpe) shared(Vin,Vout)
10 descriptor(in_desc,out_desc)
11 {
12   __dpl {
13     configuration[1] cfgMult( vector val[1], vector coeff[1] )
14     {
15       result bs( mull(val, coeff), 13 );
16     }
17     flow[4] multiFlow( vector vec[4], vector coeffs[4])
18     {
19       vector ret[4]; result out;
20       selector[iter : 4] sel[1] = {{ iter }};
21       selector[iter : 4] selRev[1] = {{ 3 - iter }};
22       ret[sel] = cfgMult(vec[sel], coeffs[selRev]);
23     }
24     vector cf[4] = { 0.5 + I * 0.0 };
25     program dlMain()
26     {
27       Vout = multiFlow(Vin, cf);
28     }
29   }
30 }
```

В примере 8 фрагмент кода на языке DPL (строки 12–29) выделен в программе на языке C с помощью синтаксической конструкции __dpl { ... }, по структуре аналогичной конструкции встраивания ассемблерного кода в стиле Microsoft/Intel __asm { ... }. Хотя способ оформления языковых расширений сходен, разница примеров 7 и 8 состоит в том, что в последнем случае в основную программу встраивается фрагмент кода на другом языке высокого уровня, а не команды ассемблера.

Среди разнообразия современных средств программирования можно найти еще немало примеров совмещения различных языков программирования в одной программе с явным синтаксическим разделением кода на разных языках. Например, совмещение кода на языке разметки гипертекста HTML и описания функций на языке сценариев, таких как JavaScript, VBScript.

---

[2] *Пер. авт.*

Совмещение языков
без явного выделения кода расширений.

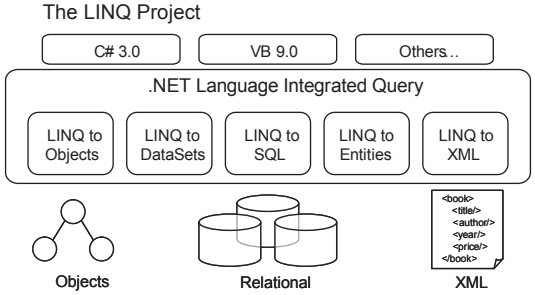
Теперь рассмотрим введение в язык расширений без маркировки на примере проекта LINQ компании Microsoft. Технология LINQ (.NET Language-Integrated Query) — это средство расширения языков на платформе .NET для интеграции с языком описания запросов [5, 6]. Эти расширения поддерживаются в языках C# 3.0 и VB 9.0 компании Microsoft, а также, возможно, появятся и в других языках на платформе .NET. В докладе одного из авторов технологии LINQ [7] рассказано, что LINQ — это абстракция, реализованная несколькими видами API, позволяющая в программе на платформе .NET единым образом оперировать с разного рода данными, выполняя традиционные для БД операции (такие как запрос, изменение и преобразование данных и т. п.). Такими данными могут быть любые .NET объекты в памяти программы, коллекции объектов, массивы данных, реляционные БД и документы XML (рисунок [7]).

Рассмотрим пример 9 [8], созданный на основе образца кода LINQ.

Пример 9. Программа на C# с конструкциями LINQ.

```
1 public void SimpleQuery()
2 {
3  Northwind db = new Northwind(...);
4  db.Log = Console.Out;
5  var query = from customer in db.Customers
6          where customer.City == "Paris"
7          select customer;
8
9  foreach (var Customer in query)  << Query Executes here
10  {
11   Console.WriteLine(Customer.CompanyName);
12  }
13 }
```

В примере 9 представлен код процедуры на языке C#, включающий конструкцию на языке описания запросов (строки 5–7, начиная с ключевого слова from), близком по синтаксису к SQL. Здесь среди конструкций C# употребляется конструкция совсем другого рода — описание запроса к БД, и заметим, без каких бы то ни было окружающих ее маркирующих символов. В этом случае конструкция расширения БЯ автоматически определяется компилятором исходя из анализа структуры выражения и по контексту (без синтаксических «подсказок» с помощью маркеров).



■ *Обзор технологии LINQ*

Примечательно, что в LINQ возможна также альтернативная запись запроса — через интерфейс API [7] (согласно первому методу интеграции расширений [1]). Поэтому строки 5–7 примера 9 (декларация переменной query, содержащей описание запроса) могут быть переписаны так, как показано в примере 10, и их исполнение приведет к получению такого же результата (т. е. обе формы записи эквивалентны по смыслу).

Пример 10. Программа на C# с конструкциями LINQ.

```
5  var query = db.Customers
6          .Where(customer => customer.City == "Paris")
7          .Select(customer => customer);
```

Методы Where и Select реализованы с помощью механизма C# введения так называемых *методов расширения* — возможности добавлять новые методы к уже описанным типам объектов. А логические выражения в скобках в вызовах методов представляют собой встроенные функции, называемые *лямбда-выражениями*.

В реализации LINQ форма описания запроса примера 9 преобразуется компилятором в конструкцию вида, описанного в примере 10, т. е. с использованием средств БЯ и применением интерфейса API (методов Where и Select), с помощью которого реализовано данное расширение.

Подводя итог, следует отметить, что для реализации описанного в этом разделе метода расширения БЯ новыми языковыми конструкциями в обоих рассмотренных вариантах необходима разработка специальных средств программирования, поддерживающих новый синтаксис и семантику расширенного языка. А это может быть достаточно трудоемкой задачей. Однако у данного метода есть неоспоримые преимущества по сравнению с представленными в первой части статьи — это максимальное удобство при программировании и возможность наиболее эффективного исполнения таких расширений.

Методы исполнения кода расширений

Помимо нескольких способов интеграции кода расширений в БЯ, можно также выделить несколько способов реализации описанных расширениями действий во время исполнения основной программы. По мнению автора, классификация расширений современных языков программирования в соответствии с основными методами исполнения их кода представляет собой список следующих категорий, перечисленных в порядке возрастания сложности реализации.

1. Программный код расширений (обычно библиотеки функций, иногда семейство классов) реализован на БЯ.

2. Расширения в виде внешних библиотек функций реализованы на другом языке программирования и, в некоторых случаях, встроены в систему исполнения.

3. Программная интерпретация расширенных возможностей во время исполнения основной программы с помощью специальной программы – интерпретатора.

4. Аппаратная интерпретация расширенных возможностей, например с помощью сопроцессора.

Первый метод исполнения расширений (для первой категории расширений) является наиболее простым с точки зрения его реализации. Он может применяться в сочетании с первым или вторым методами интеграции расширений [1]. Второй метод обычно применяется в сочетании также с первым методом интеграции расширений. Третий метод подходит для реализации расширений третьего [1] и четвертого методов интеграции расширений. Четвертый метод применяется в сочетании с первым или четвертым методами интеграции расширений.

Четвертый метод исполнения кода расширений является наиболее сложным с точки зрения его реализации, поскольку требует наличия и интеграции дополнительных специализированных аппаратных возможностей у основной вычислительной машины. Но благодаря такому способу достигается наибольшая производительность при исполнении кода расширений (хотя этот эффект может несколько ослабляться накладными расходами на передачу данных сопроцессора). Например, аппаратная интерпретация расширений применяется в системе программирования EXO-CHI для архитектуры IA с технологией интеграции с различными акселераторами (о которой говорилось при описании четвертого метода интеграции расширений). В частности, в качестве средства программирования микропроцессора архитектуры IA, интегрированного с акселератором DPE, разработана интеграция языка C со специализированным языком DPL [9]. Язык DPL является предметно-ориентированным, он предназначен для описания параллельных вычислений на акселераторе процессов обработки сигналов DPE [10].

Пример предметно-ориентированного расширения языка общего назначения средствами таблично-ориентированного программирования

Рассмотрим предметно-ориентированное расширение языка общего назначения средствами решения задач с данными в табличной форме, настраиваемое на предметную область[3]. Оно имеет рабочую настройку на предметную область эфемеридной астрономии и применяется для автоматизации вычислений в этой области [15].

Возникновение задачи разработки этого расширения связано со следующими историческими предпосылками, обусловливающими актуальность этой задачи. Специализированная система ЭРА, включающая средства программирования на предметно-ориентированном языке СЛОН (слежение и обработка наблюдений), более 20 лет успешно применяется в Институте прикладной астрономии (ИПА) РАН для автоматизации вычислительных задач эфемеридной астрономии. Ее создание и развитие продолжалось с середины 80-х гг. и нашло широкое применение в практической работе института. Ключевой идеей, положенной в основу системы ЭРА, является предложение Г. А. Красинского об использовании таблиц и алгебры таблиц как основных элементов программирования для решения задач эфемеридной астрономии в форме так называемых табличных операторов [12]. Используемая в системе ЭРА методика применения табличного подхода к обработке данных получила в дальнейшем название таблично-ориентированного программирования. Другой примечательной особенностью системы ЭРА является возможность настроить ее на выбранную предметную область [13]. И, наконец, высокое качество вычислительных моделей, положенных в основу функционального предметного наполнения, его полнота и регулярное обновление позволили системе ЭРА стать безусловным лидером в сфере программного обеспечения эфемеридной астрономии.

В процессе многолетнего опыта эксплуатации системы ЭРА назрела потребность в расширении средств, предоставляемых специализированным языком СЛОН [12, 14], средствами, доступными в традиционном языке программирования (а именно, Object Pascal [16]), что, согласно исследованиям [17, p. 15], вполне типично для предметно-ориентированных языков вообще. Эта потребность явилась побудительным мотивом к разработке нового языка Дельта на основе двух языков — СЛОН и Object Pascal, с объединением их функциональных возможностей путем расширения языка Object Pascal табличными операторами языка СЛОН. Выбор языка Object Pascal в качестве прототипа требующихся средств программирования общего назначения обусловлен несколькими причинами, в частности, тем, что Object Pascal тоже применяется для разработки вспомогательного функционального наполнения

[3] Это расширение реализовано автором под руководством канд. физ.-мат. наук *В. И. Скрипниченко* [11] на базе специализированной системы ЭРА (эфемеридных расчетов в астрономии) [12–14].

системы ЭРА, а отдельные конструкции языков Object Pascal и СЛОН схожи [18].

Рассмотрим выбор методов, примененных в реализации данного расширения. Простейший способ интеграции кода расширений на основе языка СЛОН с помощью библиотеки специализированных функций API в данном случае является неприемлемым — он неэффективен с точки зрения удобства программирования и продуктивности разработки с помощью таких расширений. Данный тезис проиллюстрирован в примере 2 [1] БД. Чтобы добиться желаемой эффективности целевого инструмента программирования при разработке языка Дельта, требуется более высокий уровень абстракции, чем уровень API, — а именно такой, как в исходном предметно-ориентированном языке СЛОН. Поэтому при проектировании языка Дельта выбран метод интеграции расширений в основной язык программирования (Object Pascal) в виде новых языковых конструкций — табличных операторов языка СЛОН в исходном виде, т. е. метод 4. При этом новые конструкции явно обозначаются в тексте на языке Object Pascal с помощью маркирующих конструкций, что позволяет выделять эти расширения на этапе предварительной трансляции (препроцессирования).

Теперь рассмотрим выбор метода исполнения данного расширения. В системе ЭРА [12–14] для исполнения программ на языке СЛОН, состоящих из табличных операторов, применяется программная интерпретация с помощью специальной программы, названной *процессором языка СЛОН* (СЛОН-процессором). Эта программа выполняет трансляцию табличных операторов в промежуточное представление, а затем его интерпретирует.

В отличие от способа, принятого для языка СЛОН, в целях исполнения программного кода на языке Дельта было решено сначала транслировать табличные операторы в промежуточный код на языке Object Pascal, а затем компилировать и компоновать всю программу с помощью имеющихся инструментов в системе Borland Delphi. В результате такого двойного преобразования получается исполняемый код. Таким образом, в системе Дельта применяется «двухпроходная» трансляция. Этот подход позволяет разделить способы исполнения отдельных частей табличного оператора — табличного выражения и использованных в нем блоков действий, а также предметно-ориентированных функций. Благодаря этому в реализации языка Дельта удалось ограничиться интерпретацией лишь части табличных операторов — табличных выражений — конструкций, не имеющих прямых аналогов в языке реализации Object Pascal (на этапе предварительной трансляции они преобразуются в строковые параметры функции интерпретации согласно методу 3 [1] интеграции расширений). Остальная часть табличных операторов программы на языке Дельта (блоки действий, включая вызовы предметно-ориентированных функций и процедур), наряду с основным текстом программы на Object Pascal, компилируется в исполняемый код с помощью двухпроходной трансляции. На первой фазе трансляции выполняется преобразование в промежуточный код на Object Pascal с вызовами функций API интерпретатора (согласно методу 1 [1] интеграции расширений), а на второй — преобразование в целевой исполняемый код.

Таким образом, в результате анализа разнообразных методов расширения языков программирования сделан обоснованный выбор следующих методов, подходящих для реализации языка Дельта [15].

- Для интеграции расширений в БЯ используются новые языковые конструкции (метод 4 интеграции расширений) на уровне разработки исходного кода приложений Дельта и вызовы функций API реализации расширений со специальными строковыми параметрами (методы 1 и 3 [1]) на уровне представления промежуточного кода на Object Pascal.
- Для исполнения кода расширений используется сочетание программной интерпретации с предварительной трансляцией в код на БЯ программирования (методы 1 и 3 исполнения расширений).

Среди методов интеграции расширений метод расширения языков программирования новыми конструкциями является наилучшим с точки зрения качеств получаемых средств программирования, но в то же время и наиболее сложным в реализации по сравнению с применением других методов. С другой стороны, в различных случаях оптимальным может оказаться выбор любого из первых трех рассмотренных наиболее простых и часто встречающихся методов интеграции расширений. Из-за относительной простоты реализации данные методы обладают некоторыми общими недостатками, связанными с необходимостью «подстраивать» расширенные возможности под существующую систему программирования на БЯ, что в результате приводит к значительным ограничениям в плане удобства программирования и автоматического контроля со стороны инструментальных средств программирования. Вместе с тем их неоспоримыми преимуществами являются возможность использовать имеющийся инструментарий и минимальность дополнительной разработки для поддержки соответствующих расширений. Благодаря этому данные методы являются востребованными, а в определенных ситуациях их применение может оказаться наиболее удачным решением.

Выбор метода исполнения кода расширений в общем случае во многом определяется условия-

ми конкретного проекта: составом имеющихся инструментальных средств программирования на БЯ; требованиями к производительности исполнения кода на расширенном языке; возможностями целевой аппаратной платформы по исполнению специализированных функций; наличием временных и человеческих ресурсов, необходимых для доработки базовых инструментальных средств программирования.

Заключение

В приведенном обзоре систематизированы сведения о разновидностях расширений в современных языках программирования, о методах интеграции и исполнения кода расширений. Этот материал дает представление о существующих расширенных возможностях в известных языках программирования, позволяющее сориентироваться в их многообразии и выбрать самое подходящее средство для решения конкретных специализированных задач наиболее удобным и эффективным способом с использованием расширений языка программирования. Эти сведения будут также полезны при выборе оптимальных методов для реализации собственных расширений в случае возникновения такой потребности.

Литература

1. **Михеева В. Д.** Методы расширения языков программирования. Ч. 1 // Информационно-управляющие системы. 2010. № 4. С. 46–52.
2. **Using Inline Assembly in C/C++:** revision 14.10.2006 // The Code Project (a community of Software development and Design developers). http://www.codeproject.com/KB/cpp/edujini_inline_asm.aspx (дата обращения: 10.09.2009).
3. **International standard:** ISO/IEC 9899:1990, Information technology —Programming Languages — C; ISO/IEC JTC1/SC22/WG14 — The international standardization working group for C. http://www.open-std.org/JTC1/SC22/WG14/ (дата обращения: 10.09.2009).
4. **Wang P. et al.** Accelerator Exoskeleton — Tera-scale Computing // Intel Technology Journal. Aug. 2007. Vol. 11. Is. 03. P. 185–196.
5. **Box D., Hejlsberg A.** LINQ: .NET Language-Integrated Query // MSDN Library. Feb. 2007. http://msdn.microsoft.com/ru-ru/library/bb308959(en-us).aspx (дата обращения: 14.07.2009).
6. **Введение в LINQ** // Библиотека MSDN. Нояб. 2007. http://msdn.microsoft.com/ru-ru/library/bb397897.aspx (дата обращения: 14.07.2009).
7. **Hejlsberg A.** DEV223: LINQ Overview // Microsoft Tech Ed Developers, 2006.
8. **Calvert Ch.** LINQ and Deferred Execution // MSDN Blogs. Charlie Calvert's Community Blog. http://blogs.msdn.com/charlie/archive/2007/12/09/deferred-execution.aspx (дата обращения: 10.08.2009).
9. **Mironov S., Pavlov V., Yakoushkin S., Ivanov V.** DPL: Domain specific programming language and tools // XI Intern. Symp. on Problems of Redundancy in Information and Control Systems, St.-Petersburg, July 2007. С. 251–255.
10. **Hoffman J., Ilitzky D., Chun A., Chapyzhenka A.** Overview of the Scalable Communications Core // IEEE Computer Society Annual Symp. on VLSI (ISVLSI '07), 9–11 May 2007, Porto Alegre, Brazil. P. 3–8. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4208886 (дата обращения: 21.03.2010).
11. **Михеева В. Д., Новиков Ф. А., Скрипниченко В. И.** Дельта-язык и система программирования для решения прикладных задач с табличными данными // Научно-технические ведомости СПбГПУ. 2007. № 4. Т. 2. С. 57–60.
12. **Krasinsky G. A., Novikov F. A., Skripnichenko V. I.** Problem Oriented Language for Ephemeris Astronomy and its Realization in System ERA // Cel. Mech. 1989. Vol. 45. P. 219–229.
13. **Новиков Ф. А.** Архитектура системы «ЭРА» — табличный подход к обработке данных // Препринт ИПА АН СССР / Л., 1990. № 16. — 32 с.
14. **Krasinsky G. A., Vasiljev M. V.** ERA-7. Knowledge Base and Programming System for Dynamical Astronomy: manual / IAA RAS. St.-Petersburg, 2001. — 232 p.
15. **Михеева В. Д.** Разработка предметно-ориентированных приложений с помощью инструментальных средств Дельта // Сообщения ИПА РАН / СПб., 2008. № 179. — 32 с.
16. **Object Pascal Language Guide.** 2001 // Borland Software Corporation. 100 Enterprise Way, Scotts Valley, CA 95066-3249. http://www.borland.com, http://docs.embarcadero.com/products/rad_studio/cbuilder6/EN/CB6_ObjPascalLangGuide_EN.pdf (дата обращения: 21.03.2010).
17. **Horowitz E.** Fundamentals of Programming Languages. Second Edition. — USA: Computer Science Press, 1984. — 446 p.
18. **Михеева В. Д., Скрипниченко В. И.** Расширение языка Object Pascal (Delphi) таблично-ориентированными средствами решения задач эфемеридной астрономии // Сообщения ИПА РАН/ СПб., 2006. № 168. — 20 с.

УДК 681.3

ПРИМИТИВЫ КРИПТОСИСТЕМ С ОТКРЫТЫМ КЛЮЧОМ: КОНЕЧНЫЕ НЕКОММУТАТИВНЫЕ ГРУППЫ ЧЕТЫРЕХМЕРНЫХ ВЕКТОРОВ

***Д. Н. Молдовян*[1],**
младший научный сотрудник
Санкт-Петербургский институт информатики и автоматизации РАН

Для синтеза производительных алгоритмов распределения открытых ключей и открытого шифрования вводится новая вычислительно трудная задача над конечными некоммутативными группами. Предложен подход к построению некоммутативных групп четырехмерных векторов над простым полем и выводится формула для порядка этих групп. Описана схема согласования общего секретного ключа двух удаленных абонентов и алгоритм открытого шифрования на основе новой трудной задачи.

Ключевые слова *— криптография, криптосистемы с открытым ключом, протокол открытого согласования ключа, открытое шифрование, конечные группы, некоммутативные группы, группы векторов, трудная задача.*

Введение

В последние годы возрос интерес к использованию некоммутативных групп в качестве примитивов криптосистем с открытым ключом [1], связанный с ожиданием разработки действующего квантового компьютера, с применением которого задачи факторизации и дискретного логарифмирования в циклической группе будут решаться за полиномиальное время [2]. Последнее означает, что все наиболее широко применяемые алгоритмы открытого распределения ключей, открытого шифрования и электронной цифровой подписи (ЭЦП) станут небезопасными, поскольку их стойкость основана на сверхполиномиальной сложности указанных двух задач при их решении на имеющихся в настоящее время компьютерах. «Постквантовая» криптография связана с использованием новых трудных задач, сложность которых была бы сверхполиномиальна и в случае применения квантового вычислителя. В работах [3, 4] такие задачи сформулированы над бесконечными некоммутативными группами переплетения, а затем применены для построения алгоритмов открытого шифрования [4] и ЭЦП [5].

В данной статье формулируется новая вычислительно трудная задача над конечными некоммутативными группами, рассматривается построение на ее основе алгоритмов открытого согласования общего секретного ключа двух удаленных абонентов и открытого шифрования, описывается подход к заданию конечных некоммутативных групп четырехмерных векторов и выводится формула для вычисления порядка таких групп.

Криптосхемы с открытым ключом на основе конечных некоммутативных групп

Одним из способов синтеза криптосистем с открытым ключом на основе некоммутативных конечных групп является использование их подгрупп, обладающих достаточно большим простым порядком. Такие подгруппы являются циклическими, а групповая операция в них коммутативна. В этом случае синтез криптосхем, основанных на сложности задачи дискретного логарифмирования, аналогичен синтезу в случае коммутативных групп. Обоснованием данного подхода является ожидание, что сведение задачи дискретного логарифмирования в циклической подгруппе некоммутативной группы к задаче дискретного логарифмирования в конечном поле или конечном кольце многочленов будет невозможно. В частных случаях такое сведение возможно, поэтому требуется выполнить выбор подгрупп, об-

[1] Научный руководитель — доктор технических наук, профессор, заместитель директора по научной работе СПИИРАН *Б. В. Соколов*.

ладающих соответствующим значением простого порядка [6].

Более интересным является подход, использующий задачу вычисления элемента некоммутативной группы X и числа x (пара этих элементов служит секретным ключом) в уравнении $Y = X \circ G^x \circ X^{-1}$, где Y — элемент группы, используемый в качестве открытого ключа; G — элемент достаточно большого простого порядка q (элементы Y и G считаются известными); X — элемент, для которого выполняются неравенства $X \circ G \neq G \circ X$ и $Y \circ G \neq G \circ Y$ (при заданном числе x значение элемента X определяет значение Y). Операции умножения на взаимно обратные элементы X и X^{-1} реализуют операцию автоморфизма [7]. Решение указанного выше уравнения представляет собой самостоятельную трудную вычислительную задачу, отличную от задачи дискретного логарифмирования. При известном X можно вычислить $Y' = X^{-1} \circ Y \circ X$ или $G' = X \circ G \circ X^{-1}$, после чего число x можно найти из уравнения $Y' = G^x$ или $Y = G'^{\,x}$ соответственно, т. е. решая задачу дискретного логарифмирования. Однако значение X является неизвестным, поэтому задача дискретного логарифмирования в циклической подгруппе не стоит. Можно ожидать, что в конечной некоммутативной группе при известном x сравнительно легко вычислить неизвестный элемент X, однако одновременное нахождение X и x является сложной задачей, несмотря на то, что существует очень большое число различных решений (если X и x — некоторое решение, то для произвольного элемента группы θ, коммутирующего со всеми другими элементами, пара $(\theta X, x)$ также является решением).

Схема открытого согласования ключа на основе предлагаемой задачи описывается следующим образом. Пусть два удаленных абонента имеют открытые ключи $Y_1 = X_1 \circ G^{x_1} \circ X_1^{-1}$ и $Y_2 = X_2 \circ G^{x_2} \circ X_2^{-1}$, где (X_1, x_1) и (X_2, x_2) — личные секретные ключи первого и второго абонентов такие, что $X_1 \circ X_2 = X_2 \circ X_1$. После открытого обмена открытыми ключами первый абонент вычисляет общий секретный ключ по формуле $K_{12} = X_1 \circ Y_2^{x_1} \circ X_1^{-1}$, а второй — по формуле $K_{21} = X_2 \circ Y_1^{x_2} \circ X_2^{-1} = K_{12}$. Комбинируя аналогичным образом операцию автоморфизма и операцию возведения в большую дискретную степень, можно построить алгоритмы коммутативного и открытого шифрования. Техническим вопросом является согласование выбора секретных значений X_1 и X_2 из одной и той же коммутативной подгруппы, который решается заданием дополнительного известного элемента Q такого, что $Q \circ G \neq G \circ Q$, также обладающего большим простым порядком. В этом случае открытые ключи вычисляются по формулам

$$Y_1 = Q^{w_1} \circ G^{x_1} \circ Q^{-w_1} \text{ и } Y_2 = Q^{w_2} \circ G^{x_2} \circ Q^{-w_2},$$

где пары (w_1, x_1) и (w_2, x_2) являются личным секретным ключом первого и второго абонентов соответственно. В случае, когда порядки элементов Q и G равны большому простому числу q, имеющему размер $|q|$ бит, вычисление пары неизвестных (w, x) может быть выполнено методом, аналогичным методу больших и малых шагов [8], с трудоемкостью $2^{|q|}$ операций возведения в степень. Из этой оценки следует, что рассмотренная схема открытого согласования ключа имеет приемлемую для практического применения криптостойкость при $q \geq 2^{80}$.

Алгоритм открытого шифрования можно построить по следующей схеме. Пусть некоторый пользователь желает послать секретное сообщение M владельцу открытого ключа $Y = Q^w \circ G^x \circ Q^{-w}$, где (w, x) — личный секретный ключ получателя сообщения. Отправитель сообщения формирует разовый личный секретный ключ в виде пары чисел (u, v), вычисляет разовый открытый ключ $R = Q^u \circ G^v \circ Q^{-u}$, вычисляет разовый общий секретный ключ $K = Q^u \circ Y^v \circ Q^{-u}$, по ключу K зашифровывает сообщение и отправляет разовый открытый ключ и зашифрованное сообщение получателю. Получатель по разовому открытому ключу R вычисляет разовый общий секретный ключ $K = Q^w \circ R^x \circ Q^{-w}$, затем по ключу K расшифровывает сообщение M. Пусть в данной схеме используется алгоритм шифрования, представленный функцией шифрования E_K, управляемой секретным ключом в виде элемента некоммутативной группы K. Тогда описанная выше схема реализуется следующим алгоритмом.

1. Отправитель генерирует пару случайных чисел (u, v), вычисляет элементы $R = Q^u \circ G^v \circ Q^{-u}$ и $K = Q^u \circ Y^v \circ Q^{-u}$ некоммутативной группы, где Y — открытый ключ получателя.

2. Используя значение K в качестве ключа шифрования, отправитель зашифровывает сообщение M: $C = E_K(M)$.

3. Отправитель направляет получателю криптограмму C и значение R.

4. Получатель вычисляет значение $K' = Q^w \circ R^x \circ Q^{-w} = K$, где (w, x) — личный секретный ключ получателя, и расшифровывает криптограмму C: $M = D_K(C)$, где D_K — функция расшифрования, обратная E_K.

В описанных выше криптосхемах используется отображение циклической подгруппы Γ_G, порождаемой элементом G в некоторую другую (скрытую) циклическую подгруппу того же простого порядка q, задаваемого отображением $\varphi_X(G^i) = X \circ G^i \circ X^{-1}$, где $i = 1, 2, \ldots, q$, которое является автоморфизмом рассматриваемой конечной некоммутативной группы. При этом элемент X за-

дается в виде $X = Q^w$. Возникает вопрос о различии циклических подгрупп, в которые отображается подгруппа Γ_G при различных значениях $1 \le w \le q'$, где q' — порядок элемента Q. Ответ на этот вопрос дает следующая теорема.

Теорема 1. Пусть G и Q — элементы некоммутативной группы, имеющие простые порядки q и q' соответственно, такие, что $Q \circ G \ne G \circ Q$ и $Z \circ G \ne G \circ Z$, где $Z = Q \circ G \circ Q^{-1}$. Тогда все элементы $Z_{ij} = Q^j \circ G^i \circ Q^{-j}$, где $i = 1, 2, ..., q - 1$ и $j = 1, 2, ..., q'$, попарно различны.

Доказательство: Очевидно, что при фиксированном j элементы $Z_{ij} = Q^j \circ G^i \circ Q^{-j}$, где $i = 1, 2, ..., q$, образуют циклическую подгруппу порядка q. Условие $Z \circ G \ne G \circ Z$ означает, что элемент Z не принадлежит подгруппе Γ_G, порождаемой степенями элемента G (при предположении противного легко устанавливается противоречие). Пусть при некоторых i, $i' \ne i$, j и $j' \ne j$ (для определенности положим $j' > j$) элементы $Z_j = Q^j \circ G^i \circ Q^{-j}$ и $Z_{j'} = Q^{j'} \circ G^{i'} \circ Q^{-j'}$ равны, т. е. $Q^j \circ G^i \circ Q^{-j} = Q^{j'} \circ G^{i'} \circ Q^{-j'}$. Умножая обе части последнего выражения справа на Q^j и слева на Q^{-j}, получаем $G^i = Q^{j'-j} \circ G^{i'} \circ Q^{-j'+j}$. Так как Γ_G есть подгруппа простого порядка q, то любой неединичный элемент подгруппы Γ_G является порождающим, т. е. при $1 \le i' \le q - 1$ элемент $P = G^{i'}$ является порождающим, а значит степени P^z ($z = 1, 2, ..., q$) пробегают все элементы подгруппы Γ_G. При этом для всех z имеет место соотношение

$$(G^i)^z = (Q^{j'-j} \circ P \circ Q^{-j'+j})^z = Q^{j'-j} \circ P^z \circ Q^{-j'+j} \in \Gamma_G,$$

т. е. отображение $\varphi_{Q^{j'-j}}(P^z) = Q^{j'-j} \circ P^z \circ Q^{-j'+j}$ переводит любой элемент подгруппы Γ_G в некоторый элемент этой же подгруппы. Это означает, что отображение $\varphi_{Q^{j'-j}}(\Gamma_G)$ отображает подгруппу Γ_G в себя. Рассмотрим отображение $\varphi_Q(\Gamma_G)$. Поскольку порядок элемента Q есть простое число q', то для $j' \ne j$ существует некоторое число $u = (j'-j)^{-1} \bmod q'$, для которого имеют место соотношения

$$Q = (Q^{j'-j})^u$$

и

$$\varphi_Q(\Gamma_G) = \varphi_{(Q^{j'-j})^u}(\Gamma_G) = \\ = \varphi_{Q^{j'-j}}(\varphi_{Q^{j'-j}}(...\varphi_{Q^{j'-j}}(\Gamma_G)...)),$$

где в правой части последнего выражения выполняется u последовательных отображений $\varphi_{Q^{j'-j}}(\Gamma_G)$. Так как $\varphi_{Q^{j'-j}}(\Gamma_G)$ — это отображение подгруппы Γ_G в себя, то u таких отображений также переводит подгруппу Γ_G в себя, т. е. $Z = \varphi_Q(G) = Q \circ G \circ Q^{-1} \in \Gamma_G$. Из сделанного предположения вытекает, что элемент Z принадлежит циклической подгруппе, порождаемой элементом G, следовательно, для элементов Z и G должно выполняться свойство коммутативности, т. е. $Z \circ G = G \circ Z$, однако это противоречит условию $Z \circ G \ne G \circ Z$ теоремы. Полученное противоречие доказывает теорему.

В соответствии с теоремой 1 число различных неединичных элементов Z_{ij} составляет $(q - 1)q'$, и они образуют вместе с единичным элементом N подгрупп простого порядка q, где $N = q'$, причем каждый неединичный элемент принадлежит только одной из этих подгрупп.

Построение конечных некоммутативных групп четырехмерных векторов

Рассмотрим множество векторов вида $(a, b, c, d) = a\mathbf{e} + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, где $\mathbf{e}$, $\mathbf{i}$, $\mathbf{j}$ и $\mathbf{k}$ — формальные базисные векторы; a, b, c и d – целые числа, называемые координатами и принадлежащие конечному простому полю $GF(p)$, где p — простое число. Выражения $a\mathbf{e}$, $b\mathbf{i}$, $c\mathbf{j}$ и $d\mathbf{k}$ обозначают векторы $(a, 0, 0, 0)$, $(0, b, 0, 0)$, $(0, 0, c, 0)$ и $(0, 0, 0, d)$ соответственно и называются компонентами вектора (a, b, c, d). Определим операцию сложения векторов как сложение одноименных координат: $(a, b, c, d) + (x, y, z, w) = (a + x, b + y, c + z, d + w)$, где знак «+» применен для обозначения двух разных операций — сложения элементов поля $GF(p)$ и сложения векторов, что не вносит неопределенности ввиду очевидности интерпретации знака в каждом случае его применения. Операцию умножения векторов $a\mathbf{e} + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$ и $x\mathbf{e} + y\mathbf{i} + z\mathbf{j} + w\mathbf{k}$ определим по правилу «умножения многочленов»:

$$(a\mathbf{e} + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}) \circ (x\mathbf{e} + y\mathbf{i} + z\mathbf{j} + w\mathbf{k}) = \\ = ax\mathbf{e} \circ \mathbf{e} + ay\mathbf{e} \circ \mathbf{i} + az\mathbf{e} \circ \mathbf{j} + aw\mathbf{e} \circ \mathbf{k} + bx\mathbf{i} \circ \mathbf{e} + \\ + by\mathbf{i} \circ \mathbf{i} + bz\mathbf{i} \circ \mathbf{j} + bw\mathbf{i} \circ \mathbf{k} + cx\mathbf{j} \circ \mathbf{e} + cy\mathbf{j} \circ \mathbf{i} + \\ + cz\mathbf{j} \circ \mathbf{j} + cw\mathbf{j} \circ \mathbf{k} + dx\mathbf{k} \circ \mathbf{e} + dy\mathbf{k} \circ \mathbf{i} + \\ + dz\mathbf{k} \circ \mathbf{j} + dw\mathbf{k} \circ \mathbf{k},$$

где координаты вектора умножаются как элементы поля $GF(p)$, а операция $\circ$ имеет более высокий приоритет по сравнению со сложением, а произведения всевозможных пар базисных векторов заменяются базисным вектором или однокомпонентным вектором в соответствии с правилом умножения, задаваемым табл. 1, в которой пара-

■ **Таблица 1.** *Правила умножения четырехмерных базисных векторов ($\varepsilon < p$)*

Базисный вектор	Базисный вектор			
	e	**i**	**j**	**k**
e	**e**	**i**	**j**	**k**
i	**i**	$-\varepsilon\mathbf{e}$	$\varepsilon\mathbf{k}$	$-\mathbf{j}$
j	**j**	$-\varepsilon\mathbf{k}$	$-\varepsilon\mathbf{e}$	**i**
k	**k**	**j**	$-\mathbf{i}$	$-\mathbf{e}$

метр $\varepsilon \in GF(p)$ называется структурным коэффициентом. Различным значениям ε соответствуют формально различные операции умножения векторов. Определенная таким способом операция умножение векторов (a, b, c, d) и (x, y, z, w) выполняется по правилу

$$(a, b, c, d) \circ (x, y, z, w) = \\ = (ax - \varepsilon by - \varepsilon cz - dw)\mathbf{e} + (bx + ay - dz + cw)\mathbf{i} + \\ + (cx + dy + az - bw)\mathbf{j} + (dx - \varepsilon cy + \varepsilon bz + aw)\mathbf{k}.$$

Легко проверить, что определенная операция умножения обладает свойством ассоциативности и в общем случае является некоммутативной. Нейтральным элементом по умножению является вектор $E = (1, 0, 0, 0)$.

Множество всех векторов $\{A\}$ такое, что каждому вектору A может быть сопоставлен обратный вектор A^{-1}, для которого выполняется соотношение $AA^{-1} = E$, образует конечную группу. Для вычисления порядка Ω построенной некоммутативной группы рассмотрим решение уравнения вида $AX = E$, которое можно представить следующим образом:

$$(a\mathbf{e} + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}) \circ (x\mathbf{e} + y\mathbf{i} + z\mathbf{j} + w\mathbf{k}) = \\ = (ax - \varepsilon by - \varepsilon cz - dw)\mathbf{e} + \\ + (bx + ay - dz + cw)\mathbf{i} + (cx + dy + az - bw)\mathbf{j} + \\ + (dx - \varepsilon cy + \varepsilon bz + aw)\mathbf{k} = 1\mathbf{e} + 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}.$$

Из последней записи вытекает, что для определения обратных значений следует решать следующую систему из четырех линейных сравнений с четырьмя неизвестными:

$$\begin{cases} ax - \varepsilon by - \varepsilon cz - dw \equiv 1 \bmod p \\ bx + ay - dz + cw \equiv 0 \bmod p \\ cx + dy + az - bw \equiv 0 \bmod p \\ dx - \varepsilon cy + \varepsilon bz + aw \equiv 0 \bmod p \end{cases}. \qquad (1)$$

Если главный определитель $\Delta(A)$ системы (1) не равен нулю, то существует решение, которое дает значение координат вектора, являющегося обратным к вектору $A = (a, b, c, d)$. Если $\Delta(A) = 0$, то вектор A необратим. Значение Ω определим как число всех четырехмерных векторов, равное p^4, за вычетом числа необратимых векторов. Запишем значение определителя $\Delta(A)$:

$$\Delta(A) = \begin{vmatrix} a & -\varepsilon b & -\varepsilon c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -\varepsilon c & \varepsilon b & a \end{vmatrix} = a \begin{vmatrix} a & -d & c \\ d & a & -b \\ -\varepsilon c & \varepsilon b & a \end{vmatrix} + \\ + \varepsilon b \begin{vmatrix} b & -d & c \\ c & a & -b \\ d & \varepsilon b & a \end{vmatrix} - \varepsilon c \begin{vmatrix} b & a & c \\ c & d & -b \\ d & -\varepsilon c & a \end{vmatrix} + d \begin{vmatrix} b & a & -d \\ c & d & a \\ d & -\varepsilon c & \varepsilon b \end{vmatrix}.$$

Четыре слагаемых в правой части последнего выражения содержат одинаковый множитель:

$$a \begin{vmatrix} a & -d & c \\ d & a & -b \\ -\varepsilon c & \varepsilon b & a \end{vmatrix} = a(a(a^2 + \varepsilon b^2) + d(ad - \varepsilon bc) + \\ + c(\varepsilon bd + \varepsilon ac)) = a^2(a^2 + \varepsilon b^2 + \varepsilon c^2 + d^2);$$

$$\varepsilon b \begin{vmatrix} b & -d & c \\ c & a & -b \\ d & \varepsilon b & a \end{vmatrix} = \varepsilon b(b(a^2 + \varepsilon b^2) + d(ac + bd) + \\ + c(\varepsilon bc - ad)) = \varepsilon b^2(a^2 + \varepsilon b^2 + \varepsilon c^2 + d^2);$$

$$-\varepsilon c \begin{vmatrix} b & a & c \\ c & d & -b \\ d & -\varepsilon c & a \end{vmatrix} = -\varepsilon c(b(ad - \varepsilon bc) - a(ac + bd) + \\ + c(-\varepsilon c^2 - d^2)) = \varepsilon c^2(a^2 + \varepsilon b^2 + \varepsilon c^2 + d^2);$$

$$d \begin{vmatrix} b & a & -d \\ c & d & a \\ d & -\varepsilon c & \varepsilon b \end{vmatrix} = d(b(\varepsilon bd + \varepsilon ac) - a(\varepsilon bc + ad) - \\ - d(-\varepsilon c^2 - d^2)) = d^2(a^2 + \varepsilon b^2 + \varepsilon c^2 + d^2).$$

Складывая правые части последних четырех выражений, получаем

$$\Delta(A) = (a^2 + \varepsilon b^2 + \varepsilon c^2 + d^2)^2,$$

откуда следует, что число необратимых векторов равно числу решений сравнения

$$a^2 + \varepsilon b^2 + \varepsilon c^2 + d^2 \equiv 0 \bmod p \qquad (2)$$

относительно неизвестных (a, b, c, d). Рассмотрим следующее утверждение.

Утверждение 1. Пусть простое число p представляется в виде $p = 4k + 1$ **при некотором натуральном** $k \geq 1$. Тогда число решений сравнения (2) равно $p^3 + p^2 - p$ при произвольных значениях $1 \leq \varepsilon \leq p - 1$.

Доказательство: Для рассматриваемых значений простого числа p число -1 является квадратичным вычетом. Действительно, в соответствии с критерием Эйлера имеем $(-1)^{(p-1)/2} = (-1)^{2k} \equiv 1 \bmod p$. Следовательно, существует квадратный корень из -1, т. е. для некоторого целого числа $\lambda < p - 1$ имеем $\lambda^2 \equiv -1 \bmod p$. Перепишем сравнение (2) в тождественном виде:

$$a^2 - (\lambda d)^2 \equiv -\varepsilon(b^2 + c^2) \bmod p;$$

$$(a + \lambda d)(a - \lambda d) \equiv -\varepsilon(b^2 + c^2) \bmod p;$$

$$\alpha\delta \equiv -\varepsilon(b^2 + c^2) \bmod p, \qquad (3)$$

где $\alpha \equiv a + \lambda d \bmod p$; $\delta \equiv a - \lambda d \bmod p$; $0 \le \alpha \le p-1$ и $0 \le \delta \le p-1$. **Легко видеть, что между множеством пар** (α, δ) и множеством пар (a, d) существует взаимно однозначное соответствие, следовательно, число решений сравнения (3) относительно неизвестных (α, b, c, δ) равно числу решений сравнения (2) относительно неизвестных (a, b, c, d). Подсчитаем число решений сравнения (3). Если $\tau = b^2 + c^2 \ne 0 \bmod p$, то $\alpha \ne 0$ и $\delta \ne 0$. **Сравнение** $b^2 + c^2 \equiv 0 \bmod p$ имеет $2p-1$ **решений относительно неизвестных** b и c: одно решение имеет вид $(b, c) = (0, 0)$ и $2(p-1)$ решений имеют вид $(b, c) = (\pm\lambda c, c)$, где $1 \le c \le p-1$. Для каждого значения $\tau \ne 0$ (это имеет место для $N_{bc} = p^2 - 2p + 1$ различных пар значений b и c) и каждого значения $\alpha \ne 0$ (число различных значений α равно $N_\alpha = p-1$) **существует** единственное δ, удовлетворяющее сравнению (3), т. е. случаю $\tau \ne 0$ соответствуют $N_{\tau \ne 0} = N_{bc}N_\alpha = (p-1)^3$ различных решений сравнения (3). Каждому из $2p-1$ вариантов пар значений b и c, при которых имеет место случай $\tau = 0$, **соответствуют** $p-1$ различных решений, образуемых парами значений $\alpha = 0$ и $\delta \ne 0$ плюс $p-1$ **различных решений**, образуемых парами значений $\alpha \ne 0$ и $\delta = 0$ плюс решение $(\alpha, \delta) = (0, 0)$, т. е. значению $\tau = 0$ соответствуют $N_{\tau = 0} = (2p-1)^2$ различных решений сравнения (3). Таким образом, число различных решений сравнения (3), а значит и сравнения (2), равно

$$N_{\tau \ne 0} + N_{\tau = 0} = (p-1)^3 + (2p-1)^2 = p^3 + p^2 - p,$$

что и требовалось доказать.

Утверждение 2. Пусть простое число p представляется в виде $p = 4k + 1$ **при некотором натуральном** $k \ge 1$. Тогда порядок некоммутативной группы четырехмерных векторов, групповая операция которой задана по табл. 1, равен $\Omega = p(p-1) \times (p^2-1)$.

Доказательство: Число всех различных четырехмерных векторов над простым полем характеристики p равно $N = p^4$. В соответствии с утверждением 1 число необратимых векторов равно $N' = p^3 + p^2 - p$. Порядок группы равен числу обратимых векторов $\Omega = N - N' = p^4 - p^3 - p^2 + p = p(p^2(p-1) - (p-1)) = p(p-1)(p^2-1)$, **что требовалось доказать.**

Утверждение 3. При $\varepsilon = 0$ порядок некоммутативной группы четырехмерных векторов равен $\Omega = p^2(p^2-1)$, если простое число p представляется в виде $p = 4k + 3$, или $\Omega = p^2(p-1)^2$, если простое число p представляется в виде $p = 4k + 1$.

Доказательство: При $\varepsilon = 0$ определитель системы сравнений (1), из которой вычисляются координаты обратного вектора, равен $\Delta(A) \equiv a^2 + d^2 \bmod p$. Для значений простого числа p, представимых в виде $p = 4k + 3$, число -1 является квадратичным невычетом. Действительно, в соответствии с критерием Эйлера имеем $(-1)^{(p-1)/2} = (-1)^{2k+1} \equiv -1 \bmod p$. Следовательно, не существует квадратный корень из -1, поэтому сравнение $a^2 + d^2 \equiv 0 \bmod p$ имеет решение только при $(a, d) = (0, 0)$, т. е. все векторы вида $(0, b, c, 0)$ являются необратимыми. Число таких векторов равно p^2. В рассмотренном случае порядок группы векторов равен $\Omega = p^4 - p^2 = p^2(p^2-1)$. **Для значений простого числа** p, представимых в виде $p = 4k + 1$, число -1 является квадратичным вычетом, поэтому сравнение $a^2 + d^2 \equiv 0 \bmod p$ имеет $2p-1$ различных решений (см. доказательство утверждения 1), которым соответствуют необратимые векторы вида $(\pm(-d)^{1/2}, b, c, d)$, где b и c — произвольные значения. Число необратимых векторов равно $p^2(2p-1)$. В этом случае порядок группы векторов равен $\Omega = p^4 - p^2(2p-1) = p^2(p-1)^2$. Утверждение 3 доказано.

Утверждение 4. Пусть простое число p представляется в виде $p = 4k + 3$ **при некотором натуральном** $k \ge 1$. **Тогда если структурный коэффициент** ε является квадратичным невычетом, то порядок некоммутативной группы четырехмерных векторов, групповая операция которой задана по табл. 1, равен $\Omega = p(p-1)(p^2-1)$.

Доказательство: Для рассматриваемых значений простого числа p число -1 является квадратичным невычетом. Действительно, в соответствии с критерием Эйлера имеем $(-1)^{(p-1)/2} = (-1)^{2k+1} \equiv -1 \bmod p$. Поскольку ε – квадратичный невычет, то $\varepsilon^{(p-1)/2} \equiv -1 \bmod p$ и $(-\varepsilon)^{(p-1)/2} \equiv 1 \bmod p$. Следовательно, число $-\varepsilon$ является квадратичным вычетом и существует квадратный корень из $-\varepsilon$, т. е. для некоторого целого числа $\lambda < p-1$ имеем $\lambda^2 \equiv -\varepsilon \bmod p$. Перепишем сравнение (2) в тождественном виде:

$$a^2 - (\lambda b)^2 \equiv -(d^2 - (\lambda c)^2) \bmod p;$$

$$(a + \lambda b)(a - \lambda b) \equiv -(d^2 - (\lambda c)^2) \bmod p;$$

$$\alpha\delta \equiv -(d^2 - (\lambda c)^2) \bmod p, \qquad (4)$$

где $\alpha \equiv a + \lambda b \bmod p$; $\delta \equiv a - \lambda b \bmod p$; $0 \le \alpha \le p-1$ и $0 \le \delta \le p-1$. **Подсчитаем число решений сравнения** (4) относительно неизвестных (a, b, c, d) аналогично тому, как это сделано в доказательстве утверждения 1. Если $\tau = d^2 - (\lambda c)^2 \ne 0 \bmod p$, то $\alpha \ne 0$ и $\delta \ne 0$. Сравнение $d^2 - (\lambda c)^2 \equiv 0 \bmod p$ имеет $2p-1$ решений относительно неизвестных c и d: одно решение имеет вид $(c, d) = (0, 0)$ и $2(p-1)$ **решений** имеют вид $(c, d) = (c, \pm\lambda c)$, где $1 \le c \le p-1$. Для каждого значения $\tau \ne 0$ (это имеет место для $N_{bc} = p^2 - 2p + 1$ различных пар значений c и d) и $\alpha \ne 0$ ($N_\alpha = p-1$ различных значений α) суще-

ствует единственное δ, удовлетворяющее сравнению (4), т. е. случаю $\tau \neq 0$ соответствуют $N_{\tau \neq 0} = N_{bc}N_{\alpha} = (p-1)^3$ различных решений сравнения (4). Каждому из $2p-1$ вариантов пар значений c и d, при которых имеет место случай $\tau = 0$, соответствуют $p-1$ различных решений, образуемых парой значений $\alpha = 0$ и $\delta \neq 0$ плюс $p-1$ различных решений, образуемых парой значений $\alpha \neq 0$ и $\delta = 0$ плюс решение $(\alpha, \delta) = (0, 0)$, т. е. случаю $\tau = 0$ соответствуют $N_{\tau = 0} = (2p-1)^2$ различных решений сравнения (4). Таким образом, число различных решений сравнения (4), а значит и сравнения (2), равно

$$N' = N_{\tau \neq 0} + N_{\tau = 0} = \\ = (p-1)^3 + (2p-1)^2 = p^3 + p^2 - p.$$

Число всех различных четырехмерных векторов равно $N = p^4$. Порядок Ω группы равен числу обратимых векторов, следовательно:

$$\Omega = N - N' = p^4 - p^3 - p^2 + p = p(p-1)(p^2-1),$$

что и требовалось доказать.

Экспериментальные результаты

Доказанные выше утверждения о порядке конечной некоммутативной группы четырехмерных векторов для различных значений p и ε были проверены экспериментально с помощью компьютерной программы, реализующей алгоритм вычисления строения группы (под строением группы понимается таблица, показывающая число векторов для каждого возможного значения их порядка). Типичные результаты, полученные в ходе вычислительного эксперимента, который подтвердил доказанные в предыдущем разделе утверждения, показывает табл. 2. Эксперимент также показал, что в утверждении 4 требование того, что структурный коэффициент является квадратичным невычетом, может быть удалено, т. е. во всех экспериментах порядок конечной некоммутативной группы четырехмерных векторов оказался равным $\Omega = p(p-1)(p^2-1)$ независимо от структуры простого числа p и значения коэффициента $\varepsilon \neq 0$.

Рассмотрим пример выбора простых чисел p, q, q' и векторов G и Q, удовлетворяющих условиям $Q \circ G \neq G \circ Q$ и $Z \circ G \neq G \circ Z$, где $Z = Q \circ G \circ Q^{-1}$, и $q = q'$, $|q| \approx |p|$, где $|q|$ ($|p|$) — длина двоичной записи числа q (p):

$$p = 751788397;\ q = q' = (p+1)/2 = \\ = 375894199;\ \varepsilon = 1;$$

$$G = (493205368, 605223810, \\ 704049712, 215749841);$$

■ **Таблица 2.** *Строение частных вариантов конечных групп четырехмерных векторов (N_ω — число элементов порядка ω)*

$p = 7; \varepsilon = 1$		$p = 11; \varepsilon = 10$		$p = 11; \varepsilon = 0$		$p = 13; \varepsilon = 0$	
ω	N_ω	ω	N_ω	ω	N_ω	ω	N_ω
2	57	2	133	2	1	2	339
3	170	3	110	3	242	3	1016
4	42	4	110	4	242	4	1692
6	618	5	1324	5	4	6	3720
7	48	6	110	6	242	12	15552
8	84	8	220	8	484	13	168
12	84	10	4492	10	4	26	168
14	48	11	120	11	120	39	336
16	168	12	220	12	484	52	336
21	96	15	440	15	968	78	336
24	168	20	440	20	968	156	672
42	96	22	120	22	120	–	–
48	336	24	440	24	968	–	–
–	–	30	440	30	968	–	–
–	–	40	880	40	1936	–	–
–	–	55	480	55	480	–	–
–	–	60	880	60	1936	–	–
–	–	110	480	110	480	–	–
–	–	120	1760	120	3872	–	–
$1+\sum_\omega N_\omega =$	2016	–	13200	–	14520	–	24336
$\Omega = p(p-1)\times$ $\times(p^2-1)$		$\Omega = p(p-1)\times$ $\times(p^2-1)$		$\Omega = p^2\times$ $\times(p^2-1)$		$\Omega = p^2\times$ $\times(p-1)^2$	

$$Q = (204543067, 267966222, \\ 209297175, 161608828);$$

$$Q^{-1} = (204543067, 483822175, \\ 542491222, 590179569);$$

$$Z = (493205368, 638573510, \\ 56561748, 103277561);$$

$$Q \circ G = (478445912, 349091248, \\ 194139031, 297937680);$$

$$G \circ Q = (478445912, 529600113, \\ 62144304, 36127512);$$

$$Z \circ G = (325816345; 478721415; \\ 216264816; 409136505);$$

$G \circ Z =$ (325816345; 196930991; 521380191; 144664353).

Алгоритм вычисления секретного ключа

В схемах открытого согласования ключа и открытого шифрования, описанных в первом разделе, используется открытый ключ Y, который вычисляется по личному секретному ключу (w, x) по формуле $Y = Q^w \circ G^x \circ Q^{-w}$, где Q и G — известные элементы некоммутативной группы достаточно большого простого порядка q и q' соответственно. Сложность вычисления секретного ключа по открытому задает верхнюю границу стойкости предложенных криптосхем. Для вычисления секретного ключа по открытому может быть использован алгоритм, аналогичный алгоритму больших и малых шагов, применяемому для решения задачи дискретного логарифмирования [8]. Рассмотрим вычисление секретного ключа в случае $q = q'$ (алгоритм может быть легко адаптирован для $q \neq q'$).

1. Для всех значений $w = 1, 2, ..., q$ вычислить таблицу значений $T(w) = Q^{-w} \circ Y \circ Q^w$. (Трудоемкость этого шага составляет $2q$ операций умножения элементов группы.)

2. Упорядочить таблицу, вычисленную на шаге 1. (Трудоемкость этого шага составляет $q\log_2 q$ операций сравнения элементов группы.)

3. Установить счетчик $i = 1$ и значение вектора $V = (1, 0, ..., 0)$.

4. Вычислить вектор $V = V \circ G$.

5. По отсортированной таблице проверить, имеется ли в ней значение V. Если в таблице имеется значение $T(w') = V$, то вывести значение секретного ключа $(w, x) = (w', i)$ и СТОП, в противном случае перейти к шагу 6.

6. Если $i \neq q$, то прирастить значение счетчика $i \leftarrow i + 1$ и перейти к шагу 4, в противном случае — СТОП и вывести сообщение «условия некорректны». (Трудоемкость шагов 5 и 6 составляет не более q операций умножения и $q\log_2 q$ операций сравнения.)

Если условия корректны, т. е. решение задачи имеется, то приведенный алгоритм при некотором значении счетчика i' найдет значение вектора V в таблице, т. е. в этот момент будет выполняться соотношение $T(w') = V = G^{i'}$. Поскольку $T(w') = Q^{-w'} \circ Y \circ Q^{w'}$, то $Y = Q^{w'} \circ G^{i'} \circ Q^{-w'}$, т. е. при корректно заданных условиях алгоритм действительно найдет решение. Трудоемкость алгоритма S составляет $3q$ операций умножения плюс $2q\log_2 q$ операций сравнения, т. е. $S = O(q)$, где $O(q)$ — обозначение порядка. Если принять достаточной верхнюю границу стойкости, равную $O(2^{80})$ операций (как в случае 1024-битовой криптосистемы RSA), то в предложенных в первом разделе криптосхемах следует выбрать некоммутативную группу, для которой значение порядка элементов Q и G равно $q \geq 2^{80}$.

Заключение

В качестве примитива протоколов открытого согласования секретного ключа и открытого шифрования предложена новая вычислительно трудная задача над конечными некоммутативными группами, которую можно назвать задачей дискретного логарифмирования в скрытой циклической подгруппе. На основе данной задачи построена схема согласования по открытому каналу общего секретного ключа двух удаленных абонентов и алгоритм открытого шифрования. В основе этих построений лежит процедура, комбинирующая функцию, задающую автоморфизм конечной некоммутативной группы, и операцию возведения элемента достаточно большого простого порядка в большую целочисленную степень. Разработан алгоритм решения рассматриваемой задачи, из которого дана оценка стойкости предложенных криптосхем. Однако данная задача является новой, поэтому требуются другие независимые исследования для более полной оценки безопасности этих криптосхем. Доказанная теорема 1 имеет общее значение для конечных некоммутативных групп различных типов. Эта теорема отражает «локальное» строение некоммутативной группы, а именно строение в той ее части, которая используется в построении криптосхем на основе новой задачи.

Предложенные криптосхемы могут быть реализованы над построенными конечными некоммутативными группами четырехмерных векторов, над конечными некоммутативными группами векторов других размерностей [9] или над конечными группами невырожденных матриц [10]. Выбор конкретного варианта некоммутативной группы для построения криптографических алгоритмов и протоколов будет определяться вычислительной эффективностью алгоритмов и протоколов при заданном уровне стойкости. Сравнительный анализ быстродействия криптосхем на основе конечных некоммутативных групп различного типа представляет самостоятельный интерес.

Предложенная вычислительно трудная задача может быть применена и для построения криптосхем других типов, например для разработки быстродействующих алгоритмов коммутативного шифрования и протоколов аутентификации с нулевым разглашением. Эти вопросы также представляют интерес как тема отдельного исследования.

Работа поддержана грантом РФФИ № 08-07-00096-а.

Литература

1. **Anshel I., Anshel M., Goldfeld D.** An Algebraic Method for Public Key Cryptography // Mathematical Research Letters. 1999. Vol. 6. P. 287–291.
2. **Shor P. W.** Polynomial-time algorithms for prime factorization and discrete logarithms on quantum computer // SIAM Journal of Computing. 1997. Vol. 26. P. 1484–1509.
3. **Ko K. H. et al.** New Public-Key Cryptosystems Using Braid Groups // Advances in Cryptology — Crypto 2000: Lecture Notes in Computer Science. Springer-Verlag, 2000. Vol. 1880. P. 166–183.
4. **Lee E., Park J. H.** Cryptanalysis of the Public Key Encryption Based on Braid Groups // Advances in Cryptology — Eurocrypt 2003: Lecture Notes in Computer Science. Springer-Verlag, 2003. Vol. 2656. P. 477–489.
5. **Verma G. K.** A Proxy Blind Signature Scheme over Braid Groups // Int. Journal of Network Security. 2009. Vol. 9. N 3. P. 214–217.
6. **Молдовяну П. А., Дернова Е. С., Костина А. А., Молдовян Н. А.** Гомоморфизм конечных групп векторов малой размерности и синтез схем цифровой подписи // Информационно-управляющие системы. 2009. № 4. С. 26–32.
7. **Каргаполов М. И., Мерзляков Ю. И.** Основы теории групп. — М.: Физматлит, 1996. — 287 с.
8. **Menezes A. J., Van Oorschot P. C., Vanstone S. A.** Handbook of Applied Cryptography. — Boca Raton, FL: CRC Press, 1997. — 780 p.
9. **Молдовян Д. Н., Куприянов А. И., Костина А. А., Захаров Д. В.** Задание некоммутативных конечных групп векторов для синтеза алгоритмов цифровой подписи // Вопросы защиты информации. 2009. № 4. С. 13–18.
10. **Дернова Е. С., Костина А. А., Молдовяну П. А.** Конечные группы матриц как примитив алгоритмов цифровой подписи // Вопросы защиты информации. 2008. № 3(82). С. 8–12.

Новиков Ф. А., Иванов Д. Ю.
Моделирование на UML. Теория, практика, видеокурс. — СПб.: Профессиональная литература, Наука и Техника, 2010. — 640 с.: ил. + цв. вклейки (+2 DVD) ISBN 978-5-94387-610-3.

Книга содержит полное описание всех основных версий унифицированного языка моделирования UML и набор рекомендаций по применению языка для моделирования программных систем. При этом высокий уровень понимания авторами UML, умение его использовать вкупе с блестящими педагогическими навыками и хорошим, доступным языком позволяют сделать из учебника (которым книга, несомненно, является) нечто большее, чем просто учебник. Передаваемый опыт и идеи, которыми авторы щедро делятся на страницах книги, делают ее интересной как для читателя уже знакомого с UML, так и для читателя, которому просто интересно узнать, что такое UML и как его применять в своей практике.

В конце книги размещены сводные таблицы, толковый словарь и развитый предметный указатель, что позволяет использовать книгу в качестве справочника. На цветной вклейке дается графическая нотация-шпаргалка, представляющая собой квинтэссенцию нотации UML с необходимыми пояснениями. К книге прилагается видеокурс по UML на двух DVD.

Книга предназначена для практикующих разработчиков программного обеспечения, руководителей IT-проектов и их заказчиков, системных архитекторов, студентов высших и средних специальных учебных заведений, а также всех желающих освоить унифицированный язык моделирования UML или познакомиться с ним.

Книгу можно приобрести на официальном сайте данного издания: www.umlmanual.ru

УДК 519.248, 621.384.3

АНАЛИЗ ПРОБЛЕМЫ ОБНАРУЖЕНИЯ В ИНФРАКРАСНЫХ СИСТЕМАХ

М. О. Колбанев,
доктор техн. наук, профессор
В. А. Рогачев,
канд. техн. наук, доцент
Санкт-Петербургский государственный университет телекоммуникаций

Проблема обнаружения сигнала в общем режиме в инфракрасных системах формулируется как задача обнаружения двухпараметрического сигнала. Применение критерия Неймана — Пирсона и принципа инвариантности позволяет получить решение — модифицированную статистику Фишера. Сравнение с известными статистиками определяет диапазоны значений сигнала и уровней помех, при которых обеспечивается максимальная вероятность правильного обнаружения.

Ключевые слова *— обнаружение, инфракрасные системы, критерий Неймана — Пирсона, модифицированная статистика Фишера.*

Введение

В инфракрасных системах при обнаружении, как правило, реализуется контрастный метод обнаружения, обусловленный наличием помехи неизвестного уровня. При этом выполняется сравнение контраста, вычисляемого по некоторому алгоритму для сигнальной и помеховой выборок, с пороговым уровнем [1].

Выбор того или иного алгоритма обнаружения в существенной мере зависит от режима, в котором находится инфракрасная система [2]. Для режимов обнаружения: ограничение внутренним шумом (определяющим в системе является внутренний шум), обнаружение случайного сигнала (в системе выполняется обнаружение случайного сигнала) и режима ограничения фоном (определяющим в системе является фоновый шум) — получены оптимальные алгоритмы обнаружения, обеспечивающие максимальную вероятность правильного обнаружения при всех амплитудах сигнала и заданной вероятности ложной тревоги [3].

Гораздо менее определенной остается ситуация с оптимальным алгоритмом обнаружения для общего режима. В этом случае выходной сигнал фотоприемника представляет собой случайный сигнал с нормальным распределением, являющийся суммой сигнала объекта, внутренней и внешней помехи. Причем внутренняя помеха имеет математическое ожидание, равное темновому току, и дисперсию, определяемую внутренним шумом. Для внешней помехи математическое ожидание равно фоновому току, а дисперсия определяется фоновым шумом, при этом математическое ожидание и дисперсия, как правило, связаны друг с другом коэффициентом пропорциональности, зависящим от типа фотоприемника.

Появление обнаруживаемого объекта в поле зрения системы вызывает изменение как математического ожидания, так и дисперсии выходного сигнала фотоприемника [3].

В случае полностью известных параметров задача достаточно легко решается с помощью критерия Неймана — Пирсона вычислением отношения правдоподобия [4].

Однако в реальных условиях, как правило, определены только объемы сигнальной и помеховой выборок, а параметры помехи и сигнала объекта полностью неизвестны.

Проблеме обнаружения сигнала объекта в инфракрасных системах в условиях априорной неопределенности и посвящена данная работа.

Формулирование подходов к решению задачи обнаружения

Поскольку обнаруживаемый сигнал объекта имеет два полезных признака — математическое ожидание и дисперсию, то задача обнаружения может быть сформулирована по-разному.

Если дисперсия сигнала объекта — «сигнальные шумы» — не учитывается как признак, то задача обнаружения формулируется как задача сравнения математических ожиданий двух выборок при неравных дисперсиях. Такая задача эквивалентна проблеме Беренса — Фишера, не имеющей непрерывного решения [5, 6].

При учете как математического ожидания, так и дисперсии, определяемых сигналом объекта, задача формулируется как задача обнаружения двух независимых параметров одновременно. Такая задача эквивалентна обнаружению многомерного (двумерного) сигнала и не имеет решения для всех амплитуд полезного сигнала (равномерно наиболее мощного — РНМ-решения) [5].

При обнаружении сигнала объекта с учетом того, что дисперсия сигнала объекта пропорциональна его математическому ожиданию, задача формулируется как задача обнаружения одного параметра. В этом случае размерность статистики больше размерности параметра и, следовательно, статистика не полна [5]. Такая задача эквивалентна обнаружению при двух функционально связанных параметрах и не имеет решений неймановской структуры [5].

Применение критерия Неймана — Пирсона

Рассмотрим, что дает применение критерия Неймана — Пирсона для обнаружения сигнала в инфракрасной системе, находящейся в общем режиме.

Синтез оптимального правила обнаружения произведем на основе распределения выходного сигнала фотоприемника, которое имеет следующий вид:

— при отсутствии сигнала объекта:

$$H_0 : x \in N(d+b, \sigma^2 + ab), y \in N(d+b, \sigma^2 + ab);$$

— при наличии сигнала объекта:

$$H_1 : x \in N(d+b, \sigma^2 + ab), y \in N(d+b+s, \sigma^2 + ab + as),$$

где d — уровень темнового тока; b — уровень фонового тока; σ^2 — дисперсия внутреннего шума; ab — дисперсия фонового шума, as — дисперсия сигнала, a — коэффициент пропорциональности между током и шумами, зависящий от типа фотоприемника; s — математическое ожидание сигнала объекта.

Совместная плотность распределения элементов сигнальной и помеховой выборок для этого класса

$$p(x, y) = C(d+b, s, \sigma^2 + ab, as) \times$$
$$\times \exp(\theta_1 T_1 + \theta_2 T_2 + \theta_3 T_3 + \theta_4 T_4),$$

где C — постоянная.

Это экспоненциальное семейство с четырьмя параметрами

$$\theta_1 = -1/(2(\sigma^2 + ab)),\ \theta_2 = -1/(2(\sigma^2 + ab + as)),$$
$$\theta_3 = (d+b)/(2(\sigma^2 + ab)),$$
$$\theta_4 = (d+b+s)/(2(\sigma^2 + ab + as))$$

и четырьмя достаточными статистиками

$$T_1 = \sum_{i=1}^{M} x_i^2,\ T_2 = \sum_{j=1}^{N} y_j^2,\ T_3 = \sum_{i=1}^{M} x_i,\ T_4 = \sum_{j=1}^{N} y_j.$$

В данном случае существует четыре достаточные статистики: две — для математического ожидания и две — для дисперсии.

В качестве мешающих параметров с априорно неизвестными значениями в данном случае выступают математическое ожидание помеховой выборки, являющееся суммой темнового и фонового тока, и дисперсия помеховой выборки, представляющая собой сумму дисперсии внутреннего и фонового шумов. Для устранения влияния мешающих параметров применим принцип инвариантности [1, 5].

Применение этого принципа основано на использовании преобразований, инвариантных относительно проверяемых гипотез. Для нормального распределения такими преобразованиями являются преобразования из группы сдвигов и масштабов [5]. В результате приходим к решающей статистике следующего вида [3]:

$$r = \left(\sum_{j=1}^{N} (y_j - \bar{x})^2 / N \right) / \left(\sum_{i=1}^{M} (x_i - \bar{x})^2 / (M-1) \right),$$

где $\bar{x} = \sum_{i=1}^{M} x_i / M$ — среднее значение помеховой выборки.

Данная решающая статистика для проверки выдвигаемой гипотезы о наличии сигнала представляет отношение оценки дисперсии сигнальной выборки к оценке дисперсии помеховой выборки. В отличие от обычной статистики Фишера в числителе используется оценка математического ожидания помеховой выборки, в которой отсутствует сигнал объекта. В этом случае учитывается то, что при появлении полезного сигнала произойдет увеличение как математического ожидания, так и дисперсии.

Применение теории инвариантности дает возможность получить статистику для проверки выдвинутой гипотезы, однако судить о степени ее близости к статистике, обеспечивающей максимальную вероятность правильного обнаружения при всех значениях сигнала (РНМ), можно лишь в сравнении с решающими статистиками для сходных задач.

Вычисление распределения полученной решающей статистики — модифицированной статистики Фишера — приводит к нецентральному распределению Фишера, зависящему от двух параметров: отношения дисперсии сигнальной и помеховой выборок и отношения сигнал / шум. Плотность распределения имеет следующий вид [3]:

$$p(r) = ((M-1)\rho / N)^{(M-1)/2} \exp(-N\tau/2) \times$$
$$\times r^{N/2-1} \sum_{k=0}^{\infty} (1/k!)(N\tau/2)^k \times$$
$$\times (r + (M-1)\rho/N)^{-N/2-(M-1)/2-k} /$$
$$/ B((M-1)/2, N/2+k)$$

где $\rho = (\sigma^2 + ab + as + \sigma^2 / M + ab / M)/(\sigma^2 + ab)$ — параметр, равный отношению дисперсий сигнальной и помеховой выборок; $\tau = s^2 / (\sigma^2 + ab + as + \sigma^2 / M + ab / M)$ — параметр, связанный с отношением сигнал / шум; $B((M-1)/2, N/2+k)$ — бета-функция.

При отсутствии сигнала объекта распределение превращается в стандартное распределение Фишера и пороговый уровень определяется из центрального распределения Фишера [5]

$$\lambda_z = F^{-1}_{N,M-1}(1-\alpha),$$

где $F^{-1}_{N,M-1}(1-\alpha)$ — квантиль распределения Фишера уровня α и с N и M степенями свободы.

Сравнение вероятностей правильного обнаружения

Для определения качества полученной статистики сравним ее вероятность правильного обнаружения со следующими статистиками:

— статистика Гаусса (при всех известных параметрах) для определения максимально возможной вероятности правильного обнаружения;

— статистика Стьюдента, учитывающая только математическое ожидание сигнала объекта;

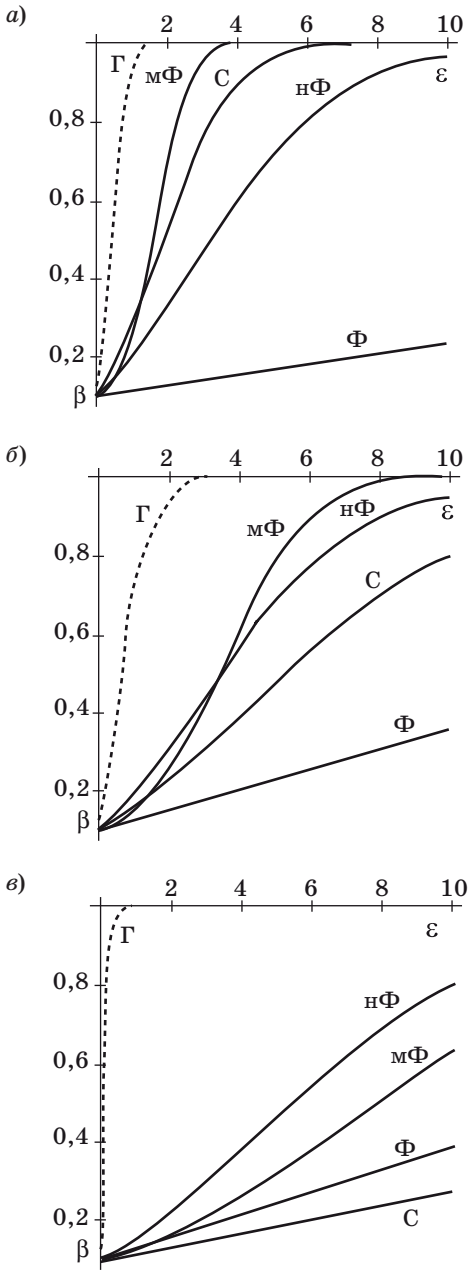
— статистика Фишера, учитывающая только дисперсию сигнала объекта;

— дважды нецентральная статистика Фишера, учитывающая математическое ожидание и дисперсию сигнала объекта, связанные друг с другом.

Вычисления вероятностей правильного обнаружения были произведены при изменении относительного уровня фонового тока в диапазоне от 0,1 до 10 и изменении нормированного коэффициента пропорциональности между средним и дисперсией в диапазоне от 0,1 до 10 для вероятности ложной тревоги 0,1; 0,01; 0,001 и объемов выборок от 10 до 100.

Как видно из приведенных графиков, при малом уровне дисперсии сигнала (рисунок, *а*) модифицированная статистика Фишера в основном обеспечивает превосходство по сравнению с остальными статистиками.

При средних уровнях дисперсии сигнала (рисунок, *б*) модифицированная статистика Фишера



■ *Вероятности правильного обнаружения* β *в зависимости от отношения сигнал / шум* ε *при a = 0,1 (а); 1 (б); 10 (в), относительном уровне фона 10, вероятности ложной тревоги 0,1 и объеме выборок, равном 10: Ф — статистика Фишера; мФ — модифицированная статистика Фишера; нФ — нецентральная статистика Фишера; С — статистика Стьюдента; Г — статистика Гаусса*

также в основном обеспечивает превосходство над остальными статистиками.

При большом уровне дисперсии сигнала (рисунок, *в*) хотя модифицированная статистика Фишера и проигрывает нецентральной статистике Фишера, по-прежнему обеспечивает уверенное превышение над статистиками Стьюдента и Фишера.

Однако при небольших уровнях дисперсии сигнала и малых отношениях сигнал / шум модифицированная статистика Фишера проигрывает статистике Стьюдента и нецентральной статистике Фишера.

Заключение

Проблема обнаружения сигнала в общем режиме в инфракрасных системах формулируется как задача обнаружения двухпараметрического сигнала на фоне внутренних и внешних помех.

Для решения задачи обнаружения двухпараметрического сигнала было сформулировано три подхода.

Показано, что эти подходы не обеспечивают максимальную вероятность правильного обнаружения для всех значений сигнала и уровней помех.

Метод, основанный на достаточных статистиках, критерии Неймана – Пирсона и принципе инвариантности, позволяет получить решение — модифицированную статистику Фишера.

Сравнение с такими статистиками, как статистика Стьюдента, статистика Фишера и нецентральная статистика Фишера позволяет определить диапазоны значений сигнала и уровней помех, при которых обеспечивается максимальная вероятность правильного обнаружения.

Литература

1. **Теория** обнаружения сигналов / Под ред. П. А. Бакута. — М.: Радио и связь, 1984. — 440 с.
2. **Хадсон Р.** Инфракрасные системы. — М.: Мир, 1972. — 536 с.
3. **Колбанев М. О., Рогачев В. А.** Оптимизация выделения полезного сигнала в многорежимных информационных системах // Вопросы радиоэлектроники. Сер. ЭВТ. 2010. Вып. 2. С. 92–103.
4. **Кендал М., Стьюарт А.** Статистические выводы и связи. — М.: Мир, 1966, — 900 с.
5. **Леман Э. Л.** Проверка статистических гипотез. — М.: Наука, 1978. — 452 с.
6. **Линник Ю. В.** Статистические задачи с мешающими параметрами. — М.: Наука, 1966. — 252 с.

УДК 007.5; 681.32

СИНТЕЗ БЫСТРОДЕЙСТВУЮЩИХ ИЗМЕРИТЕЛЬНО-УПРАВЛЯЮЩИХ СИСТЕМ НА БАЗЕ ПАРАМЕТРИЗОВАННЫХ МАРКОВСКИХ МОДЕЛЕЙ

А. П. Лапсарь,
канд. техн. наук, доцент
Ростовский военный институт Ракетных войск им. М. И. Неделина

Для синтеза измерительно-управляющих систем предложен численно-аналитический метод оценки стохастических характеристик марковской системы, описываемой эволюционными уравнениями, решения которых непрерывно зависят от вектора вещественных параметров, определяющих условия ее функционирования.

Ключевые слова *— марковская параметрическая система, эволюционные уравнения, метод редукции, интерполяция, стохастические характеристики.*

Введение

Управление сложными стохастическими системами различного назначения предполагает интеграцию в их состав измерительно-управляющих систем (ИУС), синтезируемых на основе адекватных математических моделей. Известно, что широкий класс стохастических марковских систем достаточно адекватно моделируется эволюционными уравнениями (ЭУ), например уравнением Фоккера—Планка—Колмогорова [1]. С использованием указанных уравнений эффективно решаются задачи анализа нелинейной статистической динамики различного рода динамических систем, а также синтеза алгоритмов оценки их состояния в различные моменты времени (например, вероятности безотказной работы, наработки на отказ, среднего времени достижения границ области допустимых значений и других стохастических характеристик (СХ)) [1, 2]. При этом набор оцениваемых СХ представляется в виде совокупности некоторых ограниченных непрерывных функционалов от плотности вероятности многомерного марковского процесса, удовлетворяющей используемому многомерному ЭУ.

Основной причиной, ограничивающей практическое применение марковской теории анализа и синтеза стохастических систем, является высокая сложность численного и, особенно, аналитического решения указанных ЭУ. При этом ситуация еще более усугубляется, если рассматриваются параметризованные ЭУ, т. е. заданные с точностью до вектора вещественных параметров, в качестве которых могут выступать начальные и граничные условия соответствующего уравнения, а также априори неизвестные константы, характеризующие условия функционирования стохастической системы.

Поскольку в ряде случаев, например при возникновении угрозы аварийной ситуации, предъявляются жесткие требования по быстродействию оценки СХ, то целесообразно этот процесс разбивать на два этапа. На первом этапе необходимо оценить локальные характеристики (коэффициенты сноса и диффузии) исследуемой системы и сформировать аналитико-параметрическое решение соответствующего параметризованного ЭУ. Здесь же по мере накопления информации об исследуемой системе могут уточняться параметры модели и полученное решение. На втором этапе, при выявлении признака аварийной ситуации в зависимости от конкретных значений вектора параметров, выдаваемых системой идентификации высшего уровня, вычисляются искомые СХ. Такое рассмотрение задачи оценки СХ позволяет вынести основные вычислительные, а следовательно, и временные затраты, на первый этап. В данной работе для синтеза ИУС высокого быстродействия дается обоснование метода оценки СХ марковской системы, моделируемой параметризованными ЭУ.

Постановка задачи в марковско-параметрическом виде

Рассмотрим в некотором нормированном пространстве W_0 параметризованное ЭУ в частных производных для r-мерного марковского процесса $x(t)$

$$\frac{\partial p(x,t)}{\partial t} = L^{(r)}_{\omega_0,\,t}\{p(x,t)\},$$
$$p(x,t)\in W_0,\ \ x\in X\subset R^r,\ \ t\in T, \qquad (1)$$

где $L^{(r)}_{\omega_0,\,t}$ — оператор параметризованного ЭУ (например, оператор Фоккера—Планка—Колмогорова), зависящий от вещественного векторного параметра ω_0.

Пусть искомое решение $\overset{*}{p}(x,t)$ уравнения (1) подчинено дополнительным условиям вида $\Phi_{\omega_j}\left(\overset{*}{p}(x,\omega,t)\right) = \varphi_{\omega_j}(S_i)$, $(x,\ t)\ \in\ S$, $i = 1, ..., L_0$, $j = 1, ..., L_1$, где Φ_{ω_j} — линейный непрерывный оператор, действующий в W_0 и зависящий от вещественного векторного параметра $\omega_j \in \Omega_j \subset R^{m_j}$; S_i — некоторое многообразие в области $X \times T$, число измерений которого меньше $r + 1$; $\varphi_{\omega_j}(S_i)$ — заданная функция, определенная на S_i и зависящая от ω_j.

Данную задачу можно представить в виде одного точного уравнения [3, 4]

$$p(x,\omega,t) - \lambda F(\omega)p(x,\omega,t) = f(x,\omega,t),$$
$$f(x,\omega,t)\in W, \qquad (2)$$

где $F(\omega)$ — линейный непрерывный оператор, действующий в нормированном пространстве $W \subset W_0$; λ — некоторая постоянная, не являющаяся характеристическим значением оператора $F(\omega)$ для

$$\omega = (\omega_0^T, \omega_1^T, \ldots, \omega_{L_1}^T)^T \in \Omega =$$
$$= \Omega_0 \times \Omega_1 \times \ldots \times \Omega_{L_1} \subset R^m = R^{m_0} \times R^{m_1} \times \ldots \times R^{m_{L_1}};$$

$f(x, \omega, t)$ — заданная функция из W.

На первом этапе функционирования ИУС может быть построено аналитико-параметрическое решение $\overset{*}{p}(x,\omega,t)$ уравнения (2), а на втором — совокупность искомых СХ $\overset{*}{Y}_i(\omega) = F_i\left[\overset{*}{p}(x,\omega,t)\right]$, $i = 1, ..., M_0$, где $F_i[\cdot]$ — ограниченные непрерывные функционалы.

Поскольку вместо $\overset{*}{p}(x,\omega,t)$ можно получить только приближенное решение $\tilde{p}(x, \omega, t)$ уравнения (2), то и вместо $\left\{\overset{*}{Y}_i(\omega)\right\}_{i=1}^{M_0}$ — лишь семейство $\left\{\tilde{Y}_i(\omega)\right\}_{i=1}^{M_0}$ приближенных СХ.

Требуется с учетом принятых моделей и ограничений разработать численно-аналитический метод оперативной оценки СХ марковской параметрической системы для последующего использования при синтезе ИУС высокого быстродействия.

Решение операторного уравнения

Рассмотрим в пространстве W полное подпространство $\tilde{W}$, в котором задано приближенное [по отношению к уравнению (2)] операторное уравнение [3, 4]

$$\tilde{p}(x,\omega,t) - \lambda\ PF(\omega)\tilde{p}(x,\omega,t) = Pf(x,\omega,t), \qquad (3)$$

где P — непрерывный линейный оператор, проектирующий W на $\tilde{W}$, для которого $PW = \tilde{W}$, $P^2 = P$.

Будем считать, что выполнены следующие *условия*:

1) для любого $p(x, \omega, t) \in W$ найдется элемент $\tilde{p}(x, \omega, t) \in \tilde{W}$ такой, что $\|F(\omega)p(x, \omega, t) - \tilde{p}(x, \omega, t)\| \le \le \eta_1\|p(x, \omega, t) \in \|$;

2) существует элемент $\tilde{f}(x, \omega, t) \in \tilde{W}$ такой, что $\|f(x, \omega, t) - \tilde{f}(x, \omega, t)\| \le \eta_2\|f(x, \omega, t)\|$.

Пусть каждый элемент $\tilde{p}(x, \omega, t) \in \tilde{W}$ единственным образом представим в виде $\tilde{p}(x, \omega, t) = \sum_{i=1}^{\infty} c_i(\omega)\gamma_i(x,t)$, $\gamma_i(x, t) \in \tilde{W}$, где система элементов $\{\gamma_i(x,t)\}_{i=1}^{\infty}$ образует базис в $\tilde{W}$. Кроме того, считаем заданной полную в $\tilde{W}$ систему $\{D_j\}$ линейных функционалов такую, что из равенств $D_j[\tilde{p}(x, \omega, t)] = 0$, $j = 1, 2, ...$ следует $\tilde{p}(x, \omega, t) = 0$. В этом случае вместо (3) можно ограничиться рассмотрением системы равенств $D_j[P(I - \lambda F(\omega))\tilde{p}(x, \omega, t)] = = D_j[Pf(x, \omega, t)]$, $j = 1, 2, ...$ С учетом этого приходим к бесконечной системе линейных алгебраических уравнений (СЛАУ) метода Галеркина в абстрактной форме [4]

$$\sum_{k=1}^{\infty} c_k(\omega)D_j\left[\gamma_k(x,t)\right] - \lambda\sum_{k=1}^{\infty} c_k(\omega)D_j\left[PF(\omega)\gamma_k(x,t)\right] = D_j\left[Pf(x,\omega,t)\right],\ j = 1, 2, ...$$

Если система функционалов $\{D_j\}$ биортогональна базису $\{\gamma_i(x,t)\}_{i=1}^{\infty}$, то $c_j(\omega) - \lambda\sum_{k=1}^{\infty} c_k(\omega) \times \times D_j\left[PF(\omega)\gamma_k(x,t)\right] = D_j\left[Pf(x,\omega,t)\right]$, $j = 1, 2, ...$ В частности, если W — гильбертово пространство, а P — оператор ортогонального проектирования, то $c_j(\omega) - \lambda\sum_{k=1}^{\infty} c_k(\omega)\left\langle F(\omega)\gamma_k(x,t), \gamma_j(x,t)\right\rangle =$

$= \langle f(x, \omega, t), \gamma_j(x, t) \rangle$, $j = 1, 2, ...,$ где $\langle , \rangle$ — символ скалярного произведения.

Представим данную систему в окончательном виде

$$c_j(\omega) - \lambda \sum_{k=1}^{\infty} a_{jk}(\omega) c_k(\omega) = b_j(\omega), \ j = 1, 2, ..., \quad (4)$$

полагая, что $\sum_{j,k=1}^{\infty} |a_{jk}(\omega)|^2 < \infty$, $\sum_{j=1}^{\infty} |b_j(\omega)|^2 < \infty$, а решение $\overset{*}{\mathbf{c}}(\omega) = \left\{ \overset{*}{c}_1(\omega), \overset{*}{c}_2(\omega), ... \right\}$ удовлетворяет условию $\sum_{k=1}^{\infty} \left| \overset{*}{c}_k(\omega) \right|^2 < \infty$.

Таким образом, задача нахождения приближенного аналитического решения параметризованного ЭУ сводится к решению бесконечной СЛАУ (4).

Решение бесконечной системы уравнений

Применим к решению бесконечной СЛАУ метод редукции, который состоит в замене (4) усеченной СЛАУ:

$$c_j(\omega) - \lambda \sum_{k=1}^{n} a_{jk}(\omega) c_j(\omega) = b_j(\omega), \ j = \overline{1, n}, \quad (5)$$

решение которой $\overset{*}{c}_n(\omega) = \left\{ \overset{*}{c}_{nj}(\omega), j = \overline{1, n} \right\}$ — приближенное решение (4).

Рассмотрим систему (4) с учетом ограничений в виде одного операторного уравнения в функциональном банаховом пространстве $C = l^2$ по аналогии с $\mathbf{c}(\omega) - \lambda K(\omega)\mathbf{c}(\omega) = \mathbf{b}(\omega)$ [3], где $\mathbf{c}(\omega) = \{c_1(\omega), c_2(\omega), ...\}$; $\mathbf{b}(\omega) = \{b_1(\omega), b_2(\omega), ...\}$; $K(\omega)$ — непрерывный линейный компактный оператор в l^2, определяемый для всех $\omega \in \Omega$ матрицей $\mathbf{A}(\omega) = \{a_{jk}(\omega), j, k = 1, 2, ...\}$ системы (4); $\|\mathbf{c}(\omega)\|_{l^2} = \left[\sum_{k=1}^{\infty} |c_k(\omega)|^2 \right]^{1/2}$.

Аналогично систему (5) рассмотрим в конечномерном пространстве $C_n = l_n^2$. $\mathbf{c}_n(\omega) - \lambda K_n(\omega)\mathbf{c}_n(\omega) = \mathbf{b}_n(\omega)$, где $\mathbf{c}_n(\omega) = \{c_j(\omega), j = 1, ..., n\}$ и $\mathbf{b}_n(\omega) = \{b_j(\omega), j = 1, ..., n\}$, а оператор $K_n(\omega)$ определяется усеченной матрицей $\mathbf{A}_n(\omega) = \{a_{jk}(\omega), j, k = 1, ..., n\}$; $\|\mathbf{c}_n(\omega)\|_{l_n^2} = \left[\sum_{k=1}^{n} |c_k(\omega)|^2 \right]^{1/2}$.

Наряду с пространствами C и C_n рассмотрим вспомогательное пространство $C_{[n]} \subset l^2$, состоящее из элементов, все координаты которых, начиная с $(n + 1)$-й, равны нулю. Обозначим через H_n непрерывный линейный оператор, отображающий $C_{[n]}$ взаимно однозначно на C_n, т. е. элементу $\mathbf{c}_{[n]}(\omega) = \{c_1(\omega), c_2(\omega), ..., c_n(\omega), 0, 0, ...\} \in C_{[n]}$ ставится в соответствие элемент $\mathbf{c}_n(\omega) = \{c_j(\omega), j = 1, ..., n\} \in C_n$.

Очевидно, что существует непрерывный обратный оператор H_n^{-1}. Наряду с H_n существует также непрерывный линейный оператор Q_n, являющийся продолжением оператора H_n, т. е. отображающий C на C_n и совпадающий с H_n на $C_{[n]}$. Оператор Q_n сопоставляет элементу $\mathbf{c}(\omega) = \{c_1(\omega), c_2(\omega), ...\} \in C = l^2$ элемент $\mathbf{c}_n(\omega) = \{c_j(\omega), j = \overline{1, n}\} \in C_n = l_n^2$: $\mathbf{c}_n(\omega) = Q_n c(\omega) = \{c_1(\omega), c_2(\omega), ..., c_n(\omega)\} \in l_n^2$. Также очевидно, что $\|Q_n\| = \|H_n\| = \|H_n^{-1}\| = 1$, $\|K_n(\omega)H_n\mathbf{c}_{[n]}(\omega) - Q_n F(\omega)\mathbf{c}_{[n]}(\omega)_n\| = 0$,

$$\left\| K(\omega)\mathbf{c}(\omega) - [K(\omega)\mathbf{c}(\omega)]_n \right\| = \left[\sum_{j=n+1}^{\infty} \sum_{k=1}^{\infty} |a_{jk}(\omega) c_k(\omega)|^2 \right]^{1/2} \le$$

$$\le \left[\sum_{j=n+1}^{\infty} \sum_{k=1}^{\infty} |a_{jk}(\omega)|^2 \sum_{k=1}^{\infty} |c_k(\omega)|^2 \right]^{1/2} \le \sigma_n \|\mathbf{c}(\omega)\|,$$

где под $[K(\omega)\mathbf{c}(\omega)]_n$ следует понимать усеченный элемент, получающийся из элемента $K(\omega)\mathbf{c}(\omega) \in l^2$ заменой всех его координат, начиная с $(n + 1)$-ой, нулями, $\sigma_n = \sup_{\omega} \left[\sum_{j=n+1}^{\infty} \sum_{k=1}^{\infty} |a_{jk}(\omega)|^2 \right]^{1/2}$, $\omega \in \Omega$. Очевидно, что с учетом принятых ограничений $\sigma_n \to 0$ при $n \to \infty$.

Кроме того:

$$\left\| \mathbf{c}(\omega) - [\mathbf{c}(\omega)]_n \right\| = \left[\sum_{j=n+1}^{\infty} |b_j(\omega)|^2 \right]^{1/2} \le \mu_n \|\mathbf{c}(\omega)\|,$$

где $\mu_n = \sup_{\omega} \left[\sum_{j=n+1}^{\infty} |b_j(\omega)|^2 \Big/ \sum_{j=1}^{\infty} |b_j(\omega)|^2 \right]^{1/2}$, $\omega \in \Omega$. При этом $\mu_n \to 0$ при $n \to \infty$.

На основе результатов работы [3] и с учетом вышесказанного можно заключить, что если λ не является характеристическим значением системы (4), то для фиксированного $\omega \in \Omega$ при достаточно больших n система (5) разрешима относительно $\overset{*}{\mathbf{c}}_n(\omega) = \left\{ \overset{*}{c}_{n1}(\omega), \overset{*}{c}_{n2}(\omega), ..., \overset{*}{c}_{nn}(\omega) \right\}$ и имеет место сходимость приближенных решений $\overset{*}{\mathbf{c}}_{[n]}(\omega) = \left\{ \overset{*}{c}_{n1}(\omega), \overset{*}{c}_{n2}(\omega), ..., \overset{*}{c}_{nn}(\omega), 0, 0, ... \right\}$ к точному $\overset{*}{\mathbf{c}}(\omega)$.

Скорость сходимости определяется неравенством $\left\| \overset{*}{\mathbf{c}}(\omega) - H_n^{-1} \overset{*}{\mathbf{c}}_n(\omega) \right\| = \left\| \overset{*}{\mathbf{c}}(\omega) - \overset{*}{\mathbf{c}}_{[n]}(\omega) \right\| \le q_1 \sigma_n +$

$+q_2\mu_n$, $\omega\in\Omega$, где $\overset{*}{\mathbf{c}}(\omega)$ и $\overset{*}{\mathbf{c}}_n(\omega)$ — решения систем (4) и (5) соответственно; q_1 и q_2 — положительные постоянные, не зависящие от ω и n. Отсюда ясно, что каждая координата $\overset{*}{c}_k(\omega)$ вектора $\overset{*}{\mathbf{c}}(\omega)$ мало отличается от каждой координаты $\overset{*}{c}_{nk}(\omega)$ вектора $\overset{*}{\mathbf{c}}_n(\omega)$ для всех $k = 1, ..., n$ и $\omega \in \Omega$, а при $k > n$ координата $\overset{*}{c}_k(\omega)$ мала для всех $\omega \in \Omega$. Кроме того, следует сходимость $\lim\limits_{n\to\infty} \overset{*}{c}_{nk}(\omega) = \overset{*}{c}_k(\omega)$, $k = 1, 2, ...$

Ниже рассмотрим общий подход к построению приближенного параметризованного решения системы (4) на базе усеченной системы (5).

Решение усеченной системы уравнений

Для сокращения записей, не снижая общности рассуждений, положим $\omega \in \Omega \subset R^1$. Пусть внутри области Ω задан набор точек (узлов) $\omega_{(i)}$, $i = 1, ..., N$. Поставим в соответствие набору $\omega_{(1)}, \omega_{(2)}, ..., \omega_{(N)}$ семейство $\overset{*}{c}_n(\omega_{(1)}), \overset{*}{c}_n(\omega_{(2)}), \ldots, \overset{*}{c}_n(\omega_{(N)})$ точных решений системы (5), т. е. $\overset{*}{c}_{nj}(\omega_{(i)}) - \lambda\sum\limits_{k=1}^{n} a_{jk}(\omega_{(i)})\overset{*}{c}_{nj}(\omega_{(i)}) = b_j(\omega_{(i)})$, $j = 1, ..., n$, $i = 1, ..., N$. Данные решения могут быть построены заранее, в нормальных (безаварийных) условиях эксплуатации, с использованием известных методов решения СЛАУ на базе ЭВМ.

Используя введенное семейство, рассмотрим процедуру построения приближенного параметризованного решения $\tilde{\mathbf{c}}_n(\omega) = \{\tilde{c}_{n1}(\omega), \tilde{c}_{n2}(\omega), ..., \tilde{c}_{nn}(\omega)\}$ системы (5), справедливого для всех $\omega\in\Omega$. На базе данного решения сформируем вектор $\tilde{\mathbf{c}}_{[n]}(\omega) = \{\tilde{c}_{n1}(\omega), \tilde{c}_{n2}(\omega), \ldots, \tilde{c}_{nn}(\omega), 0, 0, \ldots\} = H_n^{-1} \times \times \tilde{\mathbf{c}}_n(\omega)$, который принимается в качестве приближенного параметризованного решения для системы (4) и обеспечивает при этом выполнение следующего неравенства: $\sup\limits_{\omega}\left\|\overset{*}{\mathbf{c}}(\omega) - \tilde{\mathbf{c}}_{[n]}(\omega)\right\| = \sup\limits_{\omega}\left\|\overset{*}{\mathbf{c}}(\omega) - H_n^{-1}\tilde{\mathbf{c}}_n(\omega)\right\| \le \delta_{n,N}$, $\omega \in \Omega$, где $\delta_{n,N}$ — положительная постоянная, задающая границу допустимой погрешности вычислений.

Очевидно, что количество и правило расположения указанных выше узлов $\omega_{(1)}, \omega_{(2)}, ..., \omega_{(N)}$ зависит от выбора области Ω и требуемой точности построения параметризованного решения (4), которая определяется константой $\delta_{n,N}$.

Для фиксированного $j = 1, ..., n$ поставим в соответствие узлам $\omega_{(1)}, \omega_{(2)}, ..., \omega_{(N)}$ набор чисел $\overset{*}{c}_{nj}(\omega_{(1)}), \overset{*}{c}_{nj}(\omega_{(2)}), \ldots, \overset{*}{c}_{nj}(\omega_{(N)})$, который соответствует j-м координатам построенных опорных решений $\overset{*}{c}_n(\omega_{(1)}), \overset{*}{c}_n(\omega_{(2)}), \ldots, \overset{*}{c}_n(\omega_{(N)})$ системы (5). Проведем интерполяцию данного набора, сопоставив ему скалярную функцию $\psi_{nj}(\omega)$ известного класса: $\psi_{nj}(\omega) = \theta_n(\omega, \boldsymbol{\nu}_j)$, $\boldsymbol{\nu}_j \in R^N$, $j = 1, ..., n$, где вектор коэффициентов $\boldsymbol{\nu}_j = \{\nu_{j1}, \nu_{j2}, ..., \nu_{jN}\}$ находится путем решения следующей СЛАУ: $\psi_{nj}(\omega_{(i)}) = \theta_n(\omega_{(i)}, \boldsymbol{\nu}_j) = \overset{*}{c}_{nj}(\omega_{(i)})$, $i = 1, ..., N$.

Данное соотношение показывает, что вектор коэффициентов $\boldsymbol{\nu}_j$ выбирается таким образом, чтобы значения функции $\psi_{nj}(\omega)$ совпадали со значениями функции $\overset{*}{c}_{nj}(\omega)$ в N узлах интерполяции, а его решением являлся вектор коэффициентов $\overset{*}{\boldsymbol{\nu}}_j = \left\{\nu_{jk}, k=\overline{1, N}\right\}$. Далее проводится интерполяция для всех $j = 1, ..., n$, т. е. определяется совокупность параметризованных коэффициентов $\psi_{n1}(\omega), \psi_{n2}(\omega), ..., \psi_{nn}(\omega)$, удовлетворяющих характеристическому свойству. Указанные коэффициенты принимаются в качестве параметризованных координат приближенного решения $\tilde{\mathbf{c}}_n(\omega) = \{\tilde{c}_{n1}(\omega), \tilde{c}_{n2}(\omega), ..., \tilde{c}_{nn}(\omega)\}$ системы (5), т. е. $\tilde{c}_{nj}(\omega) = \psi_{nj}(\omega)$ и $\tilde{\mathbf{c}}_n\left(\omega_{(i)}\right) = \boldsymbol{\Psi}_n\left(\omega_{(i)}\right) = \boldsymbol{\Theta}_n\left(\omega_{(i)}, \overset{*}{V}_n\right) = \overset{*}{\mathbf{c}}_n\left(\omega_{(i)}\right)$, $i = 1, ..., N$, где $\boldsymbol{\Theta}_n\left(\omega_{(i)}, \overset{*}{V}_n\right) = \left\{\theta_n\left(\omega_{(i)}, \overset{*}{\nu}_1\right), \theta_n\left(\omega_{(i)}, \overset{*}{\nu}_2\right), \ldots, \theta_n\left(\omega_{(i)}, \overset{*}{\nu}_n\right)\right\}$, $\boldsymbol{\Psi}_n(\omega_{(i)}) = \{\psi_{n1}(\omega_{(i)}), \psi_{n2}(\omega_{(i)}), ..., \psi_{nn}(\omega_{(i)})\}$. Данные соотношения показывают, что построенное приближенное параметризованное решение $\tilde{\mathbf{c}}_n(\omega)$ системы (5) совпадает с ее точным параметризованным решением $\overset{*}{\mathbf{c}}_n(\omega)$ в узлах интерполяции $\omega_{(i)}$, $i = 1, ..., N$.

Приближенным параметризованным решением системы (4) следует принять

$$\tilde{\mathbf{c}}_{[n]}(\omega) = \left\{\tilde{c}_{n1}(\omega), \tilde{c}_{n2}(\omega), \ldots, \tilde{c}_{nn}(\omega), 0, 0, \ldots\right\} = \left\{\theta_n\left(\omega, \overset{*}{\nu}_1\right), \theta_n\left(\omega, \overset{*}{\nu}_2\right), \ldots, \theta_n\left(\omega, \overset{*}{\nu}_n\right), 0, 0, \ldots\right\}.$$

Применим к рассмотренной выше процедуре построения параметризованных решений известные методы интерполяции. Так, в случае параболической интерполяции на основе степенных полиномов $\psi_{nj}(\omega) = \sum\limits_{k=1}^{N} \nu_{jk}\omega^k$ получим СЛАУ

$$\psi_{nj}\left(\omega_{(i)}\right)=\sum_{k=1}^{N}\nu_{jk}\omega_{(i)}^{k}=\overset{*}{c}_{nj}\left(\omega_{(i)}\right),\ i=\overline{1,N},$$
$$j=1,\ldots,n. \qquad (6)$$

Решая систему (6) относительно $\boldsymbol{\nu}_j=\{\nu_{j1},\ \nu_{j2},\ \ldots,\ \nu_{jN}\}$ для $j=1,\ \ldots,\ n$, получим $\tilde{\mathbf{c}}_{[n]}(\omega)=\left\{\sum_{k=1}^{N}\overset{*}{\nu}_{1k}\omega^{k},\ \sum_{k=1}^{N}\overset{*}{\nu}_{2k}\omega^{k},\ldots,\ \sum_{k=1}^{N}\overset{*}{\nu}_{nk}\omega^{k},0,0,\ldots\right\}$. Очевидно, что

$$\tilde{\mathbf{c}}_{[n]}\left(\omega_{(i)}\right)=$$
$$=\left\{\tilde{c}_{n1}\left(\omega_{(i)}\right),\tilde{c}_{n2}\left(\omega_{(i)}\right),\ldots,\tilde{c}_{nn}\left(\omega_{(i)}\right),0,0,\ldots\right\}=$$
$$=\left\{\overset{*}{c}_{n1}\left(\omega_{(i)}\right),\overset{*}{c}_{n2}\left(\omega_{(i)}\right),\ldots,\overset{*}{c}_{nn}\left(\omega_{(i)}\right),0,0,\ldots\right\},$$

где $\left\{\overset{*}{c}_{n1}\left(\omega_{(i)}\right),\ \overset{*}{c}_{n2}\left(\omega_{(i)}\right),\ldots,\overset{*}{c}_{nn}\left(\omega_{(i)}\right)\right\}=\overset{*}{\mathbf{c}}_{n}\left(\omega_{(i)}\right)$ — точное решение системы (5), соответствующее узлу $\omega_{(i)}\in\Omega$.

Предложенный подход к построению параметризованного решения на базе интерполяционного полинома имеет существенный недостаток: при увеличении семейства опорных решений необходимо многократно решать СЛАУ (6) в целях вычисления искомых коэффициентов $\boldsymbol{\nu}_{jk}$, $j=1,\ \ldots,\ n$, $k=1,\ \ldots,\ N$. От этого недостатка свободна реализация рассматриваемого подхода к построению параметризованных решений на базе интерполяционного полинома Лагранжа.

Для параболической интерполяции на основе полинома Лагранжа получим

$$\psi_{nj}(\omega)=\sum_{k=1}^{N}\nu_{jk}L_{k}(\omega)=$$
$$=\sum_{k=1}^{N}\nu_{jk}\prod_{\substack{p=0\\p\neq 0}}^{N}\frac{\omega-\omega_{(p)}}{\omega_{(k)}-\omega_{(p)}},\ j\in\overline{1,n}.$$

Учитывая, что $L_k(\omega_{(i)})$ равно 1 для $k=i$ и 0 для $k\neq i$, несложно убедиться в выполнении следующего характеристического свойства: $\psi_{nj}\left(\omega_{(i)}\right)=$ $=\sum_{k=1}^{N}\nu_{jk}L_{k}\left(\omega_{(i)}\right)=\nu_{ji}=\overset{*}{c}_{nj}\left(\omega_{(i)}\right),\ i=1,\ \ldots,\ N.$

Тогда получим

$$\psi_{nj}(\omega)=\sum_{k=1}^{N}\psi_{nj}\left(\omega_{(k)}\right)L_{k}(\omega)=$$
$$=\sum_{k=1}^{N}\overset{*}{c}_{nj}\left(\omega_{(k)}\right)L_{k}(\omega),\ j=1,\ldots,n. \qquad (7)$$

Аналогично приближенное параметризованное решение $\tilde{\mathbf{c}}_{[n]}(\omega)$ системы (4) представим в следующем виде:

$$\tilde{\mathbf{c}}_{[n]}(\omega)=\left\{\sum_{k=1}^{N}\overset{*}{c}_{n1}\left(\omega_{(k)}\right)L_{k}(\omega),\ \sum_{k=1}^{N}\overset{*}{c}_{n2}\left(\omega_{(k)}\right)L_{k}(\omega),\ldots,\right.$$
$$\left.\sum_{k=1}^{N}\overset{*}{c}_{nn}\left(\omega_{(k)}\right)L_{k}(\omega),0,0,\ldots\right\}, \qquad (8)$$

при этом несложно убедиться в выполнении характеристического свойства.

Обозначив через $\overset{*}{\mathbf{c}}(\omega)$ точное аналитическое параметризованное решение системы (4), через $\left[\overset{*}{\mathbf{c}}(\omega)\right]_n$ — вектор, получающийся из $\overset{*}{\mathbf{c}}(\omega)$ путем обнуления всех координат, начиная с $(n+1)$-й, а через $\tilde{\mathbf{c}}_{[n]}(\omega)$ — приближенное решение системы (5), для оценки результирующей погрешности вычислений можно воспользоваться неравенством треугольника

$$\sup_{\omega}\left\|\overset{*}{\mathbf{c}}(\omega)-\tilde{\mathbf{c}}_{[n]}(\omega)\right\|\leq\sup_{\omega}\left\|\overset{*}{\mathbf{c}}(\omega)-\left[\overset{*}{\mathbf{c}}(\omega)\right]_n\right\|+$$
$$+\sup_{\omega}\left\|\left[\overset{*}{\mathbf{c}}(\omega)\right]_n-\tilde{\mathbf{c}}_{[n]}(\omega)\right\|. \qquad (9)$$

Конкретизация данной формулы зависит от метода интерполяции, выбранного в ходе построения параметризованного решения.

Если оценка слагаемого $\sup_{\omega}\left\|\overset{*}{\mathbf{c}}(\omega)-\left[\overset{*}{\mathbf{c}}(\omega)\right]_n\right\|$ давалась ранее, то в качестве оценки слагаемого $\sup_{\omega}\left\|\left[\overset{*}{\mathbf{c}}(\omega)\right]_n-\tilde{\mathbf{c}}_{[n]}(\omega)\right\|$ можно в каждом конкретном случае использовать известные оценки для остаточного члена выбранного метода интерполяции.

Полученные результаты несложно распространить на многомерный случай. Тогда процедуры одномерной интерполяции, рассмотренные выше, заменяются процедурами многомерной интерполяции [5]. При этом в качестве погрешности вычислений могут быть приняты остаточные члены, соответствующие принятой многомерной интерполяции.

Выбор семейства опорных решений

Для рассматриваемого в настоящей работе подхода важнейшим является вопрос, связанный с выбором узлов интерполяции $\omega_{(i)}\in\Omega$, $i=1,\ \ldots,\ N$,

обеспечивающим минимизацию результирующей погрешности. Рассмотрим решение этого вопроса для случая $\Omega = [d_1, d_2] \subset R^1$.

Согласно [5], погрешность интерполяции на основе полинома Лагранжа оценивается с помощью остаточного члена $\overset{*}{c}_j(\omega) - \tilde{c}_{nj}(\omega) = \frac{Z_N(\omega)}{N!} \times \frac{\mathrm{d}^N \overset{*}{c}_j(\omega)}{\mathrm{d}\omega^N}$, $j = 1, ..., n$, где $Z_N(\omega) = \prod_{i=1}^{N} (\omega - \omega_{(i)})$.

При выборе семейства узлов $\omega_{(1)}, \omega_{(2)}, ..., \omega_{(N)}$, предполагающем обоснование значения N и оптимальное размещение узлов в области Ω, можно воспользоваться следующим неравенством:

$$\max_j \sup_\omega \left| \overset{*}{c}_j(\omega) - \tilde{c}_{nj}(\omega) \right| \le \frac{\overset{*}{Z}_N}{N!} \overset{*}{\mathrm{G}}_N, \quad j = \overline{1, n}, \ \omega \in \Omega, \tag{10}$$

где $\overset{*}{Z}_N = \sup_\omega Z_N(\omega)$; $\overset{*}{G}_N = \max_j \sup_\omega \left| \frac{\mathrm{d}^N \overset{*}{c}_j(\omega)}{\mathrm{d}\omega^N} \right|$.

Для минимизации погрешности интерполяции на основе полинома Лагранжа достаточно выбрать в качестве узлов $\omega_{(1)}, \omega_{(2)}, ..., \omega_{(N)}$ корни многочленов Чебышева, принадлежащие отрезку $\Omega = [d_1, d_2]$, причем для оценки сверху величины $\overset{*}{Z}_N$ воспользуемся соотношением $\overset{*}{Z}_N \le 2\left(\frac{d_2 - d_1}{4}\right)^N$. При этом вместо (10) имеем

$$\max_j \sup_\omega \left| \overset{*}{c}_j(\omega) - \tilde{c}_{nj}(\omega) \right| \le 2\left(\frac{d_2 - d_1}{4}\right)^N \frac{\overset{*}{G}_N}{N!}. \tag{11}$$

С учетом (11) можно дать оценку второго слагаемого в правой части неравенства (9):

$$\sup_\omega \left\| \left[\overset{*}{\mathbf{c}}(\omega) \right]_n - \tilde{\mathbf{c}}_{[n]}(\omega) \right\| \le 2\left(\frac{d_2 - d_1}{4}\right)^N \frac{\overset{*}{G}_N}{N!} n^{1/2}.$$

Таким образом, по заданному значению $\delta_{n,N}$ можно выбрать такие n и N, при которых для случая оптимального расположения узлов интерполяции $\omega_{(1)}, \omega_{(2)}, ..., \omega_{(N)}$ в области Ω достигается выполнение неравенства (11).

Результаты численных экспериментов показывают, что при гладкой зависимости параметризованных коэффициентов СЛАУ (4) от координат вектора ω количество узлов интерполяции для выбранной области оказывается незначительным. Например, для обеспечения погрешности менее 4,5 % решения уравнения Фоккера—Планка—Колмогорова вида [6] $\frac{\partial p(x,t)}{\partial x} = \frac{\partial}{\partial x}\left[x p(x,t) \right] + \frac{\partial^2}{\partial x^2}\left[p(x,t) \right]$, где $0 \le t \le T$, $-\infty \le x \le +\infty$, с параметризованным начальным условием $p(x,0) = \frac{1}{\omega_2 (2\pi)^{0,5}} \exp\left[-\frac{(x - \omega_1)^2}{2\omega_2^2} \right]$ для $2 \le \omega_1 \le 10$ и $0 \le \omega_2 \le 1$, было достаточно выбрать по четыре узла по каждому параметру ω.

Параметризованное решение эволюционного уравнения

Приближенное решение параметризованного ЭУ (2) можно представить в виде

$$\overset{*}{\tilde{p}}_n(x, \omega, t) = \sum_{i=1}^{n} \overset{*}{c}_{ni}(\omega) \gamma_i(x,t) = \sum_{i=1}^{n} \overset{*}{\theta}_n(\omega, \nu_i) \gamma_i(x,t),$$

а применительно к конкретным видам интерполяции: $\overset{*}{\tilde{p}}_n(x, \omega, t) = \sum_{i=1}^{n} \sum_{k=1}^{N} \overset{*}{\nu}_{ik} \omega^k \gamma_i(x,t)$ — для степенных полиномов; $\overset{*}{\tilde{p}}_n(x, \omega, t) = \sum_{i=1}^{n} \sum_{k=1}^{N} \overset{*}{c}_{ni}(\omega_{(k)}) \times L_k(\omega) \gamma_i(x,t)$ — для полинома Лагранжа.

Считаем, что для любого заданного $\varepsilon > 0$ можно указать такое n, при котором для всех $\omega \in \Omega$ выполняется ограничение $\left\| \sum_{i=n+1}^{\infty} \overset{*}{c}_i(\omega) \gamma_i(x,t) \right\| < \varepsilon$.

Для оценки результирующей погрешности воспользуемся неравенством треугольника

$$\left\| \overset{*}{\tilde{p}}_n(x, \omega, t) - \overset{*}{p}(x, \omega, t) \right\| \le \left\| \overset{*}{\tilde{p}}(x, \omega, t) - \overset{*}{p}(x, \omega, t) \right\| + \left\| \overset{*}{\tilde{p}}_n(x, \omega, t) - \overset{*}{\tilde{p}}(x, \omega, t) \right\|.$$

С учетом условий 1 и 2 для оценки нормы $\left\| \overset{*}{\tilde{p}}(x, \omega, t) - \overset{*}{p}(x, \omega, t) \right\|$ можно воспользоваться следующим соотношением [4]:

$$\left\| \overset{*}{\tilde{p}}(x, \omega, t) - \overset{*}{p}(x, \omega, t) \right\| \le q \left\| \overset{*}{p}(x, \omega, t) \right\|,$$

где $q = |\lambda| \varepsilon \| I - P \| \left\| (I - \lambda F(\omega))^{-1} \right\|$, на базе которого легко получить оценку близости приближенного $\overset{*}{\tilde{p}}(x, \omega, t)$ и точного $\overset{*}{p}(x, \omega, t)$ решений:

$$\left\| \overset{*}{\tilde{p}}(x, \omega, t) - \overset{*}{p}(x, \omega, t) \right\| \le q(1 - q)^{-1} \left\| \overset{*}{\tilde{p}}(x, \omega, t) \right\|, \quad q < 1, \ \omega \in \Omega.$$

Близость решений $\tilde{p}(x, \omega, t)$ и $\tilde{p}_n(x, \omega, t)$ определяется выражением [4]

$$\left\| \overset{*}{\tilde{p}}_n(x,\omega,t) - \overset{*}{\tilde{p}}(x,\omega,t) \right\| \le \left\| \sum_{i=1}^{n} \left[\overset{*}{c}_{ni}(\omega) - \overset{*}{c}_i(\omega) \right] \gamma_i(x,t) \right\| + \left\| \sum_{i=n+1}^{\infty} \overset{*}{c}_i(\omega)\gamma_i(x,t) \right\|.$$

Минимизация первого слагаемого в правой части данного неравенства достигается за счет выбора эффективного метода интерполяции и требуемого числа узлов $\omega_{(1)}, \omega_{(2)}, ..., \omega_{(N)}$. Второе слагаемое удовлетворяет принятому ограничению $\left\| \sum_{i=n+1}^{\infty} \overset{*}{c}_i(\omega)\gamma_i(x,t) \right\| < \varepsilon$, которое определяется значением n. Результирующая погрешность может быть получена, если учесть указанные выше соотношения.

Вычисление требуемых стохастических характеристик

На основе найденного решения $\overset{*}{\tilde{p}}(x, \omega, t)$ получим семейство оценок искомых приближенных СХ: $\overset{*}{\tilde{Y}}_i(\omega) = F_i\left[\overset{*}{\tilde{p}}(x,\omega,t)\right]$, $i = 1, ..., M_0$. Очевидно, что методическая погрешность оценки складывается из двух составляющих: погрешности получения $\overset{*}{\tilde{p}}(x, \omega, t)$ — решения параметризованных ЭУ и погрешности вычисления СХ, определяемой свойствами линейных функционалов $F_i[\cdot]$, $i = 1, ..., M_0$. Искомые СХ представим в следующем виде:

$$\overset{*}{\tilde{Y}}_i(\omega) = F_i\left[\overset{*}{p}(x,\omega,t) + \Delta p(x,\omega,t)\right] =$$

$$= F_i\left[\overset{*}{p}(x,\omega,t)\right] + F_i\left[\Delta p(x,\omega,t)\right] = \overset{*}{Y}_i(\omega) + \Delta\overset{*}{Y}_i(\omega).$$

С учетом этого получим оценку методической погрешности $\left\| \Delta\overset{*}{Y}_i(\omega) \right\| \le \|F_i\| \left\| \Delta p(x,\omega,t) \right\|$, где $\|F_i\|$ — норма функционала F_i; $\|\Delta p(x, \omega, t)\|$ — норма погрешности интегрирования ЭУ.

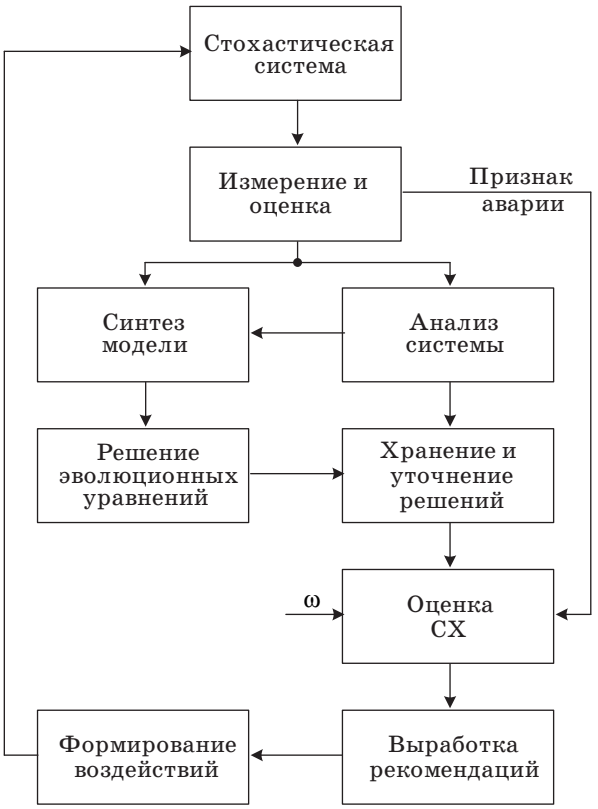
Сравнительный показатель быстродействия разработанного и классического методов оценки СХ определяется выражением

$$S_T(M_0) = \left(T_0 + \sum_{i=1}^{M_0} T_i^K \right) \left(\sum_{i=1}^{M_0} T_i^P \right)^{-1},$$

где T_0, T_i^K, T_i^P — время, затрачиваемое на интегрирование параметризованных ЭУ и на вычисление i-й СХ классическим и разработанным методами соответственно.

Поскольку время решения соответствующего параметризованного ЭУ значительно больше времени, затрачиваемого на вычисление конкретной СХ, то справедливы соотношения $T_0 >> \sum_{i=1}^{M_0} T_i^K$, $T_0 >> \sum_{i=1}^{M_0} T_i^P$. Полагая, что $\sum_{i=1}^{M_0} T_i^K = \sum_{i=1}^{M_0} T_i^P = T_{M_0}$ приходим к оценке $S_T \approx T_0 \left(T_{M_0}\right)^{-1}$. Таким образом, применительно к разработанному опорно-параметрическому методу, выигрыш в быстродействии оценки СХ по сравнению с классическим методом в основном определяется отношением времени на решение параметризованного ЭУ к времени вычисления всех СХ. При этом для многомерных марковско-параметрических систем с учетом выполнения условия $T_0 >> T_{M_0}$ указанный выигрыш может достигать нескольких порядков.

На основании предложенного метода может быть синтезирована ИУС, осуществляющая формирование управляющих воздействий с учетом полученных СХ (рисунок). Особенностью ее функ-



■ *Схема численно-аналитического метода оценки стохастических характеристик марковской системы*

ционирования является то, что при выявлении признака аварийной ситуации в исследуемой системе выдается соответствующий сигнал в блок оценки СХ, т. е. реализуется сразу второй этап алгоритма.

Это дает возможность существенно повысить быстродействие оценки СХ и выработки рекомендаций по формированию управляющих воздействий на систему, направленных на устранение аварийной ситуации.

Заключение

В работе предложены теоретические положения метода оперативного высокоточного оценивания СХ марковских систем в опорно-параметрической постановке. Получены аналитические соотношения, позволяющие рассчитывать основные параметры метода, при которых обеспечиваются требуемые точность и быстродействие оценки СХ. Необходимость вычисления и хранения большого количества значений коэффициентов решений параметризованных ЭУ не является препятствием при использовании современных ЭВМ, оснащенных запоминающими устройствами большой емкости.

Использование разработанного метода наиболее целесообразно для оперативной оценки совокупности СХ, в задачах, связанных с возникновением чрезвычайных ситуаций техногенного или природного характера, катастроф и т. д. [2], которые достаточно эффективно моделируются в рамках теории марковских динамических систем. На базе разработанного метода предложен вариант структуры ИУС.

Литература

1. **Пугачев В. С., Синицын И. Н.** Теория стохастических систем. — М.: Логос, 2000. — 1000 с.
2. **Острейковский В. А., Сальников Н. Л.** Вероятностное прогнозирование работоспособности ЯЭУ. — М.: Энергоатомиздат, 1990. — 416 с.
3. **Красносельский М. А. и др.** Приближенное решение операторных уравнений. — М.: Наука, 1969. — 456 с.
4. **Булычев Ю. Г., Лапсарь А. П.** Моделирование эволюционных систем с использованием опорно-проекционного метода // Математическое моделирование. 1998. Т. 10. № 1. С. 20–30.
5. **Иванов В. В.** Методы вычислений на ЭВМ: справ. пособие. — Киев: Наук. думка, 1986. — 564 с.
6. **Тихонов В. И., Миронов М. А.** Марковские процессы. — М.: Сов. радио, 1977. — 488 с.

УДК 681.52

СИСТЕМА АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ УЧЕБНОЙ ДЕЯТЕЛЬНОСТЬЮ И ЕЕ ДИАГНОСТИКИ

П. П. Дьячук,
канд. физ.-мат. наук, доцент
Л. Н. Дроздова,
канд. мед. наук, доцент
И. В. Шадрин,
канд. техн. наук
Красноярский государственный педагогический университет им. В. П. Астафьева

В рамках информационной модели развития учебной деятельности, регулируемой системой автоматического управления Tr@cK, рассмотрен процесс научения решению задач. Проведена диагностика учебной деятельности, а ее результаты сопоставлены с результатами диагностики уровня развития базовых когнитивных функций мозга.

Ключевые слова *— системы управления, автоматическое регулирование, диагностика учебной деятельности.*

Введение

Учебную деятельность обучающегося решению задач можно рассматривать как процесс развития потому, что с ней связано получение новой информации, в результате опыта совершения действий. Этот процесс происходит вследствие итеративного научения, а результат является следствием решения последовательности аналогичных задач и перехода от незнания к знанию путем продуцирования информации при взаимодействии обучающегося с проблемной средой. Особую роль в системе «обучающийся — проблемная среда» играет процесс изменения (развития) структуры системы действий обучающегося, регулируемый с помощью каналов обратной связи.

Основой функционирования главной и местной обратных связей являются исполнительные механизмы, преобразующие интерфейс проблемной среды. Они реализуют институциональное (ограничение набора допустимых действий), информационное (реализованное в виде индикатора расстояния до цели, информирующего о количестве действий, которые необходимо совершить для перехода в целевое состояние) и мотивационное (отображение изменений функции ценности состояния обучающегося с помощью дискретной системы уровней деятельности) управление учебной деятельностью. Постоянно, пока обучающийся не решит задачу (и не научится решать задачи данного типа), проблемная среда будет посылать сигналы. При этом индивидуальные способности обучающихся влияют лишь на процесс поиска решения задачи, но не на результат. Такая система, управляющая самоорганизацией деятельности обучающегося, получила название автоматического регулятора учебной деятельности *Tr@cK* (далее регулятор *Tr@cK*). Следуя работе [1], опишем принципы его функционирования.

Регулятор *Tr@cK*

Регулятор *Tr@cK* предназначен для управления учебной деятельностью обучающихся решению задач или проблем (обобщенное название компьютерных программ, созданных на его основе — «Проблемные среды»). Употребление термина «учебная деятельность» [2] обусловлено тем, что регулятор *Tr@cK* не управляет процессом обучения, а создает для обучающегося условия, позволяющие реализовать его поисковую активность с помощью некоторого набора доступных действий.

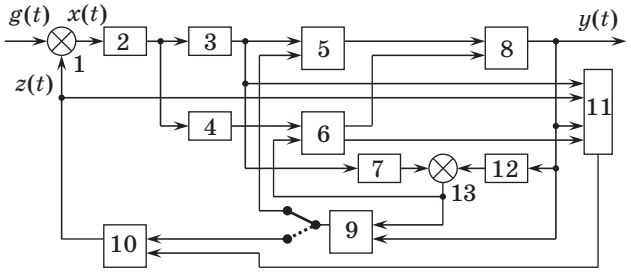
Цель функционирования регулятора *Tr@cK* состоит в том, чтобы привести структуру системы действий обучающегося — набор осуществляемых им действий и их последовательность — в такое состояние, когда каждое совершаемое действие будет приближать решение задачи. Для

достижения этой цели регулятор *Tr@cK* поощряет правильные действия и угнетает неправильные. Общая структурная схема регулятора *Tr@cK* представлена на рис. 1.

Регулятор *Tr@cK* производит поиск такого требуемого значения параметров местной обратной связи (аргументов передаточной функции звена *6*, реализующего эту связь), при котором структура системы действий $z(t)$ обучающегося *8* будет соответствовать целям функционирования регулятора $g(t)$. Проверка этого соответствия осуществляется в цепи главной обратной связи в моменты срабатывания переключателя *9*, когда включается звено *10*, определяющее параметры структуры системы действий обучающегося на основе сохраненной во внешней памяти *11* последовательности действий.

При этом истинные законы изменения параметров структуры системы действий обучающегося $z(t)$ установить невозможно в силу объективных причин, зависящих от психических, физиологических, интеллектуальных и других индивидуальных особенностей конкретного человека. Обучающийся, деятельность которого подлежит регулированию, является «черным ящиком». Подавая на его входы (органы чувств, в частности глаза или уши) управляющие сигналы, смысл которых ему знаком, на выходе мы имеем сигналы (в виде зафиксированных действий, доступных в проблемной среде, им совершаемых). Передаточную функцию этого звена нельзя определить заранее. Более того, анализ протоколов деятельности, сохраненных во внешней памяти, является наиболее интересным направлением исследования — он позволяет устанавливать вид и параметры передаточной функции для каждого обучающегося, т. е. диагностировать индивидуальные особенности осуществления учебной деятельности.

Элемент сравнения *1* производит вычитание $x(t) = g(t) - z(t)$ и тем самым определяет рассогласование между реальной структурой системы действий обучающегося $z(t)$ и требуемой $g(t)$ — исключающей неправильные действия. На основании вычисленной ошибки $x(t)$ звено *2* определяет уровень деятельности обучающегося L_i, где i — номер очередного формируемого звеном *3* задания — новой задачной (проблемной) ситуации. L_i дискретно изменяется во времени (после выполнения очередного задания) и определяется лишь параметрами структуры системы действий обучающегося при выполнении предыдущего $(i-1)$-го задания. $L_1 = 1$. Уровень деятельности отображается специальным датчиком в интерфейсе проблемной среды. В зависимости от значения уровня деятельности звено *4* определяет параметры функционирования местной обратной связи *6*.



■ ***Рис. 1.*** *Структурная схема регулятора учебной деятельности Tr@cK*

Сформированная звеном *3* задача отображается интерфейсом проблемной среды, приведенным модулем *5* в состояние, соответствующее начальным параметрам. Элемент памяти *7* сохраняет тот же набор параметров, но содержащий значения, достижение которых соответствует решению поставленной задачи. Преобразование объектов проблемной среды для достижения этого соответствия является для обучающегося локальной целью, которую он должен достичь, используя систему действий, доступных ему в проблемной среде. Кроме того, модуль *5* реализует все изменения интерфейса проблемной среды, связанные с действиями обучающегося. При этом формирование новой задачной ситуации (формирование соответствующего состояния интерфейса) происходит лишь в моменты включения главной обратной связи, а текущие изменения отображаются после каждого совершенного обучающимся действия.

Для реализации местной обратной связи после каждого действия обучающегося вычислительное звено *12* определяет изменение параметров объектов проблемной среды, а элемент сравнения *13* определяет рассогласование между текущей обстановкой и значениями, сохраненными звеном *7*. Величина этого рассогласования, выраженная в количестве дискретных шагов (каждый из которых — это конкретное действие обучающегося, дискретно изменяющее определенный параметр проблемной среды), определяет расстояние до цели (решения задачи). Эта информация, составляющая основу местной обратной связи, позволяет обучающемуся отличить правильные действия от ошибочных и достичь решения текущей задачи.

В моменты времени, когда расстояние до цели равно нулю, переключатель *9* может изменить свое состояние при поступлении от обучающегося сигнала об окончании выполнения задания. Если такой сигнал не поступает (обучающийся не нажимает соответствующую кнопку), регулятор продолжает функционировать по малому кругу через местную обратную связь. И напротив, если

расстояние до цели не равно нулю, переключатель *9* не изменит своего состояния при поступлении этого сигнала.

При изменении состояния переключателя *9* включается **контур главной обратной связи, в который** входит звено *10*, определяющее параметры структуры системы действий обучающегося на основе формализованной информации, сохраненной в модуле внешней памяти *11*. Отметим, что при формировании очередной проблемной ситуации вновь возникает рассогласование в элементе сравнения *13* и переключатель *9* переходит в состояние, когда сигналы проходят по контуру местной обратной связи и регулируют процесс поиска обучающимся решения текущей задачи.

Во внешней памяти сохраняется не только последовательность действий обучающегося с указанием затраченного времени, но и управляющие воздействия регулятора *Tr@cK*: условия поставленной задачи, параметры работы датчика «Расстояние до цели», параметры структуры системы действий обучающегося. Благодаря этой информации появляется возможность более сложного анализа деятельности обучающегося в любое удобное для исследователя время с применением различных методов и программных средств.

Приведенное описание показывает, что регулятор *Tr@cK* производит поиск такого режима работы местной обратной связи, при котором деятельность обучающегося наиболее эффективна. Учитывая, что истинные законы изменения параметров структуры системы действий обучающегося установить невозможно, регулятор *Tr@cK* можно определить как экстремальную самонастраивающуюся систему автоматического управления дискретного действия.

Учебная деятельность в проблемной среде

Учебная деятельность — это особый способ саморазвития обучающегося, направленный на освоение новых видов деятельности, продуцирование обучающимся новых знаний, умений и навыков. Актуальность диагностики индивидуальных особенностей ее осуществления сегодня не вызывает сомнений.

Структура системы действий, совершаемых обучающимся в проблемной среде, имеет сложный вид в силу разнообразия семантического смысла действий и порядка их выполнения. Однако если учитывать только синтаксическое значение данных о действиях обучающегося, то множество действий можно разделить на два подмножества: подмножество правильных действий (приближающих решение задачи) и подмножество неправильных действий (отдаляющих решение). Таким образом, деятельность можно формально представить в виде последовательности единиц и нулей — 1110110011..., т. е. в виде сообщения обучающегося (в синтаксической форме), характеризующего структуру системы его действий. В процессе научения решению задач доля правильных действий возрастает, соответственно доля неправильных действий уменьшается.

Трактуя развитие, как процесс снижения меры неупорядоченности (убывания энтропии H) действий, который проявляется в снижении неопределенности при принятии решения обучающимся о выборе действия, будем говорить о накоплении внутренней (субъективной для обучающегося) информации, такой, информации, которая позволила бы обучающемуся безошибочно находить решение задачи. Значение энтропии, характеризующее структуру системы действий обучающегося, можно вычислить по формуле Шеннона

$$H_i = -p_i \log_2 p_i - (1 - p_i)\log_2(1 - p_i),$$

где p_i — доля правильных действий при выполнении i-го задания, так как отмена ошибочного действия является правильным действием, всегда $p_i > 0{,}5$, а при $p_i \to 0{,}5$ (большом количестве ошибочных действий) $H_i \to 1$.

Отметим, что энтропия деятельности обучающегося H при условии предъявления ориентиров с частотой P_B^{i-1} при выполнении $(i - 1)$-го задания определяет параметры функционирования системы *Tr@cK* при поиске обучающимся решения i-го задания.

Таким образом, информацию, накопленную обучающимся при осуществлении деятельности в проблемной среде после выполнения i заданий, можно выразить как меру снятой неопределенности:

$$I_i = 1 - H_i. \qquad (1)$$

В системах машинного обучения с подкреплением [3] подобный параметр (1) называют функцией ценности состояния. В нашем случае эта числовая величина определяет величину вознаграждения, на которое может рассчитывать обучающийся.

В начале обучения, когда энтропия деятельности обучающегося высока, недостаток внутренней информации компенсирует регулятор *Tr@cK*. Чем больше обучающийся накопил информации о способах решения задачи, тем меньше он нуждается в дополнительной (внешней по отношению к нему) информации. В этом случае *Tr@cK* ограничивает функционирование датчика «Расстояние до цели», а на завершающем этапе обучения отключает его.

Таким образом, показателем эффективности функционирования системы «обучающийся —

проблемная среда» следует считать такой параметр, который отражал бы и состояние структуры системы действий обучающегося, и параметры проблемной среды, при которых осуществлялась деятельность. Такая мера должна отражать уровень самостоятельности обучающегося.

Следуя работе [4], возьмем в качестве показателя эффективности функционирования обучающегося в проблемной среде при выполнении i-го задания коэффициент обратной связи, который с учетом двух контуров обратной связи принимает вид

$$R_i^T = P_A^i P_B^i + H_i, \qquad (2)$$

где $P_A^i = \frac{N_1}{N_0}$ — доля неправильных действий (N_1 — количество неправильных действий; N_0 — общее количество действий); P_B^i — относительная частота включения датчика «Расстояние до цели»; H_i — энтропия деятельности обучающегося при выполнении i-го задания. Индекс T в обозначении коэффициента обратной связи (указывает количество затраченного на обучение времени на момент завершения выполнения i-го задания) позволяет рассматривать его как в масштабе выполненных заданий, так и по затраченному времени.

Целью функционирования системы *Tr@cK* является достижение коэффициентом обратной связи нулевого значения. Это означает, что действия обучающегося не зависят от датчиков проблемной среды и определяются только собственной системой управления, т. е. мозгом, на основе внутренней информации. При этом отсутствует неопределенность при выборе действия, и каждое действие приближает решение задачи.

Мера рассогласования между требуемой и реальной деятельностью обучающегося — значение функции ценности состояния — представлена в проблемной среде дискретным датчиком, отображающим систему уровней в диапазоне от 1 до 10. Благодаря этому датчику обучающийся имеет возможность осуществлять саморегулирование своей учебной деятельности.

Проблемная среда выступает в роли регулятора учебной деятельности *Tr@cK*. Она связана с обучающимся двумя линиями связи — прямой линией передачи управляющих сигналов от проблемной среды к обучающемуся и линией обратной связи, передающей в проблемную среду информацию о действительном состоянии деятельности обучающегося.

Система управления через датчик «Расстояние до цели» содействует обучающемуся в снятии структурного дисбаланса, обусловленного отрицательной обратной связью между деятельностью обучающегося и проблемной средой. Для этого в системе *Tr@cK* имеется модуль, который обеспечивает через датчик «Расстояние до цели» положительную обратную связь между множеством правильных действий и отрицательную обратную связь с множеством неправильных действий.

Положительная обратная связь, реализуемая системой *Tr@cK*, поддерживает (усиливает) правильные действия обучающегося, а отрицательная обратная связь угнетает неправильные. По мере научения относительная частота правильных действий возрастает, т. е. деятельность обучающегося становится самодостаточной и не нуждается во внешнем подкреплении. Благодаря этому потребность в датчике «Расстояние до цели» снижается и вероятность его подключения уменьшается.

Диагностика учебной деятельности

Примером реализации изложенных принципов управления является пазловая проблемная среда «Динамические пазлы», с помощью которой проводилась диагностика учебной деятельности по конструированию пространственного объекта из фрагментов [5]. Задание состоит в сборке чертежа из 25 фрагментов. Обучающийся может совершать три вида действий: 1 — просмотр фрагментов в специальном окне; 2 — установка выбранного фрагмента на рабочее поле; 3 — отмена установленного ранее фрагмента. Несмотря на то что задание одно и то же и потенциально проблемная среда одинакова для всех обучающихся, реальная проблемная среда зависит как от их поведения (учебной деятельности), так и от личности обучающегося. Это определяется тем, что поведение обучающегося является саморегулируемым и взаимосвязанным с проблемной средой и личностными особенностями обучающегося.

Научение проходит в итеративном режиме, т. е. обучающиеся повторяют конструирование объекта до тех пор, пока их деятельность не станет безошибочной. При выполнении первого задания обучающийся осуществляет конструирование, незамедлительно получая информацию о правильности или неправильности каждого действия.

Функции вознаграждения и базовые когнитивные функции мозга

Представляет интерес сопоставление стратегии действий обучающегося с состоянием развития его базовых когнитивных функций мозга (БКФМ). К таким функциям относятся: распознавание, дифференцирование, направленное внимание, объем оперативной памяти и скорость

обработки информации. Проследим, какое отражение находят психические процессы, характеризуемые БКФМ, в деятельности обучающегося, осуществляющего конструирование пространственных объектов в проблемной среде.

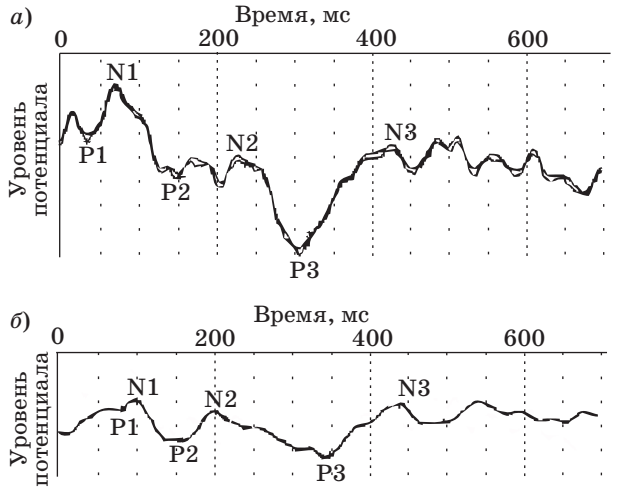
Проведенный в ходе исследования эксперимент состоял в сопоставлении результатов психофизиологического обследования обучающихся методом когнитивных вызванных потенциалов P300 и результатов, полученных при обработке протоколов деятельности обучающихся в проблемных средах.

Вызванные потенциалы (ВП) являются индикаторами электрических процессов работы мозга, связанных с механизмами восприятия информации, ее обработки. Одной из таких методик, значительно продвинувших анализ этих процессов, является методика когнитивных вызванных потенциалов (КВП), или методика P300. Этот вид ВП в последнее время все больше находит применение в клинической практике при оценке доклинической стадии когнитивных нарушений различного типа [6].

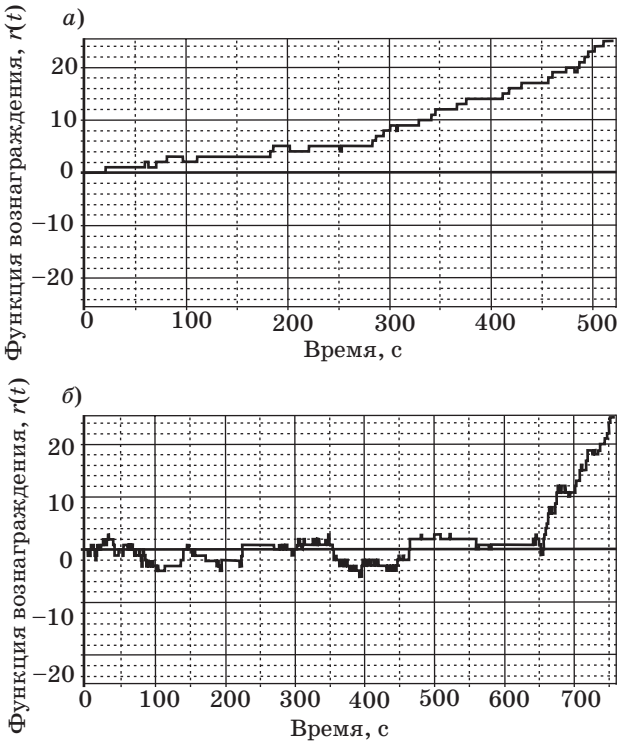
Сущность этой методики заключается в том, что не просто выделяются ответные реакции на тот или иной стимул, связанные с приходом афферентации, а анализируются эндогенные события, происходящие в мозгу, связанные с опознанием стимула, его дифференциацией, удержанием в памяти и пр., — все, что создает сущность когнитивных процессов.

В ходе эксперимента нейрофизиологическая диагностика БКФМ была проведена в группе студентов из 63 чел. Средний возраст составил 17–18 лет. После анализа ВП были выделены три группы: 1) в группе из 36 чел. (57 % от числа обследованных) показатели соответствовали норме, что свидетельствует о том, что процессы опознания, дифференцировки, направленного внимания и объем оперативной памяти не страдают; 2) в группе из 15 чел. (24 % от числа обследованных) отмечалось нарушение процессов направленного внимания и снижение объема оперативной памяти; 3) в группе из 12 чел. (19 % от числа обследованных) отмечались нарушения ответа в виде удлинения пика P3 и слабо выраженного пика N2, что свидетельствует не только о снижении объема оперативной памяти и направленного внимания, но и о нарушении процессов опознания и дифференцировки. Представлены диаграммы обследования КВП P300 для характерных представителей первой (обучающийся № 1, рис. 2, *а*) и третьей (обучающийся № 2, рис. 2, *б*) групп.

Из рисунка видно, что обучающиеся первой и третьей группы сильно отличаются по форме и характеристикам ВП. Проследим, какие отличия имеют место в способах осуществления деятельности в проблемной среде для этих обучающихся. Графически учебная деятельность, регулируемая системой *Tr@cK*, в проблемной среде для первого выполнения задания обучающимся № 1 и обучающимся № 2 представлена на рис. 3, *а*, *б* функциями вознаграждения $r(t)$. Функция вознаграждения задает отображение каждого действия в числовую меру, определяющую степень эффективности принятия действия в данном со-



■ ***Рис. 2.*** *Диаграммы обследования КВП P300: а — обучающийся № 1; б — обучающийся № 2*



■ ***Рис. 3.*** *Функция вознаграждения при первом выполнении задания в масштабе времени: а — обучающийся № 1; б — обучающийся № 2*

стоянии проблемной ситуации для достижения цели. Она определяет сиюминутную эффективность пары «действие — состояние проблемной среды», а достижение решения задачи соответствует максимуму общего вознаграждения.

Функция вознаграждения вычисляется из обработки данных синтаксической информации (см. выше) о действиях обучающегося и представляет траекторию его деятельности. Значение $r(t)$ увеличивается на единицу, если совершено правильное действие, и уменьшается на единицу, если — неправильное. На рис. 3 представлены функции вознаграждения при выполнении первого задания: максимальное вознаграждение равно количеству исходных фрагментов изображения.

Из сравнения траекторий деятельности видно, что обучающийся № 2 совершает гораздо больше неправильных действий по сравнению с обучающимся № 1. Исходя из того, что при выполнении первого задания $P_B^1 = 1$ и коэффициент обратной связи зависит только от доли неправильных действий, его значение говорит лишь о том, что обучающийся № 1 справился с заданием более успешно. Для определения особенностей функционирования системы «обучающийся — проблемная среда» следует рассматривать последовательность значений функции ценности состояния.

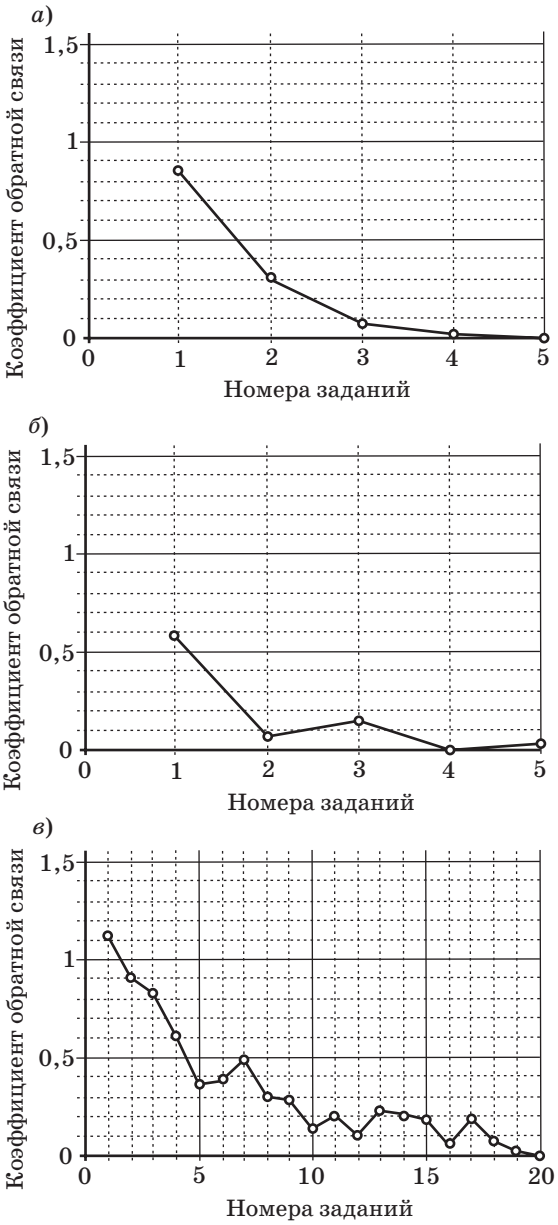
Динамика изменения коэффициента обратной связи

Представим графически особенности функционирования регулятора *Tr@cK* для нескольких обучающихся (звено 8 на рис. 1), отличающихся друг от друга способами осуществления учебной деятельности в проблемной среде и уровнем развития БКФМ. Построим график изменения коэффициента обратной связи (2) в масштабе выполненных заданий (рис. 4).

Для большинства обучающихся от задания к заданию P_A^i уменьшается, что делает структуру системы действий более совершенной, т. е. функция ценности состояния обучающегося возрастает, а значение энтропии деятельности — второго слагаемого в уравнении (2) — убывает. По мере научения недостаток внешней помощи (уменьшение P_B^i, затем отключение датчика «Расстояние до цели») компенсируется накопленными знаниями и деятельность обучающегося перестает нуждаться в регулировании. Такому положению вещей соответствует уменьшение коэффициента обратной связи до нуля (см. рис. 4, *а*).

Некоторая часть обучающихся, успешно осуществляющих деятельность при повышенной частоте P_B^i (датчик «Расстояние до цели» компенсирует внутреннюю неопределенность), при уменьшении частоты подкрепления совершают больше ошибочных действий, и проблемная среда увеличивает P_B^i при выполнении следующего задания. Происходит колебание показателя общей эффективности функционирования регулятора *Tr@cK* R_i^T (см. рис. 4, *б, в*).

В эксперименте все обучающиеся достигают десятого уровня (безошибочной деятельности в отсутствие подкрепления). Но графики изменения коэффициента обратной связи показывают, насколько разным может быть процесс научения



■ ***Рис. 4.*** *Функция ценности состояния в масштабе выполненных заданий: а — обучающегося № 1; б — обучающегося № 3; в — обучающегося № 2; ○ — выполненные задания*

решению задач в проблемной среде. Сопоставляя данные нейрофизиологической и компьютерной диагностики следует отметить, что обучающийся № 1 относится к первой группе (когнитивные функции мозга достаточно развиты), обучающийся № 3 — ко второй, а обучающийся № 2 — к третьей группе.

Выводы

Предлагаемая система автоматического управления учебной деятельностью, состоящая из проблемной среды, включающей систему автоматического регулирования *Tr@cK*, снимает структурный дисбаланс между желанием обучающегося обучиться решению проблемы и несовершенством структуры его системы действий.

Сравнительный анализ данных нейрофизиологической диагностики БКФМ с данными компьютерной диагностики учебной деятельности обучающихся показал, что одной из причин возникающих проблем в обучении студентов является недостаточный уровень развития БКФМ.

Система автоматического управления учебной деятельностью содействует процессу саморегуляции деятельности обучающегося, количественно определяет функции вознаграждения, позволяет измерить скорости изменения функции ценности состояния (обучаемость). Эта информация может использоваться при индивидуализации обучения.

Литература

1. **Бесекерский В. А., Попов Е. П.** Теория систем автоматического управления. 4-е изд., перераб. и доп. — СПб.: Профессия, 2003. — 752 с.
2. **Зимняя И. А.** Педагогическая психология: учеб. пособие. — Ростов н/Д.: Феникс, 1997. — 480 с.
3. **Рассел С., Норвиг П.** Искусственный интеллект: современный подход. — М.: Вильямс, 2006. — 1408 с.
4. **Светлов В. А.** Конфликт: модели, решения, менеджмент. — СПб.: Питер, 2005. — 540 с.
5. **Шадрин И. В.** Инструментальный метод исследования деятельности обучающихся конструированию пространственных объектов//Системы управления и информационные технологии. 2008. № 2.2(32). С. 308–311.
6. **Дроздова Л. Н., Дьячук П. П.** Диагностика динамики когнитивных стратегий поиска решения задач и когнитивных функций мозга студентов в процессе обучения // Competences and teacher competence: Proc. Conf., Osijek, 18–19 April 2007. P. 168–175.

УДК 681.2; 615.47

БИОТЕХНИЧЕСКАЯ СИСТЕМА ДЛЯ ИССЛЕДОВАНИЯ ИНТЕЛЛЕКТУАЛЬНОЙ ДЕЯТЕЛЬНОСТИ ЧЕЛОВЕКА

Н. Б. Суворов,
доктор биол. наук, профессор
НИИ экспериментальной медицины Северо-Западного отделения РАМН
В. А. Абрамов,
инженер
А. В. Козаченко,
канд. техн. наук, доцент
Санкт-Петербургский государственный университет информационных технологий, механики и оптики
Ю. З. Полонский,
доктор биол. наук, старший научный сотрудник
Институт мозга человека им. Н. П. Бехтеревой РАН

Разработана и испытана в реальных исследованиях биотехническая система для изучения психофизиологических механизмов напряженной интеллектуальной деятельности (во время игры в шахматы). Участвовали шахматисты высшей квалификации (коэффициент Эло ≥ 2300). При разработке решена главная задача: психофизиологические параметры играющего с шахматной программой синхронизированы с текущей позицией на шахматной доске.

Ключевые слова *— интеллектуальная деятельность, психофизиологические параметры, шахматисты.*

Введение

В настоящее время в медицинской диагностике, в задачах комплексной оценки функционального состояния человека в системах управления при минимальной физической нагрузке, в исследованиях умственной деятельности, при решении интеллектуальных операторских задач широко применяются биотехнические системы (БТС) различного назначения. Одним из наиболее трудоемких и ответственных процессов при диагностике, прогнозе текущего и последующего состояний является анализ и формулирование научно-практических выводов из комплекса психофизиологических показателей. Эффективность этого процесса зависит, в частности, от состава БТС и объема аналитических возможностей. Аппаратно-программные средства обеспечивают быструю и достоверную переработку больших объемов научной информации, получаемой в исследовании.

Использование различных средств и методов анализа, реализуемых электронными устройствами, позволяет существенно расширить пределы медицинского и психофизиологического обследования и заметно уменьшить вероятность ошибки при оценке состояния оператора в реальном времени. Наибольшее распространение имеют биотехнические системы, включающие такие аппаратно-программные средства, как тренажеры, имитаторы пультов управления и др., моделирующие ту или иную деятельность, связанную с различными манипуляциями, зрительно-моторным слежением, наблюдением за информационными табло, приводящим в свою очередь к состояниям монотонии и опасности засыпания и т. п. [1]. БТС для исследования умственных нагрузок также используют модельные ситуации — от простых арифметических или алгебраических задач до достаточно сложных тестов на распознавание образов в условиях действия помех, решения логических или творческих задач, предъявляемых испытуемому. Однако в настоящее время практически нет технических средств, направленных на изучение психофизиологических характеристик человека во время реальной интеллектуальной деятельности, а не на модели последней.

Целью работы являлась разработка и испытание БТС для изучения психофизиологических механизмов напряженной интеллектуальной деятельности. Реализация этой цели потребовала выбора аппаратуры и программного обеспечения; разработки структуры БТС и алгоритма ее работы; синхронизации психофизиологических параметров с шахматной партией; испытания в реальных исследованиях с шахматистами высшей квалификации.

Методические подходы, реализованные в биотехнической системе

Интеллектуальная деятельность человека является одной из его специфических особенностей и наиболее сложно организованных психических функций. Инструментальные исследования в изложенной постановке ранее не проводились, поэтому разработанный комплекс является в своем роде уникальным. Под напряженной интеллектуальной деятельностью подразумевается реальная шахматная игра с современной компьютерной программой. В исследованиях участвовали молодые шахматисты высокой квалификации г. Санкт-Петербурга — элитные гроссмейстеры, мастера спорта с высоким рейтингом Эло. Профессионализм участников позволил проводить шахматные партии вслепую с закрытыми глазами, что дало возможность минимизировать помехи при регистрации электроэнцефалограммы (ЭЭГ — 21 канал), электрокардиограммы (ЭКГ с предплечий), кардиоритмограммы (КРГ). Подобную совокупную регистрацию комплекса электрофизиологических параметров обеспечивает электроэнцефалограф «Мицар — ЭЭГ—202 (24+8)», имеющий полосу пропускания от 0 (DC) до 150 Гц и диапазон измерений до 300 мВ (разработчик и производитель ООО «Мицар», Санкт-Петербург, сертификат соответствия № РОСС RU.ИМ17.В00017). Помимо этого фиксировались функция дыхания — пневмограмма (ПГ) и голоса шахматиста, сообщающего свой ход, и «транслятора», сообщающего ход, сделанный шахматной программой (использовались штатный датчик дыхания и специальные микрофоны фирмы Panasonic). Проводилось также психологическое тестирование (тест Люшера).

Шахматная программа, используемая в описываемой БТС, должна быть адекватна мастерству участвующих шахматистов или превосходить их. Для синхронизации и сопоставления шахматной партии с психофизиологическими характеристиками в каждый момент времени необходимо знать, сколько времени потрачено на каждый ход в отдельности и полный протокол партии, из которого известно, сколько времени прошло от начала партии до конкретного хода. Перечисленным условиям удовлетворяет программа Deep Fritz 11 4CPU [2, 3]. В ней предусмотрена гибкая система задания контроля времени, диапазон по рейтингу 900–3000 (максимальный рейтинг Эло участвовавших шахматистов составлял 2711), поэтому шахматную квалификацию программы можно варьировать — равные силы, незначительно сильнее или слабее, намного сильнее или слабее шахматиста. Некоторая коррекция возможна также путем изменения контроля времени в ту или иную сторону. Протокол партии программы Fritz 11 оказался неприемлемым для временной оценки сделанных ходов, поэтому была написана дополнительная программа, которая демонстрировала время, затраченное на ход и накопленное от начала партии. Минимальные системные требования: Pentium II 300 МГц, RAM 64 Мб, 1,5 Гб свободного места на жестком диске, Windows 2000/XP/Vista.

Описание биотехнической системы

В состав разработанной БТС входят следующие основные блоки.

Главный компьютер — для синхронного отображения на мониторе в реальном времени всех происходящих «событий».

В качестве блока регистрации использовали электроэнцефалограф «Мицар — ЭЭГ—202 (24+8)». ЭЭГ регистрировалась электродами, расположенными на голове испытуемого. Сигналы ЭЭГ от блока усилителей в цифровой форме через гальваническую развязку поступают в главный компьютер и отображаются на экране монитора.

Электрокардиограмма регистрировалась через один из полиграфических каналов электроэнцефалографа. Электрокардиосигнал отображается на том же мониторе.

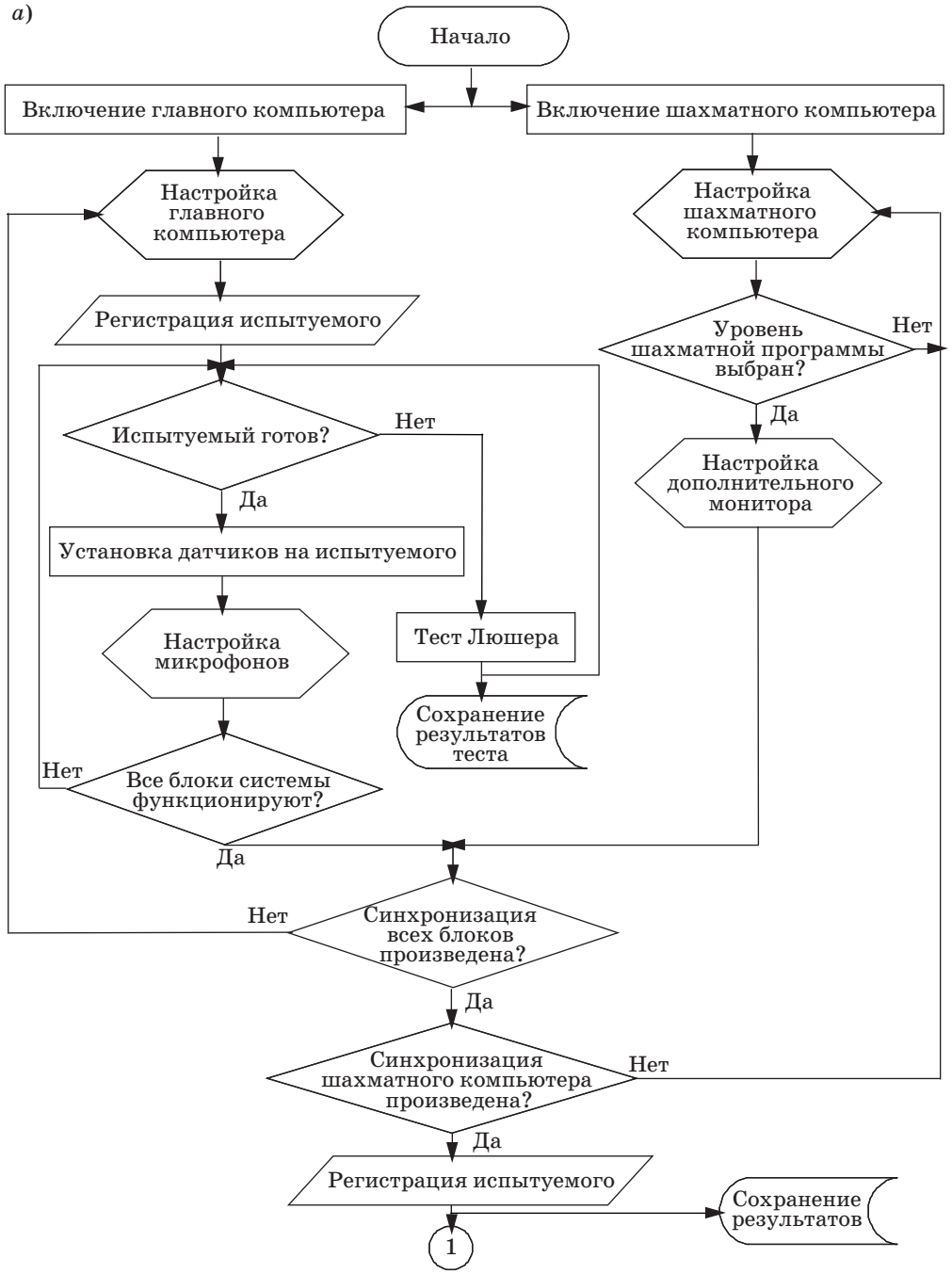
Дыхание — ПГ регистрировалась с помощью носового (назального) терморезисторного датчика, подсоединяемого через один из полиграфических каналов с отображением на экране.

Аудиосигналы от микрофонов «транслятора» и испытуемого через полиграфические каналы в виде меток также выносятся на монитор вместе с остальными параметрами.

В состав БТС входит также блок обработки информации, созданный на базе программных комплексов WinEEG, WinHRV.

Шахматный компьютер.

Фотостимулятор — прибор, генерирующий световые сигналы с заданными параметрами (интенсивностью, длительностью, частотой и т. п.) для воздействия на зрительный анализатор. Светодиодный фотостимулятор способен генерировать частоту фотостимуляции от 1 до 50 вспышек/с с шагом 1 вспышка/с.

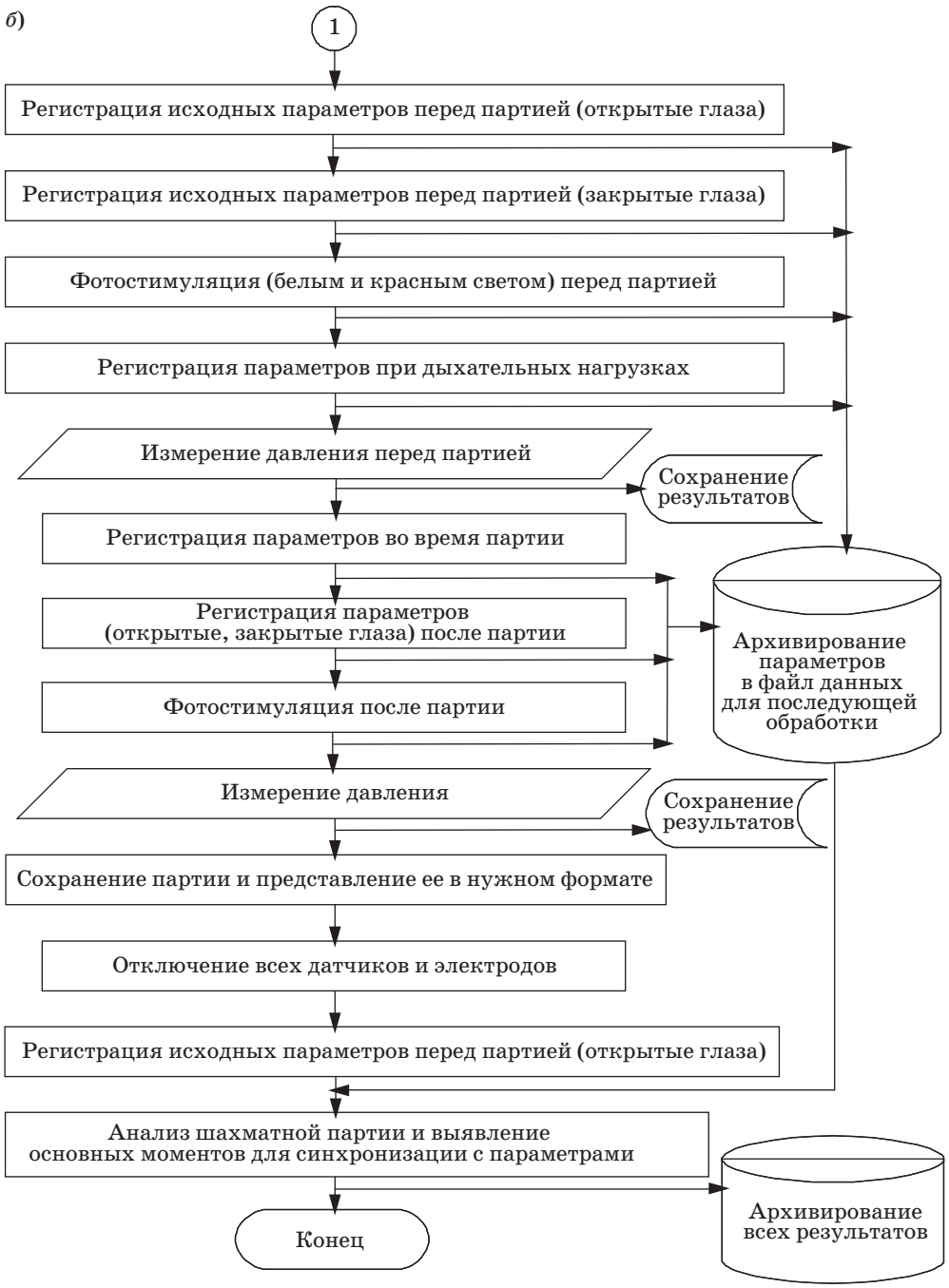


■ ***Рис. 1.*** *Блок-схема проведения испытаний биотехнической системы*: *а — подготовительный этап*;

Испытуемый — профессиональный шахматист, способный вести партию вслепую на высоком уровне.

«Транслятор» — опытный шахматист, обеспечивающий передачу информации между игроком и шахматным компьютером, умеющий обращаться с шахматной программой, понимать шахматную нотацию, быстро и четко передавать ходы, сделанные компьютером, и вводить в компьютер ходы, сделанные игроком, с минимальной задержкой. От «транслятора» зависит величина отставания видеоизображения от остальных электрофизиологических параметров на мониторе.

Дополнительный монитор подключен на шахматный компьютер, обеспечивает необходимое экранное разрешение для визуализации шахматной партии.



б — этапы измерений и завершения исследования

Видеокамера направлена на дополнительный монитор, служит для передачи шахматной позиции на монитор главного компьютера, обеспечивает синхронизацию психофизиологических параметров и шахматной партии. Видеоизображение шахматной доски с текущей позицией воспроизводится синхронно с выводом на экран монитора соответствующих участков всех регистрируемых параметров испытуемого или может быть отключено.

Испытания БТС состояли из трех этапов (рис. 1, *а*, *б*).

Последовательность действий на подготовительном этапе.

Настройка главного и шахматного компьютеров.

Настройка шахматной силы программы Deep Fritz 11. В зависимости от задачи это: выбор ее силы, выбор контроля времени программы и ис-

пытуемого. В связи с тем, что игра проходила вслепую, необходимо было компенсировать время, уходившее на устные сообщения между шахматным компьютером и игроком. Поэтому было принято решение добавлять игроку несколько секунд после каждого хода. Кроме того, в настройках программы предусмотрена возможность получить играющим преимущество во времени (гандикап).

Подсоединение к шахматному компьютеру дополнительного монитора для визуализации шахматной партии через видеокамеру (настраивается фокус, угол наклона камеры). Это необходимо для синхронизации позиции и психофизиологических параметров.

Регистрация главных персональных данных испытуемого (шахматиста).

Оценка текущего психологического состояния испытуемого. Цветовая диагностика Люшера позволяет измерить стрессоустойчивость, активность, коммуникативные способности и другие характеристики на момент проведения теста.

Установка датчиков на испытуемого, проверка качества их установки. На протяжении всего исследования игрок сидел в удобном кресле, позволявшем ему полностью расслабиться.

Настройка чувствительности и частотных фильтров электроэнцефалографа и полиграфических каналов.

Настройка микрофонов игрока и «транслятора».

Последовательная проверка всех блоков системы.

Синхронизация отдельных блоков. Она занимает важное место в данном исследовании. Необходимо представить на мониторе главного компьютера одномоментные показания ЭЭГ, ЭКГ, ПГ, сигналов микрофонов, видеоизображения и др. При полной синхронизации перечисленных параметров можно начинать исследование.

Последовательность действий во время основного этапа (этапа измерений).

Регистрация исходных (фоновых) психофизиологических показателей перед партией для их последующего сравнения во время партии, после партии и т. д.

Проведение перед партией стандартных для электрофизиологического исследования тестов — фотостимуляция и дыхательная нагрузка.

Измерение артериального давления (АД) перед партией.

Регистрация психофизиологических показателей во время партии — самая продолжительная часть исследования.

Фоновая запись при открытых и закрытых глазах после игры.

Измерение АД после партии.

Последовательность действий на этапе завершения исследования и анализа данных.

Сохранение (архивирование) всех данных, включая шахматную партию, и представление ее в нужном формате.

Отключение всех датчиков. Отключение всех блоков.

Анализ шахматной партии и выявление наиболее важных моментов для сопоставления с соответствующими фрагментами комплекса психофизиологических параметров.

Обработка данных и заключение по результатам испытаний БТС.

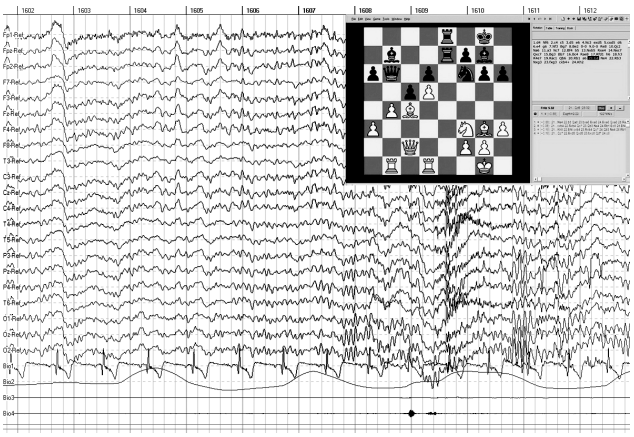
Фрагмент записи полного комплекса регистрируемых показателей на мониторе главного компьютера представлен на рис. 2.

Биотехническая система позволяет анализировать электрокардиосигнал путем построения КРГ. В анализ входит: построение гистограмм распределения RR-интервалов, скаттерограмм, спектров мощности и вычисление ряда производных параметров. Часть расчетных параметров сведена в таблицу.

Из таблицы видно, что большая часть параметров во время принятия решения на 699-й секунде после начала партии не изменилась по сравнению с 341-й секундой (начало обдумывания). Исключение — спектральные характеристики.

Изменения ЭЭГ во время шахматной партии демонстрирует рис. 3, *а*, *б*.

Анализ фрагментов рис. 3 показывает, что спектральная мощность системообразующего альфа-ритма частотой около 9 Гц на фрагментах А (фон) в 2,5 раза выше, чем во время принятия

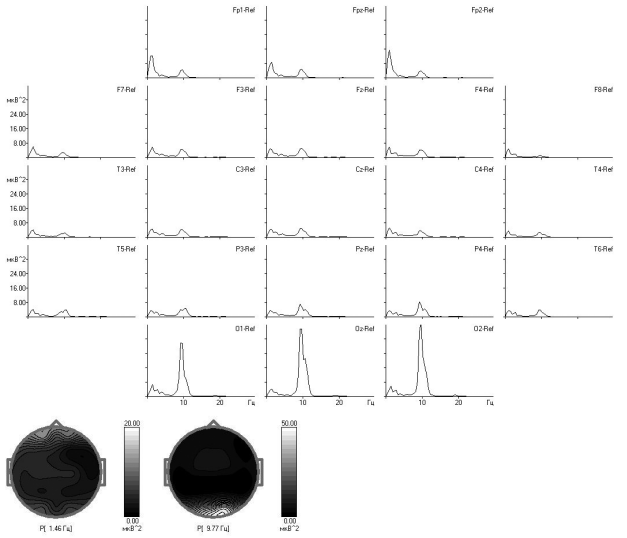


■ ***Рис. 2.*** *Электроэнцефалограмма (21 канал) мастера ФИДЕ А. во время обдумывания и принятия решения о 21-м ходе белых b4 (1609-я секунда партии): Bio1 — ЭКГ; Bio2 — кривая дыхания; Bio3 — отметка сообщения «транслятора» о ходе, сделанном программой; Bio4 — отметка сообщения шахматиста о сделанном ходе*

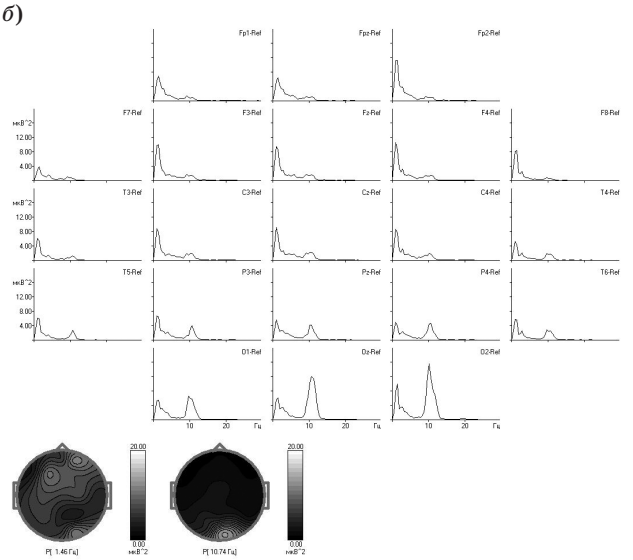
Расчетный параметр	341-я секунда	699-я секунда
SDNN, мс	51,7	49,5
pNN50, %	2,3	2,2
M, мс	710	707
ЧСС, уд/мин	84	85
CV, %	7,28	7,00
RRmin, мс	604	606
RRmax, мс	834	856
MODE, с	0,70	0,70
AMO, %	36,2	37,5
X, с	0,23	0,25
ИВР, %/с	157,2	150,0
ИН, %/с^2	112,3	107,1
Total, мс^2	1493	1454
LF, мс^2	1040	654
LFnorm, %	83,6	80,6
HF, мс^2	204	158
HFnorm, %	16,4	19,4
LF/HF	5,10	4,14
N	213	224

SDNN — стандартное отклонение RR-интервалов;
PNN50 — число соседних пар RR-интервалов, отличающихся более чем на 50 мс;
M — средняя длительность RR-интервалов;
ЧСС — частота сердечных сокращений;
CV — коэффициент вариабельности;
RRmin — минимальное значение RR-интервала;
RRmax — максимальное значение RR-интервала;
MODE — мода распределения RR-интервалов;
AMO — амплитуда моды;
X — вариационный размах (RRmax — RRmin);
ИВР — индекс вегетативного равновесия;
ИН — индекс напряжения регуляторных систем;
Total — мощность колебаний RR-интервалов в диапазоне 0,003–0,4 Гц;
LF — мощность медленных колебаний RR-интервалов 0,04–0,15 Гц;
LFnorm — нормированная мощность медленных колебаний RR-интервалов;
HF — мощность быстрых колебаний RR-интервалов 0,15–0,4 Гц;
HFnorm — нормированная мощность быстрых колебаний RR-интервалов;
LF/HF — отношение LF к HF;
N — число RR-интервалов.

а)



б)



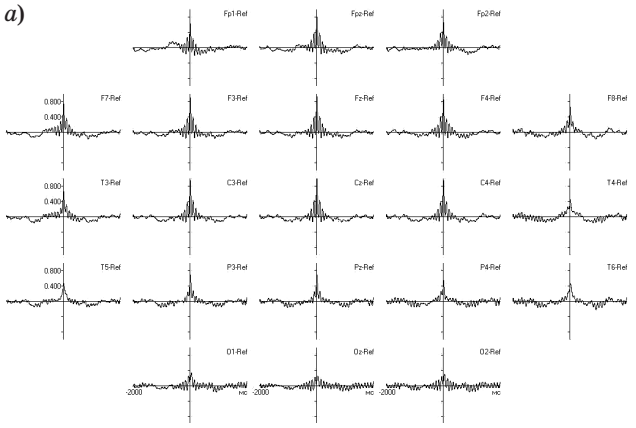
■ ***Рис. 3.*** *Спектры мощности ЭЭГ на разных этапах исследования: а — расслабленное бодрствование, глаза закрыты (фон до партии); б — непосредственно перед 13-м ходом после 11-минутного обдумывания (принятие решения). На графиках спектров по абсциссе — частота колебаний ЭЭГ, по ординате — мощность. На топограммах слева — распределение 1-й и 2-й гармоник по поверхности мозга, справа — шкала спектральной мощности*

решения после обдумывания хода (Б), его частота на этих фрагментах превышает 10 Гц — свидетельство умственного напряжения.

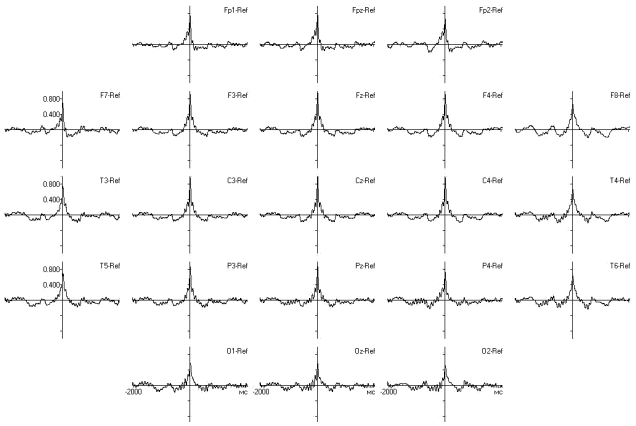
Кросскоррелограммы (рис. 4) демонстрируют наличие организованной ритмической (альфаритм) ЭЭГ в фоновом состоянии (рис. 4, *а*) и значительное изменение ритмической структуры ЭЭГ перед принятием решения в пользу медленных дельта- и тета-ритмов (рис. 4, *б*). Последний, возникающий при разнообразных нагрузочных пробах, часто называют ритмом напряжения.

Изложенные выше далеко не полные психофизиологические данные, полученные с помощью разработанной биотехнической системы, являются коррелятами интеллектуальной деятельности и инструментом для специалистов в обла-

а)



б)



■ ***Рис. 4.*** *Кросскоррелограммы отведений ЭЭГ: а и б соответствуют тем же фрагментам ЭЭГ, что и на рис. 3, а, б*

сти психонейрофизиологии для изучения тонких механизмов активных мыслительных процессов. В данной статье не приведены КРГ, функции когерентности в графическом и топографическом отображении, авто- и кросскорреляционные функции, диаграммы пространственно-временного взаимодействия 21 структуры головного мозга, координаты максимальной плотности распределения мозговых источников токов, взаимодействие КРГ и функций дыхания, не рассмотрены индивидуальные особенности шахматного творчества нескольких гроссмейстеров и мастеров Санкт-Петербурга и др.

Заключение

Разработан комплекс, предназначенный для регистрации и количественной оценки ряда психофизиологических характеристик шахматистов высшей квалификации во время игры в шахматы с использованием шахматной программы. Разработанная БТС испытана в качестве инструмента для психофизиологических исследований механизмов напряженной интеллектуальной деятельности. В соответствии с целью и задачами решена проблема синхронизации всех этапов и элементов системы — текущая позиция на шахматной доске в любой момент времени соответствует временно́му срезу психофизиологических параметров. Динамика обдумывания, принятия решений, осознанные и неосознанные ошибки, просмотры и т. п. характеризуются комплексом этих параметров, на основании которых специалисты в состоянии сделать определенные выводы. Проведены испытания БТС в реальных электро- и психофизиологических исследованиях шахматистов высшей квалификации.

Результаты, получаемые с помощью разработанной БТС, дают возможность до ответственных турниров оценить состояние шахматиста, функциональный резерв (его уровень готовности), после — понять степень утомления и определить время и мероприятия, необходимые для восстановления. Кроме того, специальный интерес представляют психофизиологические реакции шахматиста на ошибочные тактические и стратегические решения, цейтнот, «зевки» и т. п.

Научные результаты, полученные на разработанном и испытанном комплексе, могут иметь большое значение для нейрофизиологов, занимающихся изучением нераскрытых психофизиологических механизмов интеллектуальной деятельности, а также для специалистов в области искусственного интеллекта.

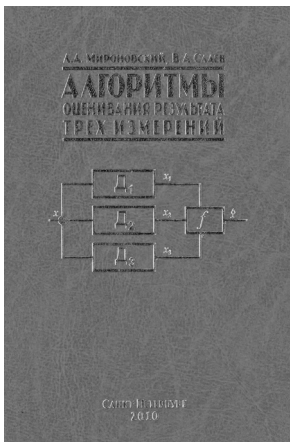
Перспективы развития комплекса: разработка системы безманжетной регистрации АД позволит проводить непрерывный на протяжении всего исследования контроль АД, полученные данные будут синхронно отображаться вместе с остальными параметрами; для обеспечения возможности игры с открытыми глазами необходимо разработать комплекс программных средств для подавления помех при открытых глазах и движениях рук; замена шахматной игры моделью любого другого вида деятельности делает разработанную БТС универсальной: она может использоваться для исследования текущего и «рабочего» функционального состояния операторов информационных систем управления, диспетчеров управления воздушным движением, операторов различных транспортных средств, военной техники и т. д. Результаты подобных работ необходимы для оперативного отбора и прогноза качества деятельности человека в системе управления.

Биотехническая система внедрена и используется в исследованиях лаборатории нейроэкологии НИИ экспериментальной медицины РАМН и лаборатории стереотаксических методов Института мозга человека им. Н. П. Бехтеревой РАН.

Исследования поддержаны грантом научной программы Санкт-Петербургского научного центра РАН «Исследование возможностей психофизиологической поддержки лиц, занятых непрерывной напряженной интеллектуальной деятельностью — шахматистов высшей квалификации» за 2009 год и выполнялись при содействии шахматной федерации Санкт-Петербурга.

Литература

1. **Суворов Н. Б., Мясников А. В., Попечителев Е. П.** Аппаратная часть биотехнического комплекса для исследования кардиореспираторного взаимодействия // Изв. СПбГЭТУ «ЛЭТИ». Сер. Биотехнические системы в медицине и экологии. 2004. Вып. 2. С. 38–42.
2. **Fritz 11**: Обзор статей об игре Fritz 11. http://gameguru.ru/articles/525/view.html (дата обращения: 17.01.2010).
3. **Chessmaster**: Grandmaster Edition / Chessmaster 11: Grandmaster Edition. http://tav.su/9046-chessmaster-grandmaster-edition-akella.html (дата обращения: 17.01.2010).



Мироновский Л. А., Слаев В. А. Алгоритмы оценивания результата трех измерений. — СПб.: «Профессионал», 2010. — 192 с.: ил. ISBN 978-5-91259-041-2, УДК 389.

Монография состоит из пяти глав и трех приложений. В ней собраны, классифицированы и проанализированы алгоритмы оценивания, направленные на решение «задачи о трех измерениях».

В Главе I приведена классификация погрешностей измерений, а также методов оценивания, оптимизирующих выбранные критерии. Эти методы по виду критериев подразделяются на вероятностные, детерминированные, эвристические и диагностические. Описаны классические средние оценки и их свойства.

Глава II посвящена вероятностному и детерминированному подходам к оцениванию. В ней рассмотрены оценки максимального правдоподобия, марковские, байесовские, квадратические, модульные и степенные оценки, а также оценки, оптимизирующие составные и комбинированные критерии.

Глава III описывает принципы эвристического оценивания, основанные на математическом определении средних величин по Коши и Колмогорову. На этом пути строятся классические средние, линейные, квазилинейные, а также разностные квазилинейные и нелинейные оценки.

В Главе IV рассматриваются диагностические методы получения оценок, основанные на применении алгебраических инвариантов. Наличие алгебраических инвариантов позволяет осуществить отбраковку искаженных измерений методами технической диагностики по минимальному или максимальному рассогласованию. Алгоритмы оценивания скалярной величины по трем измерениям сведены в таблицу, в которой отражено более семидесяти различных оценок.

Глава V касается применения средних оценок для фильтрации сигналов. Охарактеризован принцип использования «гладкости» сигналов для борьбы с погрешностями, применение которого приводит к фильтрам с конечной памятью. Описаны медианные и диагностические фильтры, приведен пример фильтрации навигационной информации.

В Приложения вынесены современная терминология по характеристикам точности, соотношение между неопределенностями и характеристиками погрешности, а также статистические свойства получаемых оценок.

Для метрологов, приборостроителей и разработчиков алгоритмов, реализуемых в программно управляемых средствах измерений, а также для экспертов, осуществляющих их аттестацию. Может быть полезна студентам и аспирантам технических вузов.

Книгу можно приобрести за наличный и безналичный расчет во ВНИИМ им. Д. И. Менделеева: 190005, Санкт-Петербург, Московский пр., 19; контактный телефон +7 (812) 323-93-79;
e-mail: abl@bi10.vniim.ru, Любомиров Андрей Борисович. Цена экземпляра 413 руб.

УДК 615.471:617.7

АВТОМАТИЧЕСКОЕ ВЫДЕЛЕНИЕ УЧАСТКОВ ЭЛЕКТРОКАРДИОСИГНАЛА С НОРМАЛЬНЫМ СИНУСОВЫМ РИТМОМ

***В. М. Бахилин*[1],**
научный сотрудник
Санкт-Петербургский научно-исследовательский институт уха, горла, носа и речи

Предложен самонастраивающийся алгоритм выделения фрагментов с нормальным синусовым ритмом в длинных записях ЭКГ, численные параметры которого рассчитаны по результатам анализа записей Физиобанка (PhysioBank).

Ключевые слова *— автоматический анализ ЭКГ, вариабельность сердечного ритма, вариабельность RT-, TP-, PR- и PT-интервалов.*

Введение

Массовое обследование населения холтеровским мониторированием, нацеленное на выявление ранних стадий кардиологических заболеваний с бессимптомным течением, является одной из важнейших задач современной медицины. Очевидно, что поточная обработка 24-часовых записей электрокардиосигналов невозможна без средств автоматизации. Ясно также, что полная автоматизация диагностики ритма сердца нереальна, поскольку нечеткие медицинские знания не поддаются необходимой формализации. Известно [1], что профессиональная деятельность специалиста всегда опережает возможности автоматических систем. Поэтому перспективным решением задачи медицинской диагностики по длинным записям ЭКГ является создание интерактивных систем, предполагающих участие врача в принятии решения. При построении интерактивной системы представляется целесообразным выделение участков записей с повторяющимися с заданной точностью фрагментами сердечного цикла (например, PQRST-комплексами в норме или QRST-комплексами в пароксизмах мерцательной аритмии). В результате автоматической обработки выделенных участков врачу должны предъявляться графическая информация (в виде усредненного повторяющегося фрагмента, ритмограммы и т. п.) и численные оценки разброса параметров повторяющихся фрагментов.

Одной из важнейших задач интерактивной системы обработки ЭКГ, не имеющей до настоящего времени окончательного решения, является надежное выделение участков с нормальным синусовым ритмом. Актуальность этой задачи обусловлена тем, что для большинства обследуемых (в том числе страдающих различными видами аритмий) участки с нормальным синусовым ритмом составляют основную часть записи ЭКГ. Так, более 99 % записи 105 MIT-BIH Arrhythmia Database имеет нормальный синусовый ритм, прерываемый эктопическими вентрикулярными ударами, средняя длительность участка с нормальным синусовым ритмом составляет около 6 мин. Можно ожидать, что при поточном обследовании в основном здорового населения участки с нормальным синусовым ритмом будут более продолжительными. Выделение участков с нормальным синусовым ритмом позволит, во-первых, локализовать аномальные фрагменты и, во-вторых, провести анализ и выявление патологий при сохранении синусового ритма (к которым относятся, например, синусовые брадикардия и тахикардия, ишемия и т. п.).

Основная сложность задачи автоматического выделения участков с нормальным синусовым ритмом в записях амбулаторных суточных ЭКГ

[1] Научный руководитель — доктор технических наук, заслуженный деятель науки РФ, профессор кафедры биомедицинской электроники и охраны среды Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» *Е. П. Попечителев*.

заключается в большом разбросе параметров (длительностей PR-, RR- и RT-интервалов) по множеству записей ЭКГ и нестационарности обрабатываемых процессов в рамках каждой записи. Для обработки таких записей необходимы самонастраивающиеся алгоритмы. Основой выбора параметров самонастройки, таких как длительность адаптации, ширина зон поиска и т. п., являются априорные знания. Другая сложность рассматриваемой задачи состоит в обеспечении надежного обнаружения QRS-комплексов и P- и T-волн в зашумленных записях амбулаторных суточных ЭКГ с низкой частотой дискретизации (ЧД). Многочисленные известные методы и алгоритмы обнаружения QRS-комплексов непрерывно пополняются новыми [3]. Известны различные методы автоматического обнаружения характерных точек P- и T-волн и выделения интервалов, характеризующих длительности предсердных и желудочковых систол. Большинство методов основано на предварительном выделении зон поиска относительно опорных точек — вершин R-волн (см., например, [4] и ссылки в ней).

Целью настоящей работы является разработка самонастраивающегося алгоритма выделения участков с нормальным синусовым ритмом в длинных записях ЭКГ. Первым этапом разработки является анализ записей PhysioBank [2] для извлечения априорной информации, необходимой при проектировании самонастраивающегося алгоритма. Для расчета численных параметров алгоритма анализируются записи банков Fantasia и MIT-BIH Normal Sinus Rhythm, исследуются статистические характеристики длительностей PR-, RR- и RT-интервалов по множеству записей, а также в рамках отдельных записей — временные свойства последовательностей измерений PR- и RT-интервалов как случайных рядов.

На следующем этапе разработки предлагаются: 1) новый простой эвристический алгоритм распознавания QRS-комплексов, обладающий высокой эффективностью, и 2) самонастраивающийся алгоритм автоматического обнаружения максимальных точек P- и T-волн на длительных интервалах времени, основанный на непрерывно уточняющемся вычислении зоны поиска вершин этих волн и использующий результаты первого этапа разработки.

Исходные данные и методы обработки

В качестве экспериментальных данных были использованы 50 записей ЭКГ из банков Fantasia Database (FD) и MIT-BIH Normal Sinus Rhythm Database (MB NSRD) PhysioBank [2]. Рассчитывались средние значения и среднеквадратические отклонения (СКО) интервалов RR, RT_{max}, $T_{max}P_{max}$ и $P_{max}R_{max}$ на одноминутных фрагментах записей в различное время суток (всюду далее в обозначениях интервалов индексы max опускаются: например, запись RT означает интервал между вершинами R- и T-волн). Рассчитывались также коэффициенты корреляции между изменениями во времени RR-интервалов и изменениями всех остальных перечисленных выше интервалов на HF-, LF-, VLF- и ULF-частотах. Для расчетов этих коэффициентов на VLF-частотах использовались одночасовые фрагменты записей, на ULF-частоте — полные записи. СКО и коэффициенты корреляции $\rho(x, y)$ вычислялись по формулам:

$$\text{СКО} = \sqrt{D_x - \tau^2/6};$$

$$\rho(x, y) = \frac{\text{cov}(x, y)}{\sqrt{D_x D_y}}, \qquad (*)$$

где $D_x = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - x_m)^2$ — дисперсия последовательности интервалов x_i, i = 1, 2, ..., N, N — число циклов работы сердца на обрабатываемом фрагменте записи электрокардиосигнала; $\tau^2/6$ — суммарная дисперсия ошибки дискретизации на двух концах интервала [5]: τ — период дискретизации (при частоте дискретизации ЧД = = 128 Гц $\tau^2/6 \cong 10{,}2$ мс^2, при ЧД = 250 Гц $\tau^2/6 \cong$ $\cong 2{,}7$ мс^2); $\text{cov}(x, y) = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - x_m)(y_i - y_m)$ — ковариация последовательностей x_i и y_i; здесь x_i, y_i — i-е значения длительности интервалов RR, RT и т. д., мс; x_m, y_m — средние значения длительности, мс.

Анализ одноминутных записей нормальных ЭКГ Физиобанка осуществлялся с помощью интерактивного алгоритма, включающего распознавание QRS-комплексов методом, предложенным в работе [5], определение координат вершин P- и T-волн в заданной окрестности вершины R QRS-комплекса и статистический анализ массивов RR-, TP-, RT- и PR-интервалов. Для обработки выбирались фрагменты записей, на которых выделение вершин QRS-, P- и T-волн осуществлялось без ошибок. Настройка программы по уровню QRS и параметрам зоны поиска P- и T-волн проводилась вручную отдельно для каждой записи. Для обработки часовых фрагментов и суточных записей использовалась программа, составленная по приведенному ниже алгоритму.

Признаками нормальной кардиограммы является наличие P- и T-волн с интервалами $P_{end}Q$ и QT_{end}, лежащими в пределах 120–200 и 350–420 мс соответственно [6]. В настоящей работе принято, что P (T)-волна обнаружена, если разность между потенциалом в момент P_{max} (T_{max})

и средним значением потенциала в окрестностях этого момента ±[60 мс, 100 мс] (±[150 мс, 400 мс]) превышает разность между потенциалом в момент P_{max} (T_{max}) и средним значением потенциала в ±[0, 60 мс] (±[0, 150 мс])-окрестности этого момента не менее чем в 1,5 раза.

Результаты анализа записей Физиобанка

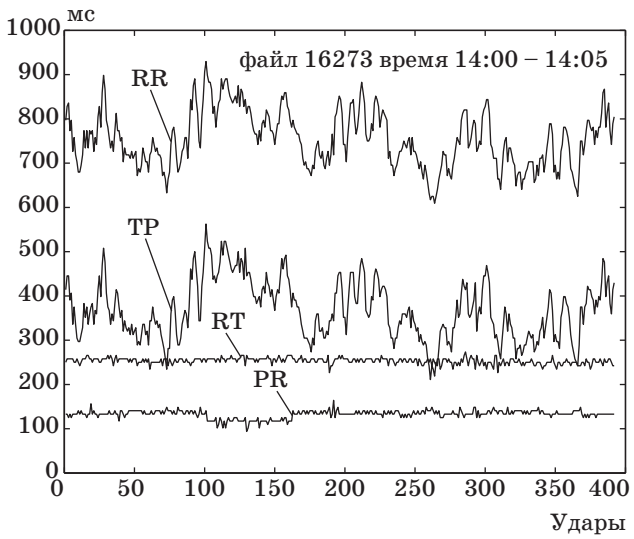
Для анализа статистических и динамических свойств TP-, PR- и RT-интервалов на HF- и LF-частотах обрабатывались одноминутные фрагменты записей в различное время суток. Обследуемые были ранжированы по частоте сердечных сокращений (ЧСС), и в табл. 1–3 приведены результаты расчетов для среднего и двух крайних обследуемых.

■ ***Таблица 1***

База данных, имя файла и частота опроса датчиков	ЧСС	Коэффициент корреляции		
		RR-TP	RR-RT	RR-PR
FD, F1y04, 250 Гц	47	1,0	0,15	0,13
MB NSRD, 16273, 128 Гц	65	0,99	0,13	0,11
FD, F2y05, 250 Гц	80	0,98	0,25	0,08
Разброс значений по выборке	47 ÷ 80	0,96 ÷ ÷ 1,0	−0,11 ÷ ÷ 0,53	−0,5 ÷ ÷ 0,4
Средние значения и СКО по выборке	–	0,98 ± ± 0,02	0,12 ± ± 0,18	−0,11 ± ± 0,23

В нижних графах таблиц приведены пределы изменения, средние значения и СКО рассчитанных по формулам (*) параметров по выборке 50 записей. На рис. 1 представлены ритмограмма и интервалограммы одной из записей. Видно, что:

1) кривая изменения **TP-интервала практически** повторяет RR-ритмограмму, тогда как PR- и RT-интервалы изменяются мало и связь этих изменений с изменениями RR не заметна;



■ ***Рис. 1.*** *RR-ритмограмма и TP-, PR- и RT-интервалограммы* (*запись 16273 банка MB NSRD с 14:00 по 14:05*)

■ ***Таблица 2***

База данных и имя файла		ЧСС	Средние значения и СКО, мс				RT/RR	PR/RR
			RR	TP	RT	PR		
FD, F1y04		47	1271 ± 123	816 ± 123	310 ± 4,8	145 ± 2,4	0,24 ± 0,02	0,11 ± 0,01
MB NSRD, 16273		65	928 ± 67	490 ± 66	286 ± 6,0	137 ± 5,4	0,31 ± 0,03	0,15 ± 0,02
FD, F2y05		80	750 ± 23	342 ± 21,6	248 ± 2,7	161 ± 3,4	0,33 ± 0,03	0,21 ± 0,02
Пределы средних значений по выборке		47 ÷ 80	750 ÷ 1271	309 ÷ 816	233 ÷ 310	122 ÷ 203	0,30 ± 0,03	0,16 ± 0,025
Пределы СКО	σ_{min}	–	22,1	21,9	2,3	2,2	0,02	0,01
	σ_{max}	–	123,4	122,6	14,6	10,9	0,04	0,03

■ ***Таблица 3***

База данных и имя файла	ЧСС	Коэффициент корреляции			СКО, мс		
		dRR-dTP	dRR-dRT	dRR-dPR	dRR	dRT	dPR
FD, F1y04	47	1,0	0,17	0,2	± 165	± 5,4	± 6,2
MB NSRD, 16273	65	1,0	−0,02	0,18	± 160	± 5,7	± 6,4
FD, F2y05	80	0,91	0,18	−0,16	± 18,1	± 4,3	± 5,2
Пределы изменения коэффициента корреляции и СКО	–	0,7 ÷ 1,0	−0,23 ÷ 0,29	−0,25 ÷ 0,23	8,9 ÷ 165,0	3,0 ÷ 27,0	2,4 ÷ 21,8
Средние значения и СКО коэффициента корреляции по выборке	–	0,89 ± 0,10	0,036 ± 0,111	−0,024 ± 0,017			

2) изменение знака отклонения RR от среднего значения происходит на LF-частоте, и, следовательно, коэффициенты корреляции, рассчитанные по формуле (*), характеризуют взаимосвязи интервалов на частоте LF.

В табл. 1 и 2 приведены 3 примера расчетов коэффициентов корреляции, средних значений, СКО и отношений RT/RR и PR/RR для записей, имеющих минимальное, среднее и максимальное значение ЧСС по обрабатываемой выборке. Такие расчеты были проведены для каждой записи выборки, содержащей более 50 записей банков FD и MB NSRD PhysioBank.

Согласно результатам расчетов, приведенным в табл. 1:

1) коэффициенты корреляции между RR и TP близки к единице, откуда следует, что вариабельность сердечного ритма в диапазоне LF-частот осуществляется в основном за счет изменения длительности сердечной диастолы (TP-интервала);

2) коэффициенты корреляции между RR и PR, а также между RR и RT близки к нулю, и, следовательно, последовательности PR- и RT-интервалов практически не содержат LF-частот, связанных с сердечным ритмом. Многие исследователи отмечают наличие в спектральных плотностях RT-интервалограмм пиков на LF-частоте, мощность которых соответствует колебаниям со среднеквадратическими значениями, приблизительно равными 0,6 мс ([4] и ссылки в ней). Однако столь малые колебания можно обнаружить только в записях ЭКГ с ЧД, превышающей 600 Гц (напомним, что СКО дискретизации при ЧД = 128 Гц равна $\tau/\sqrt{6} \cong 3{,}2$ мс и при ЧД = 250 Гц – $\cong 1{,}6$ мс). Авторы [4] обрабатывали электрокардиосигналы, записанные с ЧД = 1000 Гц.

По результатам расчетов, приведенным в табл. 2, могут быть сделаны следующие выводы:

1) средние по выборке обследуемых СКО RT- и PR-интервалов приблизительно на порядок меньше СКО RR- и TP-интервалов;

2) максимальные по выборке записей СКО RT-интервала составляют 14,6 мс, PR-интервала — 10,9 мс.

Хорошо известна формула Базетта, связывающая средние значения RR- и QT_{end}-интервалов: $QT_{end} \cong 0{,}4 \cdot RR_{ср}$. Так как в алгоритме обнаружения P- и T-волн предполагается находить вершины этих волн, были рассчитаны и сведены в таблицу значения отношений RT_{max}/RR и $P_{max}R/RR$ (обозначенные в таблице как RT/RR и PR/RR соответственно) и оценены средние значения этих отношений. Полученные значения: RT/RR = $= 0{,}30 \pm 0{,}03$, PR/RR = $0{,}16 \pm 0{,}025$.

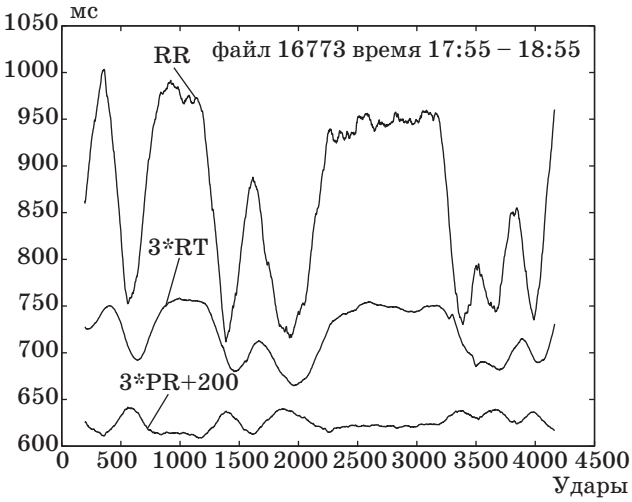
Вычислить коэффициенты корреляции на частоте HF можно различными способами, например, выбирая небольшие участки записи в окрестностях экстремумов LF-колебаний, содержащие достаточное количество HF-колебаний, или отфильтровав LF-колебания низкочастотным фильтром. В настоящей работе для анализа связей между интервалами RR, TP, RT и PR на HF-частотах использовались разности между значениями этих интервалов в очередном i-м сердечном цикле и их значениями в предыдущем $(i-1)$-м цикле, обозначенные ниже как dRR, dRT и dPR. Исследование таких разностей рекомендовано в работе [7]. Результаты расчетов коэффициентов корреляций и СКО последовательностей dRR, dRT и dPR интервалов сведены в табл. 3. Расчеты показывают:

1) коэффициенты корреляции между RR и TP на HF-частотах так же, как на LF-частотах, близки к единице, откуда следует, что вариабельность сердечного ритма в диапазонах высоких (HF) частот осуществляется в основном за счет изменения длительности сердечной диастолы (TP-интервала);

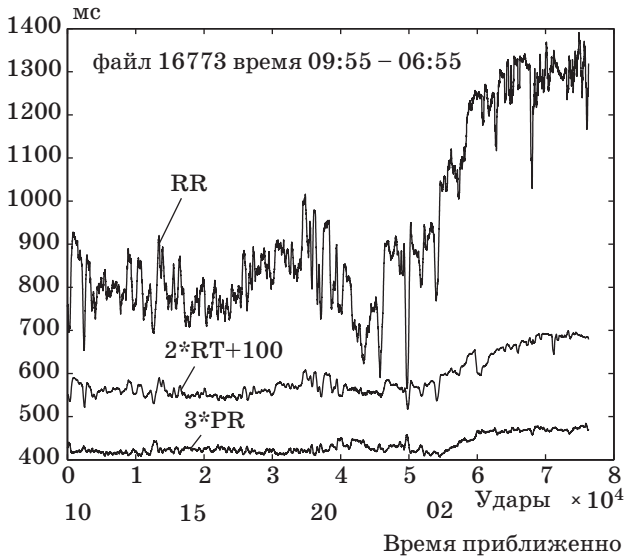
2) коэффициенты корреляции между RR и PR, а также между RR и RT близки к нулю. Согласно многим исследованиям спектральные плотности последовательностей PR- и RT-интервалов содержат пики в области HF-частот, превышающие по амплитуде пики на LF-частотах. Однако существует мнение [4], что колебания PR- и RT-интервалов на частоте дыхания являются артефактами, вызванными изменениями формы грудной клетки и входного сопротивления измерительной аппаратуры при дыхании обследуемого. Наши результаты подтверждают такое предположение;

3) СКО первых разностей RT- и PR-интервалов в 6–7 раз меньше СКО первых разностей RR-интервалов, что говорит о незначительном присутствии колебаний HF-диапазона в последовательностях RT- и PR-интервалов.

Ритмо- и интервалограммы на VLF- и ULF-частотах исследовались с помощью программы, реализующей приведенный ниже алгоритм. Для анализа связей ритмо- и интервалограмм на VLF-частотах использовались одночасовые фрагменты записей, не содержащие заметных суточных дрейфов. Из последовательностей RR-, RT- и PR-интервалов удалялись HF- и LF-частоты путем осреднения на текущем окне длиной 25 мс (один период граничной гармоники между VLF- и LF-частотами 0,04 Гц [7]). Рассчитанные осредненные по множеству записей коэффициенты корреляции составляют $\rho(RR, RT) = 0{,}78 \pm 0{,}12$ и $\rho(RR, PR) = -0{,}75 \pm 0{,}07$. На рис. 2 представлены графики зависимостей средних значений интервалов от времени для записи банка MB NSRD. Для лучшего восприятия связей между кривыми интервалограмма RT изображена в масштабе 3:1, интер-



■ ***Рис. 2.*** *Графики зависимостей усредненных в текущем окне длительностью 25 с значений интервалов от времени для записи 16773 MB NSRD с 17:55 по 18:55*



■ ***Рис. 3.*** *Графики зависимостей усредненных в текущем окне длительностью 333 с значений интервалов от времени для записи 16773 банка MB NSRD с 09:55 по 06:55*

валограмма PR — в масштабе 3:1 и смещена вверх на 200 мс.

Высокие значения коэффициентов корреляции ρ(RR, RT) и ρ(RR, PR) показывают, что одновременные изменения RR-, RT- и PR-интервалов вызваны одной причиной, связанной с работой сердца. Следует отметить различные знаки корреляций ρ(RR, RT) и ρ(RR, PR): при возрастании ЧСС (уменьшении **RR-интервала) время поляризации** и реполяризации желудочков сокращается, тогда как скорость проведения возбуждения через AV-узел увеличивается.

Исследования характера ритмо- и интервалограмм на ULF-частотах проводились с использованием только банка данных **MB NSRD. Значения RR-, PR- и RT-интервалов усреднялись текущим** окном длительностью 333 с, равной периоду граничной частоты между VLF- и ULF-диапазонами. На рис. 3 изображены ритмо- и интервалограммы для записи 16773 указанного банка (интервалограмма RT изображена в масштабе 2:1 и смещена вверх на 100 мс, интервалограмма PR — в масштабе 3:1). Рассчитаны коэффициенты корреляции: ρ(RR, RT) = 0,96, ρ(RR, PR) = 0,79. На рисунке видно, что все три интервала (RR, RT и PR) одновременно возрастают в ночное время.

Проведенный анализ приводит к следующим утверждениям, принятым за основу при построении алгоритма выделения вершин P- и T-волн — наиболее сложной части общего алгоритма выделения участков с нормальным синусовым ритмом.

1. При построении алгоритмов автоматического поиска вершин **P- и T-волн последовательности измерений PR- и RT-интервалов могут рассматриваться** как случайные временные ряды, содержащие:

а) высокочастотные составляющие (HF- и LF-частоты), которые могут быть отнесены к внешним помехам (без существенного ущерба истине);

б) низкочастотные составляющие (VLF- и ULF-частоты), коррелированные с медленными изменениями RR-интервала и отражающие работу сердца.

2. Граничным значением между высокочастотными и низкочастотными составляющими колебаний PR- и RT-интервалов можно считать граничную частоту между LF- и VLF-диапазонами, равную 0,04 Гц [7].

3. Среднеквадратические отклонения RT- и PR-интервалов на одноминутных записях приблизительно на порядок меньше СКО RR-интервала, а средние (за 1 мин) значения RT- и PR-интервалов изменяются в 6–8 раз медленнее соответствующих средних значений RR-интервала.

4. Максимальные значения СКО по множеству одноминутных записей составляют: $\sigma_{\max RR} \cong$ $\cong 125$ мс, $\sigma_{\max PR} \cong 11$ мс, $\sigma_{\max RT} \cong 15$ мс.

5. Средние по выборке записей отношения длин интервалов RT и PR к среднему за 25 с значению длины интервала RR составляют: $RT/RR_{mid} = 0{,}30 \pm 0{,}03$, $PR/RR_{mid} = 0{,}16 \pm 0{,}025$.

Самонастраивающийся алгоритм выделения фрагментов с нормальным синусовым ритмом

Такой алгоритм включает алгоритм распознавания **QRS-комплексов и алгоритм идентификации** координат вершин P- и T-волн. Оба алгоритма

работают в едином цикле по времени, и временная координата вершины R каждого QRS-комплекса является опорной точкой обнаружения P- и T-волн.

Самонастройка алгоритма распознавания QRS-комплексов осуществляется на первых 25 с записи, на которых определяется максимальное по абсолютной величине значение первой разности сигнала — dR_{max} — и приближенно оценивается среднее значение RR-интервала — RR_{mid25} (мс). Подавляющее большинство QRS-комплексов обнаруживается простейшим методом — по превышению абсолютного значения первой разности сигнала величины $0{,}6 \cdot dR_{max}$. Если QRS-комплекс не обнаружен на временном интервале, превышающем RR_{mid25} (мс), производится углубленный поиск низкоамплитудного QRS-комплекса путем снижения порога по первой разности в 1,6 раза и введения дополнительных ограничений на ширину комплекса и временные интервалы между соседними комплексами. Оценка эффективности алгоритма проведена на произвольно выбранных фрагментах записей базы данных MB NSRD PhysioBank, содержащих в общей сложности более $2 \cdot 10^5$ QRS-комплексов. Чувствительность алгоритма составляет 99,8 %, избирательность — 99,7 %. Такая высокая эффективность алгоритма может быть объяснена, на наш взгляд, малой вероятностью появления низкоамплитудных QRS-комплексов в условиях повышенной мышечной активности обследуемого.

Представление последовательности PR- и RT-интервалов в виде низкочастотных временных рядов, зашумленных высокочастотными помехами (см. выше п. 1 результатов анализа записей PhysioBank), делает целесообразным построение алгоритма поиска вершин P (T)-волн, основанного на прогнозе центра зоны поиска очередной вершины по низкочастотной составляющей последовательности PR (RT)-интервалов и вычислении ширины зоны поиска по оценкам СКО этих интервалов. Итеративное вычисление низкочастотной составляющей PR (RT)-последовательности может быть осуществлено с помощью какого-либо дискретного низкочастотного фильтра (например, фильтра Баттерворта) или осреднением в текущем окне. Согласно п. 2 представленного выше анализа, граничной частотой между высокочастотными и низкочастотными составляющими колебаний PR- и RT-интервалов можно считать граничную частоту между LF- и VLF-диапазонами, равную, согласно [7], 0,04 Гц. Поэтому при расчете низкочастотного фильтра частота среза f_c выбрана равной 0,04 Гц. Частота дискретизации f_s PR- и RT-последовательностей приближенно равна средней ЧСС здорового человека, равной 68 уд/мин ≅ 1,14 Гц [6]. При фильтрации PR- и RT-последовательностей методом текущего окна длина окна L_c выбирается равной периоду частоты среза: $L_c = 1/f_c = 25$ с, что приближенно соответствует длительности 28 QRS-комплексов здорового человека.

Самонастраивающийся алгоритм выделения фрагментов с нормальным синусовым ритмом включает следующие шаги.

1. Предварительная грубая обработка первых 25 с файла данных. Работает только алгоритм поиска QRS-комплексов, вычисляются dR_{max} и RR_{mid25}.

2. Определение начальных значений:

— порога поиска QRS-комплексов по первой разности — $0{,}6 \cdot dR_{max}$;

— центров зон поиска вершин P- и T- волн в отсчете от опорных точек R: $PR_c = 0{,}16 \cdot RR_{mid25}$ (мс) и $RT_c = 0{,}3 \cdot RR_{mid25}$ (мс) согласно п. 5 результатов анализа записей PhysioBank;

— ширины зоны поиска вершин P- и T-волн: $\pm 3\sigma_{maxPR}$ для P- и $\pm 3\sigma_{maxRT}$ для T-волны, где $\sigma_{maxPR} = 11$ мс и $\sigma_{maxRT} = 15$ мс согласно п. 4 результатов анализа записей PhysioBank.

3. Далее по шагам $i = 1, 2, \ldots$:

— идентификация временной координаты вершины R i-го QRS-комплекса;

— идентификация временных координат вершин i-х P- и T-волн;

— обновление средних значений и СКО RR-, PR- и RT-интервалов с учетом найденных i-х значений этих интервалов;

— прогноз $(i + 1)$-х центров зон поиска P- и T-волн;

— уточнение ширины $(i + 1)$-х зон поиска P- и T-волн по обновленным СКО PR- и RT-интервалов;

— контроль длины RR-интервала и наличия P(i)- и T(i)-волн по признаку выпуклости окрестностей найденных вершин P_{max} и T_{max};

— переход к следующему шагу п. 3: $i = i + 1$.

Признаком конца фрагмента с нормальным синусовым ритмом и условием обнаружения аномального удара являются: 1) отклонение длительности RR-интервала от текущего среднего $\sigma_{RR}(i)$ более чем на три σ_{maxRR}; 2) отсутствие локального максимума на интервалах поиска вершин P- или T-волн или 3) отсутствие выпуклой волны в окрестности P_{max} или T_{max}.

Заключение

В работе предложен самонастраивающийся алгоритм выделения фрагментов с нормальным синусовым ритмом в длинных записях ЭКГ. Алгоритм включает поиск опорных точек — вершин R QRS-комплексов и последующий поиск вершин P- и T-волн, основанный на обновлении зон поиска на каждом шаге.

Для автоматического обнаружения вершин QRS-комплексов предложен эвристический алгоритм, показавший при испытаниях на базе данных MB NSRD PhysioBank высокую эффективность: чувствительность алгоритма превышает 99,8 %, избирательность – 99,7 %.

Для проектирования самонастраивающегося алгоритма поиска вершин P- и T-волн проведен анализ более 50 записей Физиобанка с нормальным синусовым ритмом. Проведенный анализ показал, что последовательности измерений PR- и RT-интервалов могут рассматриваться как случайные временные ряды, содержащие: а) низкочастотные составляющие (< 0,04 Гц), коррелированные с медленными изменениями RR-интервала, и б) высокочастотные помехи с частотой, превышающей 0,04 Гц. Такое рассмотрение стало основой построения самонастраивающегося алгоритма, а численные результаты анализа позволили рассчитать параметры самонастройки алгоритма. Программа, реализующая предложенный алгоритм, надежно обрабатывает 24-часовые записи ЭКГ.

В заключение отметим также интересный результат, полученный в процессе анализа записей Физиобанка и не известный автору из литературных источников: значительные по абсолютной величине коэффициенты корреляции между RR- и PR-интервалами (~0,8) имеют различные знаки на VLF- и ULF-частотах. Этот результат может говорить о различных физиологических механизмах регулирования проводимости AV-узла при суточных колебаниях и колебаниях на VLF-частоте.

Литература

1. **Ракчеева Т. А.** Образный анализ ЭКГ // Медицинская техника. 1995. № 2. С. 9–16.
2. **PhysioBank**. http://physionet.org/physiobank (дата обращения: 01.11.2009).
3. **Köhler B. U., Henning C., Orglmeister R.** The Principles of software QRS detection, Reviewing and comparing algorithms for detecting this important ECG waveform // IEEE Eng. Med. Biol. 2002. Jan. — Feb. P. 42–57.
4. **Porta A. et al.** Performance assessment of standard algorithms for dynamic R-T interval measurement: comparison between R-Tapex and R-Tend approach // Med. Biol. Eng. Comput. 1998. N 36. P. 35–42.
5. **Бахилин В. М.** Помехоустойчивые алгоритмы обнаружения характерных точек электрокардиосигналов // Изв. СПбГЭТУ «ЛЭТИ». 2008. № 5. С. 56–60.
6. **Норма** в медицинской практике: справ. пособие / Редактор-составитель А. В. Литвинов. — М.: МЕДпресс, 2000. — 144 с.
7. **Heart** rate variability: Standards of measurement, physiological interpretation, and clinical use // European Heart Journal / Task Force of The European Society of Cardiology and The North American Society of Pacing and Electrophysiology. 1996. N 17. P. 354–381.

УДК 334.021.1

НЕКОТОРЫЕ ПРОБЛЕМЫ ИНСТИТУАЛИЗАЦИИ ГОСУДАРСТВЕННО-ЧАСТНОГО ПАРТНЕРСТВА

М. Р. Орлов,
генеральный директор
ООО «Стилэкс»

Рассмотрены проблемы институализации проектов государственно-частного партнерства в части формализации процессов принятия решения и управления такими проектами. Проанализировано состояние отечественной законодательной базы, связанной с реализацией государственно-частного партнерства. Показано, что наиболее эффективным инструментом согласования интересов и управления проектами государственно-частного партнерства служит система сбалансированных показателей.

Ключевые слова *— проектное управление, государственно-частное партнерство, система сбалансированных показателей.*

В последнее время во всем мире происходят важные институциональные изменения в отраслях, которые раньше всегда находились в государственной собственности и управлении: энергетике, коммунальном хозяйстве, на транспорте, в портах и т. п. Правительства передают во временное пользование бизнесу объекты этих отраслей, оставляя за собой право регулирования и контроля за их деятельностью. Это связано с тем, что, с одной стороны, предприятия инфраструктурных отраслей (главным образом их сетевые, монопольные объекты) не могут быть приватизированы ввиду своей стратегической, экономической и социально-политической значимости, а, с другой стороны, бюджет не имеет достаточного количества средств для их воспроизводства. За рубежом это противоречие нашло свое разрешение путем использования концепции государственно-частного партнерства (ГЧП, Public-Private Partnership — PPP), представляющего собой альтернативу приватизации важных объектов государственной собственности. Такое партнерство имеет вид организационного альянса между государством и бизнесом для реализации общественно значимых проектов разного масштаба в широком спектре сфер деятельности. Как любой проект, каждый такой альянс является временным, поскольку создается на определенный срок в целях осуществления конкретного проекта и прекращает свое существование после его реализации.

Ограниченные ресурсы отечественного бюджета заставляют обратить пристальное внимание на эту форму привлечения средств бизнеса для решения задач, связанных с совершенствованием инфраструктуры, а также других актуальных социальных задач. Однако на этом пути общество и государственные институты столкнулись с существенными трудностями. Для того чтобы понять их суть и предложить пути преодоления возникающих проблем, рассмотрим цели и задачи, решаемые участниками ГЧП, сложившуюся на сегодняшний день в РФ законодательную базу, а также возможные риски сторон при реализации проектов ГЧП.

Очевидно, что в отношении ГЧП государственный сектор преследует как внутренние, так и внешние цели. Внутренние цели администраций всех уровней состоят в извлечении выгоды из знания рынка и использовании деловых компетенций частных партнеров, что позволяет, выполняя задачи в форме ГЧП, сокращать численность аппарата и одновременно повышать качество его работы. Внешний результат ГЧП, ради которого, по сути, и реализуются соответствующие проекты, направлен на повышение экономического потенциала региона, создание рабочих мест, решение социальных и иных задач, определенных на политическом и административном уровнях.

Целью частных партнеров в рамках структуры ГЧП служит увеличение или поддержка (по крайней мере, в долгосрочной перспективе) уровня прибыли. Помимо этого у них появляется возможность, опираясь на государственную под-

держку, снизить уровень своих рисков и увеличить долю рынка, занимаемую бизнесом. Наконец, связи с государственной администрацией и участие в проектах местного развития существенно способствуют улучшению их имиджа [1].

Юридическая (договорно-правовая) институализация взаимодействия государства и бизнеса в рамках ГЧП на основе консолидации отечественных и зарубежных моделей такого партнерства принципиально может принять одну из следующих форм.

1. Сервисные контракты:

— контракты на выполнение работ и оказание общественных услуг; по поставке продукции для государственных нужд в виде прямой покупки государством конкретной продукции, услуг или имущества у частных производителей;

— контракты технической помощи в виде использования частных компаний для решения конкретных задач, в которых у государства возникают проблемы, например из-за недостатка конкретного ресурса (ликвидация последствий стихийных бедствий).

2. Управляющие контракты:

— контракты на обслуживание, когда администрация заключает контракт с частным сектором на обслуживание того или иного объекта;

— контракты на управление объектом, связанным с заключением контракта между администрацией и частным сектором на управление объектом;

— контракты «под ключ», в соответствии с которыми администрация финансирует, а частный сектор проектирует, строит и управляет объектом.

3. Договоры аренды и временной передачи прав:

— лизинговые контракты, близкие по содержанию к арендным, при которых арендатор, не участвуя в строительстве, получает объект от государства с налагаемыми обязанностями: обслуживание, взимание платы, платежи государству за пользование;

— соглашения о разделе продукции (СРП) на условиях передачи частным производителем части выпускаемой продукции государству в качестве оплаты права реализации конкретного проекта;

— инвестиционные контракты, когда государство отказывается на указанный в контракте срок от получения фискальных доходов, позволяя частному предприятию завершить инвестиционный проект, реализация которого имеет большой социально-экономический эффект, с использованием этих средств.

4. Концессионные соглашения разных типов:

— BOOT (Build — Own — Operate — Transfer) — «Строительство — Владение — Эксплуатация/управление — Передача», тип классической концессии, в соответствии с которой частный партнер получает правомочие не только на строительство и пользование, но и владение объектом в течение срока соглашения, по истечении которого он передается исполнительной власти;

— DBOOT (Design — Build — Own — Operate — Transfer) — «Проектирование — Строительство — Владение — Эксплуатация/управление — Передача», отличающийся от первого тем, что на частного партнера возлагается ответственность не только за строительство инфраструктурного объекта, но и за его проектирование;

— ROOT (Reconstruct — Own — Operate — Transfer) — «Реконструкция — Владение — Эксплуатация/управление — Передача», связанный с тем, что частный партнер вместо строительства получает инфраструктурный объект на реконструкцию с последующим владением, пользованием и передачей исполнительной власти;

— ROO (Reconstruct — Own — Operate) — «Реконструкция — Владение — Эксплуатация/управление», не предполагающий возврат объекта после восстановления и эксплуатации в собственность государства;

— MFO (Maintain — Finance — Operate) — «Обслуживание — Финансирование — Эксплуатация/управление», направленный на предоставление частным партнером товаров, работ или услуг потребителям с использованием предоставляемого для этой цели объекта соглашения.

5. Акционирование, долевое участие частного капитала в государственных предприятиях (совместные предприятия).

В этом случае участие государства распространяется до степени получения блокирующего меньшинства. Тем самым государство приобретает достаточное влияние в компании. В случае многочисленных смешанных компаний SEM во Франции, например, закон предписывает, что государственный сектор должен обладать большей частью активов [2].

Основные классификационные признаки перечисленных выше форм ГЧП приведены в табл. 1.

В силу своей комплексности общие вопросы ГЧП могут регулироваться гражданским законодательством и, кроме того, нормами бюджетного, налогового и иных отраслей права, Федерального закона от 21.07.2005 г. № 94-ФЗ «О размещении заказов на поставки товаров, выполнение работ, оказание услуг для государственных и муниципальных нужд», Федерального закона от 21.12.2001 г. № 178-ФЗ «О приватизации государственного и муниципального имущества», Федерального закона от 26.07.2006 г. № 135-ФЗ «О защите конкуренции» и т. д. Поэтому активно дискутируемая в настоящее время необходимость

■ *Таблица 1*

Виды ГЧП	Продолжительность	Компенсация для исполнителя	Функции исполнителя
Сервисные контракты	Короткий срок (1–3 года)	Вознаграждение от заказчика за выполнение услуг	Определенный, всегда связанный с техникой вид услуг
Управляющие контракты	Средний срок (3–8 лет)	Вознаграждение от правительства за оказание услуг	Управление деятельностью, переданной государством
Аренда и временная передача прав	Длительный срок (8–15 лет)	Все доходы, вознаграждения и сборы от потребителя за оказанные услуги; поставщик услуг выплачивает государству ренту за объект	Управление, ремонт и техническое обслуживание (возможно, инвестирование) муниципальным имуществом, которое оказывает услуги по специфическим стандартам
Концессионное соглашение	Длительный срок (15–30 лет)	Все доходы от потребителя за оказанные услуги; поставщик услуг выплачивает государству установленные договором выплаты и может брать на себя выплату существующей задолженности	Управление, ремонт, техническое обслуживание и инвестирование в государственную/муниципальную инфраструктуру по заданным параметрам
Участие в капитале	Неограниченный срок	Доходы компании с участием государственного и частного капитала распределяются пропорционально участию	В соответствии с ГК РФ

принятия федерального закона о ГЧП не является абсолютно очевидной, а предполагаемое его содержание вызывает множество споров [3]. Следует ожидать, что в основу такого закона будет положено легальное определение самого ГЧП, общие принципы его реализации, а также модернизация существующей (формировавшейся с начала 90-х гг.) нормативной базы в сфере инвестиционной деятельности.

На региональном уровне в настоящее время законы о ГЧП приняты лишь в нескольких субъектах РФ (Республиках Алтай, Дагестан, Калмыкия, Томской области, г. Санкт-Петербурге), при этом большая их часть имеет ряд существенных недостатков [4]. Эти законы:

— не учитывают опыта уже предпринятых попыток реализации федеральных проектов ГЧП и сложностей, возникших в ходе применения существующей нормативной базы;

— не охватывают значительного количества инструментов, которые действительно интересны региональным и национальным инвесторам;

— имеют невысокий уровень юридической техники, что приводит к коллизиям с нормами федерального законодательства;

— основаны на неверном представлении о реальном содержании различных инструментов ГЧП и носят поэтому откровенно декларативный характер.

Наглядное представление о законодательном закреплении различных видов хозяйственного взаимодействия в региональных законах о ГЧП можно получить из табл. 2.

■ *Таблица 2*

Субъект Федерации	Виды хозяйственного взаимодействия				
	BOOT	DBOOT	ROOT	ROO	MFO
Республика Дагестан	+	+	+	+	+
Республика Алтай	Не определены				
Республика Калмыкия	Не определены				
Город Санкт-Петербург	+	+	+	+	+
Томская область	Не определены				

В целом следует отметить, что в настоящее время указанная выше законодательная база с привлечением таких нормативных документов, как Федеральный закон от 21.07.2005 г. № 115-ФЗ «О концессионных соглашениях», Федеральный закон от 21.07.2005 г. № 116-ФЗ «Об особых экономических зонах в Российской Федерации», Постановление Правительства Российской Федерации от 01.03.2008 г. № 134 «Об утверждении правил формирования и использования бюджетных ассигнований инвестиционного фонда Российской Федерации» и др. составляет достаточную основу для реализации всего спектра форм ГЧП.

Несравненно более важным обстоятельством, стоящим на пути реализации проектов ГЧП, является анализ и распределение рисков в процессе формирования и осуществления проектов, а также тяжелый груз взаимного недоверия властных

структур и бизнеса, накопившийся за период перехода к рыночной экономике. Эти обстоятельства, помноженные на отсутствие коммуникативных навыков между представителями различных общественных институтов и дополненные недостаточно проработанным механизмом организации экономического взаимодействия сторон в рамках ГЧП [5], привели к торможению использования этой эффективной формы взаимодействия общества и бизнеса. Выход из положения может быть найден на пути преодоления взаимного недоверия и формирования таких процедур управления проектами сотрудничества, которые обеспечат максимальную их прозрачность.

Рассмотрим сначала потенциальные риски сторон при осуществлении ГЧП. К ним относятся:

для государственного сектора:

— *асимметричные информационные потоки*, связанные с тем, что частные партнеры обладают лучшим знанием рынка и выгодных инвестиционных возможностей, нежели государственный сектор, не говоря уже о более высокой компетенции частных партнеров в части выполнения проектов — дефицит всего этого составляет большое неудобство для государственного сектора;

— *возможные нечестные намерения частного партнера*, поскольку нельзя гарантировать, что частные партнеры будут действовать строго в соответствии с условиями договора, особенно когда государственный сектор отдает существенную часть управления партнерством в руки частного партнера; поэтому если государственный сектор уже связан схемой сотрудничества, он уязвим для любых возможных нечестных намерений со стороны частного партнера, который захочет воспользоваться возможностями государственного сектора;

— *опасность передачи рисков государственному сектору*, обусловленная тем, что ГЧП может привести к краткосрочному сокращению финансового давления на государственные бюджеты, однако опасность, что проектный риск будет передан обратно, остается высокой, причем это довольно большой риск, поскольку государственный сектор несет определенную ответственность за проект. При неудаче проекта государственный сектор не сможет легко оставить финансовые последствия частным партнерам, если это приведет к провалу проекта со всеми отрицательными политическими последствиями. В этом случае государственный сектор практически вынужден действовать, т. е. переносить часть бремени финансирования проекта с пользователей инфраструктуры или услуг на налогоплательщиков;

для коммерческих организаций:

— *политический и юридический риски*, связанные с возможными изменениями в правительственной политике, неблагоприятными или непостоянными рыночными условиями, неспособностью или отказом государственного сектора выполнить условия договора, а также с изменениями в налоговом законодательстве или принятием на государственном уровне мер, имеющих неблагоприятные последствия для частных партнеров;

— *технические риски*, касающиеся выполнения строительства и использования инфраструктуры и охватывающие различные компоненты: неожиданные технические проблемы при реализации проектов, неопределенно сформулированные экологические требования, отклонения от планового графика выполнения проекта, превышение бюджета затрат, прерывания сервисного обслуживания и т. д.;

— *экономический и финансовый риски*, обусловленные, прежде всего, степенью инфляции и изменением курсов обмена валют (валютные займы и валюта, связанная с доходами пользователей); в эту категорию включают также факторы, отражающие состояние и прогноз развития экономики;

— *коммерческие риски*, учитывающие факторы как взаимоотношений организации с клиентами, так и ценообразование на товары и услуги, являющиеся предметом ГЧП, поэтому коммерческие риски надо рассматривать в общем контексте экономики и общества, поскольку покупательская способность потенциальных пользователей и привлекательность для них продукции ГЧП имеют определяющее значение в экономической успешности проекта;

— *риски форс-мажора*, которые не требуют дополнительных комментариев и связаны с естественными бедствиями, социальными беспорядками и войнами, к которым в нашем веке добавилась угроза терроризма.

Необходимость уменьшения указанных рисков путем ранней диагностики угроз и их парирования наряду с преодолением взаимного недоверия сторон партнерства друг к другу требует формирования таких процедур управления проектами сотрудничества, которые обеспечат максимальную их прозрачность. К сожалению, сегодня отсутствуют как подзаконные акты, так и рекомендованные методики организации управления проектами ГЧП, поэтому приходится обращаться к разработанным в последнее время инструментам управления проектами, среди которых, в первую очередь, следует упомянуть систему сбалансированных показателей (Balanced Scorecard — BSC), получившую заслуженную популярность благодаря успешному ее применению в различных сферах деятельности [6].

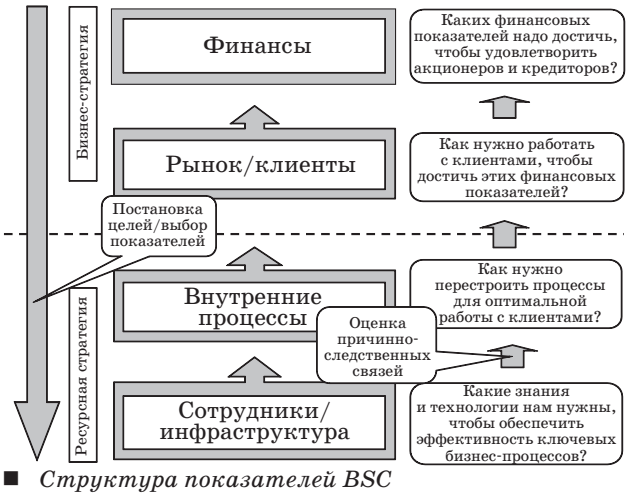
Как известно, система BSC содержит четыре основные перспективы, определяющие содержа-

ние хозяйственной деятельности субъекта и совокупность показателей, отражающих ее эффективность (рисунок). Такая система служит моделью бизнеса и связывает между собой ресурсы, бизнес-процессы, рыночные показатели и финансовые результаты.

Исходя из самой сути ГЧП, направленной на повышение эффективности использования принадлежащих обществу ресурсов, эти ресурсы передаются в пользование коммерческой организации, а получаемая рента распределяется согласно условиям договора. Таким образом, в рассматриваемом случае общая структура BSC остается неизменной с добавлением в ресурсную составляющую инвестиционных вкладов со стороны государства и учета в финансовой составляющей соответствующего оттока ренты. В том случае, когда договором предусматривается использование сложных финансовых схем (например, при организации софинансирования для осуществления программ льготного жилищного строительства), возможно разделение финансовой составляющей с выделением верхнего уровня, отражающего реализацию этих схем.

Помимо наглядного описания функционирования проекта, система **BSC позволяет обеспечить** соответствие и непротиворечивость целей всех стейкхолдеров, управляя противоречиями и конфликтами [7]. Основной их источник — это разное видение способов увеличения успешности бизнеса, разный жизненный опыт, уровень образования, глубина мышления и многое другое, отличающее стейкхолдеров как личностей между собой. В табл. 3 приведены основные типы противоречий, могущих возникнуть между собственниками и менеджментом организации и способы использования BSC для их преодоления.

Используя систему **BSC для выработки консенсуса**, действия по преодолению противоречий можно представить так:



■ *Структура показателей BSC*

■ ***Таблица 3***

Тип противоречия	Способ преодоления противоречий
Собственник и топ-менеджер в одном лице	BSC — система, ориентированная на определение баланса между инвестициями и потреблением
Собственник в поисках топ-менеджера	BSC — инструмент постановки целей наемному руководителю и контроля достижений
Собственник в конфликте с топ-менеджментом	BSC — инструмент, помогающий развернуть реализацию целевых показателей до уровня исполнения и сделать ее «прозрачной» для собственника
Собственники в конфликте друг с другом	BSC — способ устранения противоречий во взглядах на цели и средства их достижения

— в случае совмещения собственника и топ-менеджера в одном лице BSC помогает соблюсти баланс между инвестициями и потреблением, обеспечивая контроль уровня средств, необходимый для успешного функционирования и развития бизнеса;

— когда собственнику требуется сформировать команду управленцев и он находится в поисках топ-менеджера, BSC служит инструментом постановки целей наемному руководству и контроля достижений. В этом случае у топ-менеджмента появляется четкая программа действий и понятные количественные ориентиры;

— в случае конфликта между собственниками и топ-менеджментом разработанная стратегия реализации проекта, описываемая принятой сторонами системой количественных показателей, становится законом для топ-менеджмента, которому собственник передает все полномочия по управлению бизнесом, оставляя за собой текущий контроль показателей для парирования катастрофических ситуаций;

— при конфликте между собственниками основным руководящим документом для топ-менеджеров также служит утвержденная система показателей, чем исключается влияние отдельных мнений на работу организации.

Нетрудно заметить, что перечисленные варианты охватывают все возможные случаи противоречий, которые могут возникнуть в ходе подготовки и реализации проектов ГЧП.

Еще одним преимуществом использования методики **BSC для решения рассматриваемой задачи** служит большое количество программных средств для ее реализации. Общепризнанные мировые стандарты качества в области программного обеспечения для BSC определяет компания Balanced Scorecard Collaborative, Inc., созданная авторами концепции **Balanced Scorecard. Между-**

народным сертификатом BSCol Certified, подтверждающим соответствие продукта методологии Balanced Scorecard, обладают 23 программных решения со всего мира. Россию и страны СНГ представляет российская разработка «Инталев: Навигатор» [8].

Программные продукты для BSC управления можно разделить на несколько категорий. Во-первых, это отдельные программы для разработки стратегических карт BSC (примеры: QPR ScoreCard, Dialog Strategy). Во-вторых, это входящие в комплексные системы для управления предприятием (ERP, BPM и др.) модули, например Geac Performance Management, Business Performance Management, Cognos Metrics Manager, SAP Strategic Enterprise Management. Первые сравнительно просты в настройке и использовании, доступны по цене, но упускают из виду такие важные аспекты, как бюджетирование и бизнес-процессы. Вторые выигрывают за счет комплексного подхода, но обладают высокой стоимостью. Удачным компромиссом в таких случаях может выступить третий вариант — система для проектирования бизнеса, включающая в себя модуль BSC. Такое программное обеспечение обладает широкими возможностями для формирования стратегии, системы бюджетирования, структуры организации и бизнес-процессов (а иногда и других подсистем, например, клиентской подсистемы), но не замахивается на их полную автоматизацию, что существенно снижает стоимость продукта и упрощает его интерфейс.

Литература

1. **Варнавский В. Г.** Партнерство государства и частного сектора: формы, проекты, риски / ИМЭМО РАН. — М., 2005. — 176 с.
2. **Вилисов М.** Государственно-частное партнерство: политико-правовой аспект // Власть. 2006. № 7. С. 28–32.
3. **Глумов Е.** Закон о государственно-частном партнерстве: необходимость принятия и предмет регулирования // Корпоративный юрист. 2009. № 5. С. 35–37.
4. **Орлов М. Р.** Экономический анализ проектов государственно-частного партнерства // Надежность и качество — 2010: Тр. Междунар. симп. Пенза, 2010. С. 305–308.
5. **Кашин С.** Не в дружбу, а в госслужбу // Секрет Фирмы. 2005. № 30 (117). С. 17–19.
6. **Каплан Р. С., Нортон Д. П.** Организация, ориентированная на стратегию. Как в новой бизнес-среде преуспевают организации, применяющие сбалансированную систему показателей: Пер. с англ. — М.: Олимп-Бизнес, 2004. — 346 с.
7. **Орлов Р. А.** Технология создания и управления брендом / ГУАП. — СПб., 2008. — 448 с.
8. Внедрение системы сбалансированных показателей: постановка и автоматизация / КВФ «Инталев». — СПб., 2005. — 262 с.

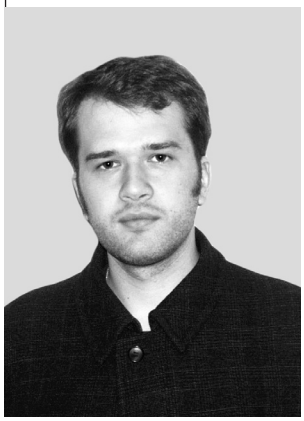
АБРАМОВ Валентин Анатольевич

Преподаватель Центра детско-юношеского технического творчества Кировского района Санкт-Петербурга.
В 2010 году окончил Санкт-Петербургский государственный университет информационных технологий механики и оптики по специальности «Приборостроение».
Область научных интересов — измерительные и диагностические системы.
Эл. адрес: Abrvalnic@mail.ru

БАХИЛИН Виктор Михайлович

Соискатель кафедры биомедицинской электроники и охраны среды Санкт-Петербургского государственного электротехнического университета «ЛЭТИ», научный сотрудник Санкт-Петербургского НИИ уха, горла, носа и речи Федерального агентства по высокотехнологичной медицинской помощи.
В 1985 году окончил Ленинградский институт инженеров железнодорожного транспорта им. В. Н. Образцова по специальности «Строительные и дорожные машины и оборудование».
Область научных интересов — обработка и анализ биомедицинских сигналов.
Эл. адрес:
Victor_b_62@mail.ru

БЕРЕЗКИН Алексей Владимирович



Аспирант кафедры компьютерных систем и программных технологий факультета технической кибернетики Санкт-Петербургского государственного политехнического университета.
В 2009 году окончил Санкт-Петербургский государственный политехнический университет, получив степень магистра техники и технологии.
Является автором пяти научных публикаций.
Область научных интересов — отказоустойчивые системы, встраиваемые системы (в том числе системы тестирования), системы на кристалле, моделирование программного и аппаратного обеспечения.
Эл. адрес:
berezkin@aivt.ftk.spbstu.ru

ГОЛУБКОВ Андрей Сергеевич

Инженер первой категории, младший научный сотрудник научно-технической лаборатории систем технического зрения и экспертных систем Института менеджмента и информационных технологий филиала Санкт-Петербургского государственного политехнического университета в г. Череповце.
В 2006 году окончил Череповецкий государственный университет по специальности «Управление и информатика в технических системах».
Является автором семи научных публикаций.
Область научных интересов — моделирование, теория автоматического управления.
Эл. адрес: golubkov@imit.ru

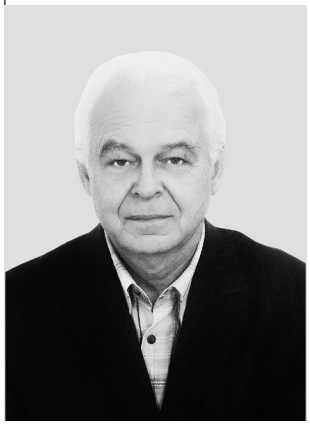
ДМИТРЕВИЧ Геннадиий Данилович

Профессор кафедры систем автоматизированного проектирования Санкт-Петербургского государственного электротехнического университета «ЛЭТИ».
В 1962 году окончил Ленинградский электротехнический институт им. В. И. Ульянова (Ленина) по специальности «Автоматика и телемеханика».
В 1990 году защитил диссертацию на соискание ученой степени доктора технических наук.
Является автором более трехсот научных публикации.
Область научных интересов — САПР в электронике и машиностроении.
Эл. адрес:
GDDmitrivech@eltech.mail.ru

ДРОЗДОВА Лариса Николаевна

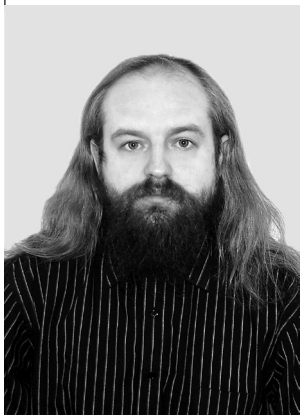
Доцент, заведующая лабораторией нейропсихологии Красноярского государственного университета им. В. П. Астафьева.
В 1974 году окончила Красноярскую медицинскую академию по специальности «Врач-невролог».
В 1986 году защитила диссертацию на соискание ученой степени кандидата медицинских наук.
Является автором более 100 научных публикаций.
Область научных интересов — нейрология, исследование базовых когнитивных функций мозга, системы с биологической обратной связью для коррекции познавательной сферы человека.
Эл. адрес: drozdov@kspu.ru

ДУБАРЕНКО Владимир Васильевич



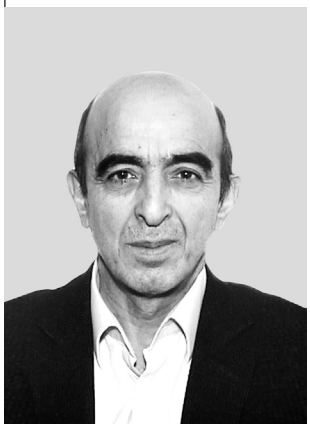
Ученый секретарь Института проблем машиноведения РАН.
В 1963 году окончил Ленинградский военно-механический институт по специальности «Механика», в 1965 году — по специальности «Системы управления».
В 2002 году защитил диссертацию на соискание ученой степени доктора технических наук.
Является автором 70 научных публикаций.
Область научных интересов — интеллектуальные системы и системы управления.
Эл. адрес: dvv@msa.impe.ru

КОЗАЧЕНКО Александр Викторович



Доцент кафедры измерительных технологий и компьютерной томографии Санкт-Петербургского государственного университета информационных технологий, механики и оптики.
В 1999 году окончил Санкт-Петербургский государственный институт точной механики и оптики по специальности «Приборостроение».
В 2007 году защитил диссертацию на соискание ученой степени кандидата технических наук.
Является автором более 20 научных публикаций.
Область научных интересов — томография, неразрушающий контроль, технические измерения.
Эл. адрес:
a_kozachenko@mail.ru

КУРБАНОВ Вугар Гариб оглы



Старший научный сотрудник лаборатории методов и средств автоматизации Института проблем машиноведения РАН.
В 1976 году окончил Азербайджанский государственный университет им. С. М. Кирова по специальности «Прикладная математика».
В 1983 году защитил диссертацию на соискание ученой степени кандидата физико-математических наук.
Является автором более 45 научных публикаций.
Область научных интересов — математическое моделирование процессов управления, методы логического анализа систем, логико-вероятностные методы.
Эл. адрес:
vugar_borchali@yahoo.com

ДЬЯЧУК Павел Петрович

Доцент, заведующий кафедрой математических методов физики и информационных технологий Красноярского государственного педагогического университета им. В. П. Астафьева, почетный работник высшего профессионального образования РФ.
В 1970 году окончил Красноярский государственный педагогический институт по специальности «Учитель физики».
В 1981 году защитил диссертацию на соискание ученой степени кандидата физико-математических наук.
Является автором более 170 научных публикаций и одного запатентованного изобретения.
Область научных интересов — системы автоматического управления учебной деятельностью и ее диагностика и др.
Эл. адрес:
ppdyachuk@rambler.ru

КОЛБАНЕВ Михаил Олегович

Профессор, заведующий кафедрой информационных управляющих систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича. Мастер связи.
В 1977 году окончил Ленинградский электротехнический институт связи им. проф. М. А. Бонч-Бруевича по специальности «Автоматическая электросвязь».
В 2004 году защитил диссертацию на соискание ученой степени доктора технических наук.
Является автором более 100 научных публикаций.
Область научных интересов — моделирование информационных систем.
Эл. адрес: mokolbanev@mail.ru

КУЧМИН Андрей Юрьевич

Старший научный сотрудник лаборатории механики управляемых систем Института проблем машиноведения РАН.
В 2005 году окончил Санкт-Петербургский государственный университет аэрокосмического приборостроения.
В 2007 году защитил диссертацию на соискание ученой степени кандидата технических наук.
Является автором 23 научных публикаций и двух запатентованных изобретений.
Область научных интересов — математическое моделирование в естественных науках, искусственный интеллект и принятие решений, математические проблемы теории управления и др.
Эл. адрес:
radiotelescope@yandex.ru

ЛАПСАРЬ
Алексей
Петрович

Доцент кафедры метрологии и метрологического обеспечения вооружения и техники Ростовского военного института Ракетных войск.
В 1980 году окончил Серпуховское высшее военное командно-инженерное училище по специальности «Физико-энергетические установки», в 1990 году — Военную академию им. Ф. Э. Дзержинского по специальности «Инженерная, оперативно-тактическая. Специальное вооружение».
В 2000 году защитил диссертацию на соискание ученой степени кандидата технических наук.
Является автором более 30 научных публикаций.
Область научных интересов — анализ и синтез стохастических систем на основе эволюционных моделей.
Эл. адрес: lapsar1958@ mail.ru

МИХЕЕВА
Вероника
Дмитриевна

Соискатель ученой степени кандидата физ.-мат. наук Института прикладной астрономии РАН.
В 1999 году окончила Санкт-Петербургский государственный политехнический университет.
Является автором девяти научных публикаций, соавтором четырех книг, одного запатентованного изобретения и двух описаний научно-технических разработок (ноу-хау).
Область научных интересов — языки программирования, компиляторы, инструментальные средства для систем-на-кристалле, методы автоматизации проектирования реконфигурируемых программно-аппаратных комплексов, моделирование программно-аппаратных комплексов.
Эл. адрес:
vdmikheeva@gmail.com

МОХСЕН
Аяд
Абдулазиз Али

Гражданин Йемена.
Аспирант кафедры систем автоматизированного проектирования Санкт-Петербургского государственного электротехнического университета «ЛЭТИ».
В 2008 году окончил магистратуру Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» по специальности «Информатика и вычислительная техника».
Является автором двух научных публикаций.
Область научных интересов — реализация Web-ориентированных систем автоматизированного проектирования.
Эл. адрес:
ayedh992001@hotmail.com

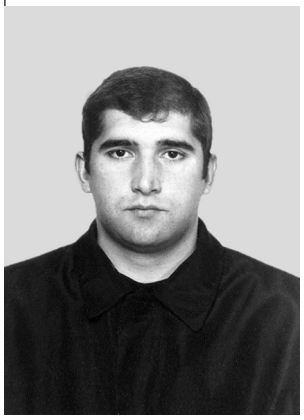
ЛАРИСТОВ
Александр
Иванович

Доцент кафедры систем автоматизированного проектирования Санкт-Петербургского государственного электротехнического университета «ЛЭТИ».
В 1974 году окончил Ленинградский электротехнический институт им. В. И. Ульянова (Ленина) по специальности «Автоматика и телемеханика».
В 1981 году защитил диссертацию на соискание ученой степени кандидата технических наук.
Является автором более 80 научных публикаций.
Область научных интересов — автоматизированные системы схемотехнического проектирования, базы данных, Web-программирование.
Эл. адрес: ailaristov@inbox.ru

МОЛДОВЯН
Дмитрий
Николаевич

Младший научный сотрудник Санкт-Петербургского института информатики и автоматизации РАН, аспирант Санкт-Петербургского государственного электротехнического университета «ЛЭТИ».
В 2009 году окончил Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» по специальности «Компьютерная безопасность».
Является автором 24 научных публикаций и четырех изобретений.
Область научных интересов — криптографические протоколы и применение конечных алгебраических структур в синтезе криптосхем с открытым ключом.
Эл. адрес: mdn.spectr@mail.ru

ОРЛОВ
Мариан
Романович



Генеральный директор компании ООО «Стилэкс», аспирант кафедры электронной коммерции и маркетинга Санкт-Петербургского государственного университета аэрокосмического приборостроения.
В 2005 году окончил Санкт-Петербургский торгово-экономический институт по специальности «Технология продуктов общественного питания».
Является автором трех научных публикаций.
Область научных интересов — системный анализ, проектное управление предприятиями и отраслевыми программами и др.
Эл. адрес: 3201650@mail.ru

ПОЛОНСКИЙ Юрий Зусьевич

Ведущий научный сотрудник лаборатории стереотаксических методов Института мозга человека им. Н. П. Бехтеревой РАН.
В 1962 году окончил Ленинградский государственный университет им. А. А. Жданова по специальности «Теория вероятностей и математическая статистика».
В 2005 году защитил диссертацию на соискание ученой степени доктора биологических наук.
Является автором более 80 научных публикаций, 5 запатентованных изобретений.
Область научных интересов — нейропсихофизиология, стереотаксическая томография.
Эл. адрес: yzpol@pochta.ru

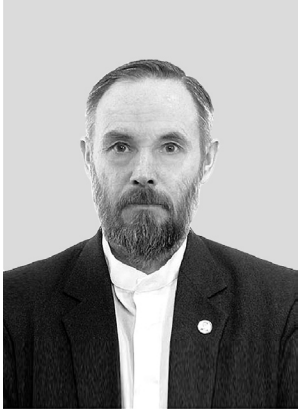
СУВОРОВ Николай Борисович

Заведующий лабораторией нейроэкологии НИИ экспериментальной медицины РАМН, профессор кафедры биотехнических систем Санкт-Петербургского государственного электротехнического университета «ЛЭТИ», действительный член Академии медико-технических наук.
В 1964 году окончил Ленинградский электротехнический институт им. В. И. Ульянова (Ленина) по специальности «Автоматика и телемеханика».
В 1993 году защитил диссертацию на соискание ученой степени доктора биологических наук.
Является автором более 270 научных публикаций.
Область научных интересов — управление в медико-биологических системах, биотехнические системы.
Эл. адрес: nbsuvorov@yandex.ru

ЦАРЕВ Владимир Александрович

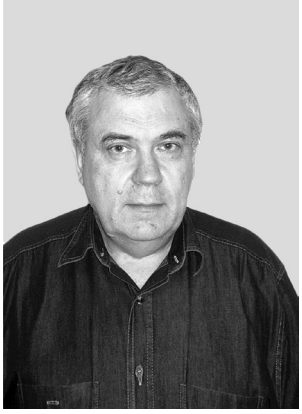
Доцент, заведующий кафедрой программного обеспечения вычислительной техники и автоматизированных систем Института менеджмента и информационных технологий филиала Санкт-Петербургского государственного политехнического университета в г. Череповце.
В 1993 году окончил Московский государственный университет им. М. В. Ломоносова по специальности «Математика, прикладная математика».
В 1998 году защитил диссертацию на соискание ученой степени кандидата технических наук.
Является автором 86 научных публикаций.
Область научных интересов — метод и средства оптоэлектронного контроля, распознавание образов и обработка изображений.
Эл. адрес: vats@imit.ru

РОГАЧЕВ Виктор Алексеевич



Доцент кафедры информационных управляющих систем Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича.
В 1977 году окончил Ленинградский электротехнический институт им. В. И. Ульянова (Ленина) по специальности «Радиотехника».
В 2009 году защитил диссертацию на соискание ученой степени кандидата технических наук.
Является автором более 30 научных публикаций и одного изобретения.
Область научных интересов — инфракрасные и оптико-информационные системы, портируемые операционные системы и программное обеспечение.
Эл. адрес: rogach@pochta.ru

ФИЛИППОВ Алексей Семенович



Доцент кафедры компьютерных систем и программных технологий Санкт-Петербургского государственного политехнического университета.
В 1973 году окончил Ленинградский политехнический институт им. М. И. Калинина.
В 1983 году защитил диссертацию на соискание ученой степени кандидата технических наук.
Является автором более 100 научных публикаций.
Область научных интересов — программируемая логика, автоматизированное проектирование встраиваемых систем.
Эл. адрес: filippov@eda-lab.ftk.spbstu.ru

ЦАРЕВ Федор Николаевич

Аспирант кафедры компьютерных технологий Санкт-Петербургского государственного университета информационных технологий, механики и оптики.
В 2009 году окончил Санкт-Петербургский государственный университет информационных технологий, механики и оптики по специальности «Прикладная математика и информатика».
Является автором 15 научных публикаций.
Область научных интересов — искусственный интеллект, машинное обучение, генетические алгоритмы, автоматное программирование.
Эл. адрес: tsarev@rain.ifmo.ru

ЧЕРНОВ
Владимир
Георгиевич

Профессор кафедры управления и информатики в технических и экономических системах Владимирского государственного университета.
В 1966 году окончил Рязанский радиотехнический институт.
В 1971 году защитил диссертацию на соискание ученой степени кандидата технических наук, в 2007 году — доктора экономических наук.
Является автором 80 научных публикаций, трех монографий, 15 запатентованных изобретений.
Область научных интересов — системы и методы поддержки принятия решений для слабоструктурированных задач, приложения аппарата нечетких множеств в исследованиях экономических процессов.
Эл. адрес:
chernov@vpti.vladimir.ru

ШАДРИН
Игорь
Владимирович

Доцент кафедры математических методов физики и информационных технологий Красноярского государственного педагогического университета им. В. П. Астафьева.
В 1997 году окончил Красноярский государственный педагогический университет.
В 2008 году защитил диссертацию на соискание ученой степени кандидата технических наук.
Является автором 25 научных публикаций и двух авторских свидетельства на разработанное программное обеспечение.
Область научных интересов — теория управления, информационные системы, автоматическое управление учебной деятельностью, компьютерная диагностика учебной деятельности.
Эл. адрес: ivsha@km.ru

ПАМЯТКА ДЛЯ АВТОРОВ

Поступающие в редакцию статьи проходят обязательное рецензирование.

При наличии положительной рецензии статья рассматривается редакционной коллегией. Принятая в печать статья направляется автору для согласования редакторских правок. После согласования автор представляет в редакцию окончательный вариант текста статьи.

Процедуры согласования текста статьи могут осуществляться как непосредственно в редакции, так и по e-mail (80x@mail.ru).

При отклонении статьи редакция представляет автору мотивированное заключение и рецензию, при необходимости доработать статью — рецензию. Рукописи не возвращаются.

Редакция журнала напоминает, что ответственность за достоверность и точность рекламных материалов несут рекламодатели.

УДК 510.647

Об одном методе вычисления вероятностей логических функций

Дубаренко В. В., Курбанов В. Г., Кучмин А. Ю. Информационно-управляющие системы, 2010. № 5. С. 2–7.

Приводится комбинаторный метод вычисления вероятностей сложных логических функций, причем элементы логических функций расположены в некотором лексикографическом порядке, т. е. нет необходимости хранить их в памяти ЭВМ в символьном виде. Описываются приближенные методы вычисления вероятности сложной логической функции по заданным вероятностям ее базисных переменных.

Ключевые слова — лингвистическая переменная, сложная логическая функция, логико-вероятностный метод.

Список лит.: 6 назв.

UDK 510.647

A method of calculating logical functions' probabilities

Dubarenko V. V., Kurbanov V. G., Kuchmin A. Y. IUS, 2010. N 5. P. 2–7.

A combinatorial method of calculating the probabilities of complex logical functions is proposed in this article. The elements of logical functions are arranged in a certain lexicographical order, i. e. there is no need to keep them in the computer memory in symbolic mode. Also, approximate methods of calculating the probability of complex logical functions according to the given probabilities of its basic variables are described.

Keywords — linguistic variable, complex logical function, logical-and-probabilistic method.

Refs: 6 titles.

УДК 519.81

Нечеткие деревья решений (нечеткие позиционные игры)

Чернов В. Г. Информационно-управляющие системы, 2010. № 5. С. 8–14.

Рассматривается решение задачи альтернативного выбора на основе нечетких деревьев решений (нечетких позиционных игр), особенностью которых является использование нечетких качественных оценок последовательности решений и состояний природы.

Ключевые слова — нечеткое множество, функция принадлежности, нечеткое дерево решений.

Список лит.: 8 назв.

UDK 519.81

Fuzzy decision trees. (Fuzzy positional games)

Chernov V. G. IUS, 2010. N 5. P. 8–14.

This article deals with solving the alternative choice problem on the basis of fuzzy decision trees (fuzzy positional games); its specific is that it uses fuzzy quality estimation of decisions consequence and nature states.

Keywords — fuzzy set, membership function, fuzzy decision tree.

Refs: 8 titles.

УДК 517.977.56, 519.876.5

Адаптивное управление дорожным движением на базе системы микроскопического моделирования транспортных потоков

Голубков А. С., Царев В. А. Информационно-управляющие системы, 2010. № 5. С. 15–19.

Описаны состав и особенности функционирования современных автоматизированных систем управления дорожным движением. Предложен способ адаптивного управления дорожным движением на основе предсказания транспортных потоков и быстрых моделей оптимизации перекрестков. Представлены характеристики системы микроскопического моделирования транспортных потоков, применяемой в системе адаптивного управления дорожным движением.

Ключевые слова — адаптивное управление дорожным движением, оптимизация управления дорожным движением, моделирование транспортных потоков, микроскопическое моделирование.

Список лит.: 18 назв.

UDK 517.977.56, 519.876.5

Adaptive traffic control based on a microscopic traffic simulation system

Golubkov A. S., Tsarev V. A. IUS, 2010. N 5. P. 15–19.

A structure and performance features of the up-to-date automatic traffic control systems are described. A way of adaptive traffic control on the basis of traffic prediction and junction fast optimization models is proposed. Characteristics of microscopic road traffic modeling engine that uses internal adaptive traffic control system are presented.

Keywords — adaptive traffic control, traffic control optimization, road traffic modeling, microscopic modeling.

Refs: 18 titles.

УДК 004.273

Архитектура Web-ориентированных САПР

Дмитревич Г. Д., Мохсен А. А., Ларистов А. И. Информационно-управляющие системы, 2010. № 5. С. 20–23.

Показана архитектура современных Web-приложений, которые представляют собой коллекцию элементов Web-узла, программно выполняющих какие-либо действия и являющихся основой Web-ориентированных САПР. Web-приложения (Web-САПР) создаются таким образом, чтобы они выполнялись на Web-серверах и использовали в качестве пользовательского интерфейса Web-браузеры. Обычно Web-приложения создаются как приложения в архитектуре «клиент—сервер», но серверная часть может иметь различные архитектурные решения. Приведем пример архитектуры конкретной Web-ориентированной САПР.

Ключевые слова — Web-приложение, Web-ориентированная САПР, архитектура Web-приложения, архитектура Web-ориентированной САПР, архитектура клиент—сервер, образовательные порталы.

Список лит.: 5 назв.

UDK 004.273

Web-oriented architecture of CAD

Dmitrevich G. D., Mohsen A. A., Laristov A. I. IUS, 2010. N 5. P. 20–23.

In this paper, we discuss the architecture of modern Web-applications that represent a collection of the elements of the Web-node, which perform some actions and are the basis of a Web-oriented CAD. Web-applications (Web-CAD) are created in a way that they are executed on Web-servers and are used as the user interface of Web browsers. Usually, Web applications are created in the «client-server» architecture, but as this paper shows, the server part has various architectural implementations. Also, we will give an example of a particular architecture of a Web-oriented CAD.

Keywords — web-application, web-oriented CAD, architecture of the web-application, web-oriented architecture of CAD, the client-server architecture, educational portals.

Refs: 5 titles.

УДК 004.4'244

Методика синтеза тестов аппаратуры по спецификациям на языке UML

Березкин А. В., Филиппов А. С. Информационно-управляющие системы, 2010. № 5. С. 24–30.

Язык UML рассматривается как язык описания спецификаций аппаратуры, из которых могут быть получены ее поведенческие тесты. Предлагается использовать данный язык в начале и в середине маршрута проектирования цифровых устройств, когда определяется их структура и функциональность на уровне последовательности управляющих воздействий. Эти спецификации являются документами, по которым создается RTL-описание устройств, а разработанная методика служит для проверки соответствия RTL-описаний UML-спецификациям. Данная проверка осуществляется путем генерации тестов устройств на основании UML-спецификаций.

Ключевые слова — UML, моделирование аппаратуры, верификация, тестирование.

Список лит.: 11 назв.

UDK 004.4'244

Hardware tests generation methodology based on UML specifications

Berezkin A. V., Filippov A. S. IUS, 2010. N 5. P. 24–30.

UML (Unified Modeling Language 2.0) is considered in this paper as a hardware specification design language, and behavioral tests of hardware can be obtained from these specifications. UML is proposed to be used in the beginning and in the middle of the design route of hardware systems, when their structure and functionality as a control signal flow is defined. These specifications are the documents that developers use to create an RTL design of a system, and the developed method is used to verify matching of an RTL and UML designs. This check is performed via test generation based on UML specifications.

Keywords — UML, hardware modeling, verification, testing.

Refs: 11 titles.

УДК 004.4'242

Метод построения управляющих конечных автоматов на основе тестовых примеров с помощью генетического программирования

Царев Ф. Н. Информационно-управляющие системы, 2010. № 5. С. 31–36.

Предлагается метод построения управляющих конечных автоматов на основе тестовых примеров с помощью генетического программирования.

Приводятся описания представления автоматов в виде особей алгоритма генетического программирования, операций мутации и скрещивания, а также генетического алгоритма. Применение метода иллюстрируется на примере построения автомата управления часами с будильником.

Ключевые слова — генетическое программирование, автоматное программирование, машинное обучение.

Список лит.: 16 назв.

UDK 004.4'242

Induction of Finite State Machines Using Genetic Programming with Fitness Based on Testing

Tsarev F. N. IUS, 2010. N 5. P. 31–36.

A method of finite state machines induction using genetic programming with fitness based on testing is described. Mutation and cross-over operations, a genetic algorithm and the structure of individuals are described. An example of this method application is given — induction of finite state machine for alarm clock control.

Keywords — genetic programming, automata-based programming, machine learning.

Refs: 16 titles.

УДК 004.43

Методы расширения языков программирования (Часть 2)

Михеева В. Д. Информационно-управляющие системы, 2010. № 5. С. 37–42.

Приводится обзор методов расширения современных языков программирования, определенных автором и использованных для построения классификации расширений по способам интеграции и исполнения кода расширений. Рассматривается метод расширения языков программирования новыми конструкциями, методы исполнения расширений, а также приводится пример предметно-ориентированного расширения языка общего назначения средствами таблично-ориентированного программирования, реализованного автором на основе средств программирования системы эфемеридных расчетов в астрономии.

Ключевые слова — предметно-ориентированный язык программирования, расширение языка программирования, инструментальные средства программирования, таблично-ориентированное программирование.

Список лит.: 18 назв.

UDK 004.43

Programming language extension methods (Part 2)

Mikheeva V. D. IUS, 2010. N 5. P. 37–42.

In this paper, an overview of the modern programming languages extension methods is presented. The methods defined by the author are used to classify programming language extensions by various kinds of the extension source code integration and execution ways. In the second part of the paper, the extension integration method with new language features and a set of extension execution methods are considered. Also considered is an example of the domain-specific language extension of a general-purpose language with the table-oriented programming features. The extension is implemented by the author on the base of the specialized programming system named ERA (Ephemeris Research in Astronomy).

Keywords — domain-specific language, programming language extension, software development tools, table-oriented programming.

Refs: 18 titles.

УДК 681.3

Примитивы криптосистем с открытым ключом: конечные некоммутативные группы четырехмерных векторов

Молдовян Д. Н. Информационно-управляющие системы, 2010. № 5. С. 43–50.

Для синтеза производительных алгоритмов распределения открытых ключей и открытого шифрования вводится новая вычислительно трудная задача над конечными некоммутативными группами. Предложен подход к построению некоммутативных групп четырехмерных векторов над простым полем и выводится формула для порядка этих групп. Описана схема согласования общего секретного ключа двух удаленных абонентов и алгоритм открытого шифрования на основе новой трудной задачи.

Ключевые слова — криптография, криптосистемы с открытым ключом, протокол открытого согласования ключа, открытое шифрование, конечные группы, некоммутативные группы, группы векторов, трудная задача.

Список лит.: 10 назв.

UDK 681.3

Primitives of the public key cryptosystems: finite non-commutative groups of four-dimension vectors

Moldovyan D. N. IUS, 2010. N 5. P. 43–50.

A new computationally difficult problem over finite non-commutative groups is proposed in this article. The problem is used to design a fast public key agreement scheme and a public encryption algorithm. An approach for constructing the finite non-commutative groups of four-dimension vectors over the finite ground field is proposed. A formula for computing the group order is derived.

Key words — cryptography, public key cryptosystem, public key agreement protocol, public encryption, finite groups, non-commutative groups, vector groups, hard problem.

Refs: 10 titles.

УДК 519.248, 621.384.3

Анализ проблемы обнаружения в инфракрасных системах

Колбанев М. О., Рогачев В. А. Информационно-управляющие системы, 2010. № 5. С. 51–54.

Проблема обнаружения сигнала в общем режиме в инфракрасных системах формулируется как задача обнаружения двухпараметрического сигнала. Применение критерия Неймана — Пирсона и принципа инвариантности позволяет получить решение — модифицированную статистику Фишера. Сравнение с известными статистиками определяет диапазоны значений сигнала и уровней помех, при которых обеспечивается максимальная вероятность правильного обнаружения.

Ключевые слова — обнаружение, инфракрасные системы, критерий Неймана — Пирсона, модифицированная статистика Фишера.

Список лит.: 6 назв.

UDK 519.248, 621.384.3

An analysis of the detection problem in infrared systems

Kolbanev M. O., Rogachev V. A. IUS, 2010. N 5. P. 51–54.

The problem of detecting the signal in general mode in infrared systems is formulated as a two parameter detection in the background of internal and external noise. The method, based on sufficient statistics, Neyman — Pearson criterion and the invariance principle, allows to obtain a solution — a modified Fisher's statistics and determine the ranges of the signal and noise levels that provide the maximum probability of correct detection.

Keywords — detection, infrared systems, Neyman-Pearson criterion, modified Fisher statistic.

Refs: 6 titles.

УДК 007.5; 681.32

Синтез быстродействующих измерительно-управляющих систем на базе параметризованных марковских моделей

Лапсарь А. П. Информационно-управляющие системы, 2010. № 5. С. 55–62.

Для синтеза измерительно-управляющих систем предложен численно-аналитический метод оценки стохастических характеристик марковской системы, описываемой эволюционными уравнениями, решения которых непрерывно зависят от вектора вещественных параметров, определяющих условия ее функционирования.

Ключевые слова — марковская параметрическая система, эволюционные уравнения, метод редукции, интерполяция, стохастические характеристики.

Список лит.: 6 назв.

UDK 007.5; 681.32

Synthesis of high speed measurement and control systems on the basis of the parameterized markovian models

Lapsar A. P. IUS, 2010. N 5. P. 55–62.

For the synthesis of the measurement and control system it has been proposed a numerical — analytical estimation method for the stochastic characteristics of the markovian system described by the evolution equations, the solutions of which depend continuously on the vector of real parameters defining the conditions of its functioning.

Keywords — markovian parametric system, evolution equations, a method reduction, interpolation, stochastic characteristics.

Refs: 6 titles.

УДК 681.52

Система автоматического управления учебной деятельностью и ее диагностики

Дьячук П. П., Дроздова Л. Н., Шадрин И. В. Информационно-управляющие системы, 2010. № 5. С. 63–69.

В рамках информационной модели развития учебной деятельности, регулируемой системой автоматического управления Tr@cK, рассмотрен процесс научения решению задач. Проведена диагностика учебной деятельности, а ее результаты сопоставлены с результатами диагностики уровня развития базовых когнитивных функций мозга.

Ключевые слова — системы управления, автоматическое регулирование, диагностика учебной деятельности.

Список лит.: 6 назв.

UDK 681.52

An automatic regulative system of learning activity and learning diagnostics

Dyachuk P. P., Drozdova L. N., Shadrin I. V. IUS, 2010. N 5. P. 63–69.

Task-solving learning activity is considered by using the informational model of learning activity, moderated by the automatic regulative «Tr@ck» system. Learning activity diagnostics is done, and its results are compared with the results of the diagnostics of basic cognitive brain functions.

Keywords — automatic regulation, automatic systems, learning activity diagnostics.

Refs: 6 titles.

УДК 681.2; 615.47

Биотехническая система для исследования интеллектуальной деятельности человека

Суворов Н. Б., Абрамов В. А., Козаченко А. В., Полонский Ю. З. Информационно-управляющие системы, 2010. № 5. С. 70–77.

Разработана и испытана в реальных исследованиях биотехническая система для изучения психофизиологических механизмов напряженной интеллектуальной деятельности (во время игры в шахматы). Участвовали шахматисты высшей квалификации (коэффициент Эло ≥ 2300). При разработке решена главная задача: психофизиологические параметры играющего с шахматной программой синхронизированы с текущей позицией на шахматной доске.

Ключевые слова — интеллектуальная деятельность, психофизиологические параметры, шахматисты.

Список лит.: 3 назв.

UDK 681.2; 615.47

A Bioengineering System of the Research of Human Intellectual Activity

Suvorov N. B., Abramov V. A., Kozachenko A. V., Polonsky Yu. Z. IUS, 2010. N 5. P. 70–77.

A bioengineering system to study the psycho physiological mechanisms of intensive intellectual activity (a game of chess) is developed and tested in real-life researches. The participants were top rate chess players (Elo coefficient ≥ 2300). The main task (psycho physiological parameters of players are synchronized with the current position on a chessboard) is solved.

Keywords — intellectual activity, psycho physiologic parameter, chess players.

Refs: 3 titles.

УДК 615.471:617.7

Автоматическое выделение участков электрокардиосигнала с нормальным синусовым ритмом

Бахилин В. М. Информационно-управляющие системы, 2010. № 5. С. 78–84.

Предложен самонастраивающийся алгоритм выделения фрагментов с нормальным синусовым ритмом в длинных записях ЭКГ, численные параметры которого рассчитаны по результатам анализа записей Физиобанка (PhysioBank).

Ключевые слова — автоматический анализ ЭКГ, вариабельность сердечного ритма, вариабельность RT-, TP-, PR- и PT-интервалов.

Список лит.: 7 назв.

UDK 615.471:617.7

An automatic detection of electrocardiogram sections with the normal sinus rhythm

Bahilin V. M. IUS, 2010. N 5. P. 78–84.

A new self-tuning (adaptive) algorithm for the detection of fragments with normal sinus rhythm in long-term Holter recordings of electrical cardiac signals is proposed. PhysioBank recordings were analyzed to estimate numerical parameters of the algorithm.

Keywords — ECG automatic analysis, heart rate variability, RT-, TP-, PR- and PT- intervals variability.

Refs: 7 titles.

УДК 334.021.1

Некоторые проблемы институализации государственно-частного партнерства

Орлов М. Р. Информационно-управляющие системы, 2010. № 5. С. 85–90.

Рассмотрены проблемы институализации проектов государственно-частного партнерства в части формализации процессов принятия решения и управления такими проектами. Проанализировано состояние отечественной законодательной базы, связанной с реализацией государственно-частного партнерства. Показано, что наиболее эффективным инструментом согласования интересов и управления проектами государственно-частного партнерства служит система сбалансированных показателей.

Ключевые слова — проектное управление, государственно-частное партнерство, система сбалансированных показателей.

Список лит.: 8 назв.

UDK 334.021.1

Some problems of public-private institualization

Orlov M. R. IUS, 2010. N 5. P. 85–90.

A problem of the public-private partnership project management as part of the making decision processes formalization and such project control has been considered. A native legislative base status entailed with the public-private partnership realization has been analyzed. It is shown that the most effective tool for the interest concordance and the public-private partnership project control is the balanced scorecard.

Keywords — project control, public-private partnership, balanced scorecard.

Refs: 8 titles.

УВАЖАЕМЫЕ АВТОРЫ!

При подготовке рукописей статей редакция просит Вас руководствоваться следующими рекомендациями.

Объем статьи (текст, таблицы, иллюстрации и библиография) не должен превышать эквивалента в 16 страниц, напечатанных на бумаге формата А4 на одной стороне через 1,5 интервала в Word шрифтом Times New Roman размером 13.

Обязательными элементами оформления статьи являются: индекс УДК, заглавие, инициалы и фамилия автора (авторов), ученая степень, звание, полное название организации, аннотация (7–10 строк) и ключевые слова на русском и английском языках, подрисуночные подписи.

Формулы в текстовой строке набирайте в Word, не используя формульный редактор (Mathtype или Equation), только в том случае, если средства Word не позволяют набрать формулу или символ (например, простая дробь, символы с «крышками» и т. д.), используйте имеющийся в Word формульный редактор Mathtype или Equation; формулы, стоящие в отдельной строке, могут быть набраны как угодно; при наборе формул в формульном редакторе знаки препинания, ограничивающие формулу, набирайте вместе с формулой; для установки размера шрифта никогда не пользуйтесь вкладкой Other..., используйте вкладку Define; в формулах не отделяйте пробелами знаки: + = –.

Для набора формул в Word никогда не используйте Конструктор (на верхней панели: «Работа с формулами» — «Конструктор»), т. к. этот ресурс предназначен только для внутреннего использования в Word и не поддерживается программами, предназначенными для изготовления оригинал-макета журнала.

При наборе символов в тексте помните, что символы, обозначаемые латинскими буквами, набираются светлым курсивом, русскими и греческими — светлым прямым, векторы и матрицы — прямым полужирным шрифтом.

Иллюстрации в текст не заверстываются и предоставляются отдельными исходными файлами, поддающимися редактированию:

— рисунки, графики, диаграммы, блок-схемы изготавливаются в векторных программах: Visio 4, 5, 2002–2003 (*.vsd); Coreldraw (*.cdr); Excel; Word; AdobeIllustrator; AutoCad (*.dxf); Компас; Matlab (экспорт в формат *.ai);

— фото и растровые — в формате *.tif, *.png с максимальным разрешением (не менее 300 pixels/inch).

В редакцию предоставляются:

— сведения об авторе (фамилия, имя, отчество, место работы, должность, ученое звание, учебное заведение и год его окончания, ученая степень и год защиты диссертации, область научных интересов, количество научных публикаций, домашний и служебный адреса и телефоны, факс, эл. адрес), фото авторов: анфас, в темной одежде на белом фоне, должны быть видны плечи и грудь, высокая степень четкости изображения без теней и отблесков на лице, фото можно представить в электронном виде в формате *.tif, *.png с максимальным разрешением — не менее 300 pixels/inch при минимальном размере фото 40 × 55 мм;

— экспертное заключение.

Список литературы составляется по порядку ссылок в тексте и оформляется следующим образом:

— для книг и сборников — фамилия и инициалы авторов, полное название книги (сборника), город, издательство, год, общее количество страниц;

— для журнальных статей — фамилия и инициалы авторов, полное название статьи, название журнала, год издания, номер журнала, номера страниц;

— ссылки на иностранную литературу следует давать на языке оригинала без сокращений;

— при использовании web-материалов указывайте адрес сайта и дату обращения.

Более подробную информацию см. на сайте: www.i-us.ru

Журнал "ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ"

выходит каждые два месяца.
Стоимость годовой подписки (6 номеров)
для подписчиков России – 3600 рублей,
для зарубежных подписчиков – 4200 рублей,
включая НДС 18%, таможенные и почтовые расходы.

На электронную версию нашего журнала (все выпуски, годовая подписка, один выпуск, одна статья), начиная с №1, 2002 г. и далее, вы можете подписаться на сайте РУНЭБ: http://www.elibrary.ru.

Подписку на печатную версию журнала можно оформить в любом отделении связи по каталогам:
«Роспечать»: № 48060 – годовой индекс,
№ 15385 – полугодовой индекс;
«Пресса России» – № 42476,
а также посредством:
- «Северо-Западное Агентство "Прессинформ"»
Эл. почта: press@crp.spb.ru, zajavka@crp.spb.ru
Сайт: http://www.pinform.spb.ru
- «Издательский дом «Экономическая газета»:
Эл. почта: arpk@akdi.ru, izdatcat@eg-online.ru
- «МК-Периодика» (РФ + 90 стран)
Эл. почта: export@periodicals.ru Сайт: http://www.periodicals.ru
- «Артос-Гал»
Сайт: http://www.artos-gal.mpi.ru/index.html
- «Интерпочта»
Эл. почта: interpochta@interpochta.ru Сайт: http://www.interpochta.ru
- «Информнаука» (РФ + ближнее и дальнее зарубежье)
Эл.почта: Alfimov@viniti.ru Сайт: http://www.informnauka.com
- «Агентство "Газеты в розницу"» (Екатеринбург)
Эл. почта: box@e-rospechat.ru Сайт: http://e-rospechat.ru
- «Коммерсант-Курьер» (Казань)
Эл. почта: kazan@komcur.ru Сайт: http://www.komcur.ru/contacts/kazan/
- «Урал-Пресс» (филиалы в 40 городах РФ)
Сайт: http://www.ural-press.ru
- «Идея» (Украина)
Сайт: http://idea.com.ua
- «BTL» (Узбекистан)
Сайт: http://btl.sk.uz/ru/cat17.html
и др.

Возможно оформление редакционной подписки, как на текущий год, так и на все вышедшие в свет номера журнала, по заявке организации или частного лица:
по эл. почте: 80x@mail.ru
по телефону: (812) 494-70-44
по факсу: (812) 494-70-18 (с пометкой «Для РИЦ»)
по почте: 190000, Санкт-Петербург, Б. Морская ул., д. 67, ГУАП, РИЦ, Редакция журнала "Информационно-управляющие системы"

После оплаты счета мы высылаем заказанные номера журнала. При необходимости высылаем журнал наложенным платежом.

ISSN 1684-8853

ИНФОРМАЦИОННО-УПРАВЛЯЮЩИЕ СИСТЕМЫ