ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

4(200) 2013

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

Информация о журнале доступна по сети Internet по адресу **http://novtex.ru/IT.**
Журнал включен в систему Российского индекса научного цитирования.
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

УДК 621.3.049.77.001.2:004.3

И. А. Конников, д-р техн. наук,

e-mail: konnikov_i@mail.ru

Математическое моделирование перекрестной помехи в САПР

Излагаются основные идеи построения и реализации математической модели для задачи, имеющей большое прикладное значение. Для оценки перекрестных помех предлагается использовать модифицированный метод эквивалентной постоянной распространения. В соответствии с этим методом функция Грина, которая является решением волнового уравнения, для слоистой среды описывается выражением того же вида, что и для однородной среды. При таком подходе наводимая ЭДС помехи может быть рассчитана с помощью интегрирования всех составляющих (а не только статической составляющей) электромагнитного поля источника помехи.

Задача сводится к вычислению пятикратного интеграла от функции Грина для волнового уравнения. Предлагаются методы дискретизации подынтегральной функции для ее численного интегрирования. Для мониторинга шага интегрирования предлагается информационный подход на основе теоремы Котельникова.

Ключевые слова: *функция Грина, математическое моделирование, эквивалентная постоянная распространения, наводки*

Введение

Тенденции развития современной электроники обусловливают постоянно возрастающую актуальность разработки и внедрения все более эффективных методов вычисления перекрестной помехи в электронном модуле.

Одной из основных причин, по которым подлежащий решению комплекс задач пока не получил должного эффективного решения, явилось то, что темпы роста возможностей технического обеспечения систем научных и инженерных расчетов и систем автоматизированного проектирования (САПР) не смогли стать выше темпов роста размерности и сложности решаемых задач даже с учетом возможности декомпозиции последних. По-видимому, такая ситуация будет иметь место и дальше. Имеется настоятельная необходимость повышения размерности задач моделирования, решение которых возможно широко доступными вычислительными средствами.

Приемлемое решение следует искать на пути прямого использования методов теории электромагнитного поля, органично учитывающих распределенный характер конструктива и позволяющих построить математические модели, область корректного использования которых отвечает тенденциям развития электроники.

Основная идея предлагаемого подхода

Для математического моделирования перекрестных помех предлагается использовать электродинамический подход на основе метода эквивалентной постоянной распространения, описанный в работе [1]. При таком подходе значение наводимой ЭДС помехи является интегральной характеристикой системы, состоящей из источника, рецептора помехи и канала паразитной связи. Для количественной оценки помехи пространственная дискретизация такой системы совершенно не обязательна. В этом случае функция Грина $G_в$, которая является решением волнового уравнения, для слоистой среды описывается выражением того же вида, что и для однородной среды:

$$G_в = M\exp(-ik_{эпр}R_0)/R_0,$$

где M — амплитудный множитель; $k_{эпр} = \omega\sqrt{\varepsilon_0\varepsilon_э\mu_0\mu_э}$ — эквивалентная постоянная распространения; R_0 — радиус в сферической системе координат; константа Кулона $\varepsilon_0 = 10^{-9}/(36\pi)$; константа Био—Савара $\mu_0 = 4\pi\cdot 10^{-7}$; $\varepsilon_э$ и $\mu_э$ — эквивалентные относительные диэлектрическая и магнитная проницаемости слоистой среды соответственно; ω — угловая частота.

Значения $\varepsilon_э$ и $\mu_э$ целесообразно рассчитывать по единым для каждого слоя среды более простым формулам, отличным от предлагаемых в работе [1]:

$$\varepsilon_э(r) = \frac{1}{R_0}\Big/\int_0^\infty J_0(\lambda r)q_\varepsilon(\lambda)d\lambda;\ \mu_э(r) = R_0\int_0^\infty J_0(\lambda r)q_\mu(\lambda)d\lambda,$$

где $q_\varepsilon(\lambda)$ — полученная при решении электростатической задачи математическая модель слоистой среды, которая соответствует конструкции электронного модуля [2]; $q_\mu(\lambda)$ — полученная при решении магнитостатической задачи математическая модель той же среды; J_0 — функция Бесселя первого рода

нулевого порядка; $r = \sqrt{(x-x_0)^2 + (y-y_0)^2}$ — длина парциального канала связи; x, y, z — абсцисса, ордината и аппликата точки, где вычисляется поле; x_0, y_0, z_0 — абсцисса, ордината и аппликата точки, где расположен элементарный источник поля; несобственные интегралы вычисляются по методике, описанной в работе [3].

При таком определении величин $\varepsilon_э$ и $\mu_э$ они не зависят от размеров проводника, что существенно снижает объем необходимых вычислений.

Формально задача вычисления помехи сводится к вычислению четырехкратного интеграла от функции Грина для волнового уравнения [1], причем для вычисления вихревой составляющей помехи функцию Грина следует проинтегрировать еще раз. В системе ортогонального монтажа интегрирование приходится проводить численно в прямоугольных координатах. Формулы численного интегрирования представляют собой суммы дискретных значений функции, подлежащей интегрированию, с соответствующими весами. Конкретизацию способов интегрирования и, в частности, оценку требуемых параметров дискретизации подынтегральной функции проведем в прямоугольных координатах на примере потенциала электрического поля; для потенциала магнитного поля интеграл вычисляется аналогично, если это не отмечено особо.

Таким образом, математическая модель задачи описывается аналитическим выражением для усредненного по толщине проводника t потенциала электрического поля, создаваемого проводником-источником длиной l [1]:

$$\varphi(x, y) = M_\varphi \int_0^t dz \int_0^t dz_0 \int_0^b \eta(y_0) dy_0 \int_{x_1}^{x_1+l} \frac{\exp(-i\psi)}{R_0 \varepsilon_э(r)}\, dx_0, \quad (1)$$

где $i^2 = -1$; ψ — аргумент (фаза) подынтегральной функции; M_φ — амплитудный множитель; x_1 — абсцисса начала проводника; b — ширина проводника; проводник ориентирован вдоль оси абсцисс.

Интегрирование по площади поперечного сечения

Поперечное распределение заряда $\eta(y_0)$ в формуле (1) целесообразно описывать дельта-функцией Дирака [4]; в этом случае интегрирование по y_0 проводится аналитически и затруднений не вызывает.

Учет неравномерности распределения тока и заряда по толщине в проводнике вообще мало влияет на создаваемое им в азимутальном направлении поле помехи, поэтому распределения тока и заряда в тонком проводнике можно считать равномерными по толщине. Тогда двукратное интегрирование по z и z_0 целесообразно проводить по приближенной кубатурной формуле

$$\int_0^l dx_0 \int_0^l dx \int_0^t dz_0 \int_0^t \frac{dz}{\sqrt{(x-x_0)^2 + b^2 + (z-z_0)^2}} \approx$$
$$\approx t^2 \int_0^l dx \int_0^l \frac{dx_0}{\sqrt{(x-x_0)^2 + b^2 + s_t^2}}, \quad (2)$$

где $s_t = t\exp(-3/2)$ — среднее геометрическое расстояние отрезка прямой, который имеет длину t, от самого себя.

Эта формула представляет собой математическую интерпретацию и обобщение метода средних геометрических расстояний, предложенного Максвеллом в монографии [5, § 691]. Вербальная формулировка названного метода приведена в работе [6, с. 31]. Погрешность кубатурной формулы (2) была исследована в [7, 8]. Часть результатов исследования представлена в таблице. Эксперимент показал, что метод средних геометрических расстояний несколько завышает результат (относительная погрешность $\Delta > 0$). Из таблицы видно, что для проводников, длина которых l на два десятичных порядка превосходит толщину t, при расстоянии между ними $b > 3t$ формула дает практически точные результаты. Учитывая, что и на печатной плате, и в микросхеме длина проводника обычно на 2...4 десятичных порядка превосходит его толщину, можно полагать, что использование формулы (2) позволяет обеспечить пренебрежимо малую погрешность при указанном соотношении размеров области интегрирования.

Выполнив интегрирование по z и z_0 с помощью кубатурной формулы (2), а интегрирование по y_0 — аналитически, по формуле (1) получим, что потен-

Относительная погрешность кубатурной формулы (2), %

b/t \ l/b	1	10	100	1000
1	7,15940	4,07044	2,37576	1,63597
3	3,77252	1,69474	$9{,}84939 \cdot 10^{-1}$	$6{,}84585 \cdot 10^{-1}$
10	1,37278	$5{,}48602 \cdot 10^{-1}$	$3{,}17077 \cdot 10^{-1}$	$2{,}21033 \cdot 10^{-1}$
100	$1{,}52316 \cdot 10^{-1}$	$5{,}66105 \cdot 10^{-2}$	$3{,}25460 \cdot 10^{-2}$	$2{,}27061 \cdot 10^{-2}$
1000	$1{,}54845 \cdot 10^{-2}$	$5{,}68230 \cdot 10^{-3}$	$3{,}26316 \cdot 10^{-3}$	$2{,}27672 \cdot 10^{-3}$
10 000	$1{,}55425 \cdot 10^{-3}$	$5{,}69291 \cdot 10^{-4}$	$3{,}26987 \cdot 10^{-4}$	$2{,}28881 \cdot 10^{-4}$

циал электрического поля, создаваемого проводником-источником,

$$\varphi(x, y) = M_{\varphi} \int_{x_1}^{x_1 + l} \frac{\exp[-i\psi(r)]}{\varepsilon_{э}(r) R}\, dx_0, \qquad (3)$$

где $R = \sqrt{(x - x_0)^2 + (y - y_0)^2 + s^2}$; $s = t\exp(-3/2)$ — среднее геометрическое расстояние отрезка прямой, который имеет длину t, от самого себя.

Дискретизация по Котельникову

Как отмечено выше, формулы численного интегрирования представляют собой суммы дискретных значений функции, подлежащей интегрированию, с соответствующими весами. Рассмотрим вычисление внутреннего интеграла по dx_0 в (1). Очевидно, что первостепенное значение при этом имеет выбор шага интегрирования как с точки зрения достижимой точности, так и в отношении времени счета. Часть результатов вычислительного эксперимента по выявлению характера зависимостей модуля подынтегральной функции от нормированного расстояния $(x - x_0)/h$ представлена на рис. 1 (h — толщина платы). Эти зависимости получены для проводников толщиной $t = 0{,}1$ мм при относительной диэлектрической проницаемости платы $\varepsilon_2 = 5{,}5$ (стеклотекстолит и др.) и являются типовыми. С точностью до постоянного множителя они показывают усредненное по толщине проводника затухание в парциальном канале связи потенциала, создаваемого источником поля в виде отрезка прямой длиной t, расположенным в точке $x_0 = x$, в зависимости от расстояния до точки, где вычисляется поле. Из рис. 1 видно, что модуль подынтегральной функции имеет острый пик.

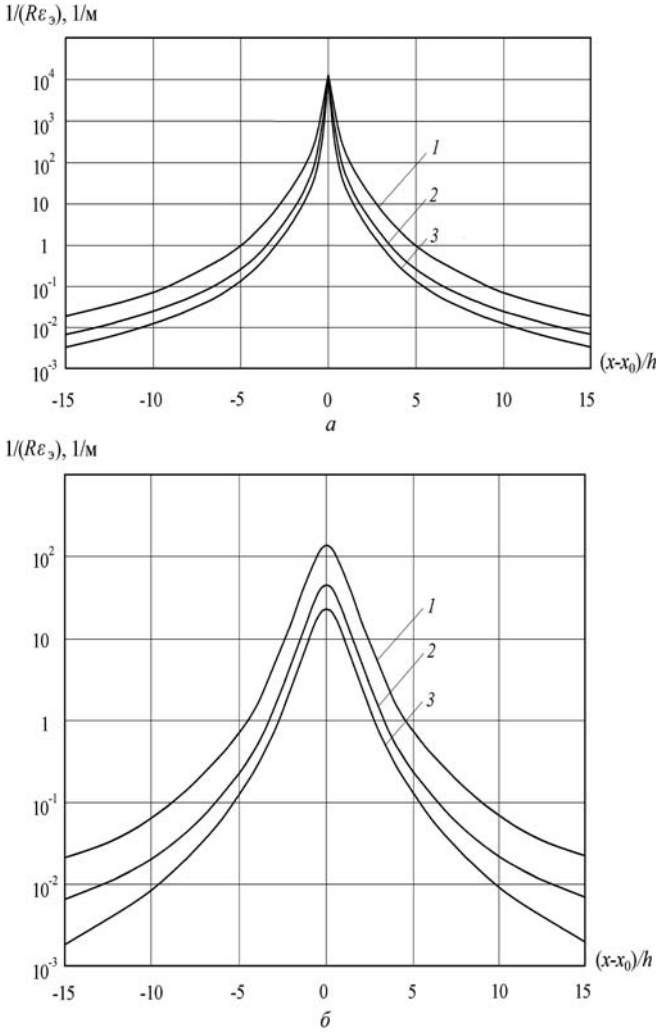


Рис. 1. Зависимость модуля подынтегральной функции от нормированного расстояния $(x - x_0)/h$ при фиксированных толщине платы h и разности ординат $|y - y_0|$:
а — случай расчета поля в объеме проводника-источника, т. е. при $|y - y_0|/h = 0$; *б* — случай взаимовлияния проводников при $|y - y_0|/h = 1$; *1* — $h = 1$ мм; *2* — $h = 3$ мм; *3* — $h = 6$ мм

С формальной точки зрения можно полагать, что подынтегральное выражение описывает функцию с ограниченным спектром пространственных гармоник. Некоторым ориентиром, дающим возможность оценить требуемый шаг интегрирования, является значение шага, получаемое на основе информационного подхода с помощью известной теоремы Котельникова [9]. Разумеется, указанная теорема была доказана для процесса, который является функцией времени, однако математическая формулировка теоремы Котельникова (как и значение интеграла) не зависит ни от обозначения переменной, ни от ее физического смысла.

Информационный подход по Котельникову реализуется во многих технических приложениях. Как показала многолетняя успешная практика дискретизации в системах цифрового управления, где выбор частоты дискретизации проводится по ширине полосы пропускания системы [10], используемые частоты дискретизации обычно существенно выше наименьшего значения, допустимого по теореме Котельникова. Обычно в системах цифрового управления проводится от 2 до 4 дискретных отсчетов за время регулирования, в противном случае качество системы будет резко ухудшаться [11]. В решаемой задаче значение шага интегрирования, вычисленное строго по теореме Котельникова описанным в работе [10] способом, можно рассматривать как граничное, превышение которого принципиально лишает возможности рассчитать потенциал поля, создаваемого проводником-источником, без существенной потери точности. Более подробно дискретизация по Котельникову рассмотрена в [10].

Дискретизация по критерию репрезентативности

Значение шага дискретизации с точки зрения достижимой точности по модулю интеграла можно оценить из следующих соображений.

Очевидно, что шаг дискретизации не может превышать ширину пика модуля подынтегральной функции. На рис. 2 показано как изменяется модуль интеграла функции Грина при изменении ширины интервала интегрирования для интегрирования за один шаг. С учетом формулы (3) это изменение характеризуется величиной

$$\alpha = \left| \int_{x-d}^{x+d} \frac{\exp[-i\psi(r)]}{\varepsilon_э(r) R} dx_0 \right| \Big/ \left| \int_{x-l/2}^{x+l/2} \frac{\exp[-i\psi(r)]}{\varepsilon_э(r) R} dx_0 \right|,$$

где l — длина проводника-источника; d — половина интервала интегрирования.

Как показал вычислительный эксперимент, с уменьшением значений h, ε_2 и увеличением t значение α снижается. Кривые *1* и *3* на рис. 2 показывают границы, в которых изменяется величина α при произвольном сочетании значений $h \in [1, 6]$ мм, $\varepsilon_2 \in [2.5, 5.5]$, $t \in [35, 100]$ мкм. При удвоении значений указанной на рис. 2 длины проводника при одном и том же значении полуинтервала интегрирования d значение α сохраняет 3...4 десятичных знака мантиссы, что естественно вследствие быстрого убывания модуля подынтегральной функции по мере удаления от точки $x_0 = x$. Отсюда следует, что требование интегрирования функции Грина по всей длине проводника-источника является совершенно избыточным вследствие неоправданного увеличения времени счета. Необходимо разумное ограничение интервала интегрирования: он должен быть минимальным, оставаясь репрезентативным. Интервал интегрирования можно считать репрезентативным, если его расширение до значения l не приводит к существенному изменению значения интеграла.

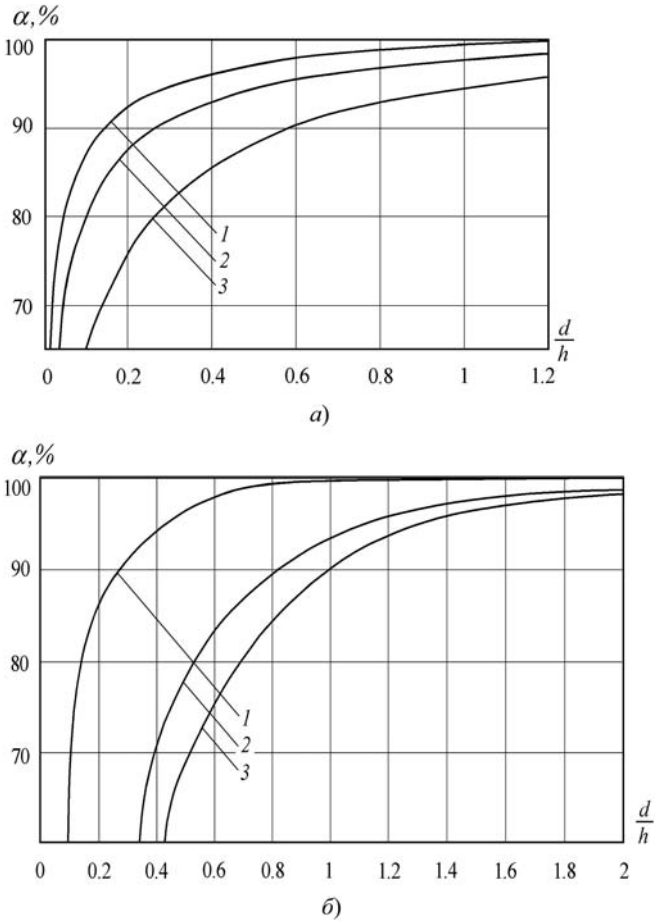


Рис. 2. Доля модуля интеграла функции Грина вдоль проводника-источника на интервале 2*d* в сравнении с модулем того же интеграла вдоль всего проводника длиной *l*:
а — $y = y_0$, длина проводника 6 см; *б* — $y = y_0 + h$, длина проводника 15 см; *1* — $h = 6$ мм, $\varepsilon_2 = 5{,}5$, $t = 35$ мкм; *2* — $h = 3$ мм, $\varepsilon_2 = 4$, $t = 35$ мкм; *3* — $h = 1$ мм, $\varepsilon_2 = 2{,}5$, $t = 100$ мкм

Подынтегральная функция при интегрировании поля по объему проводника-источника является предельным случаем расчета поля помехи при $y \to y_0$. При $y \neq y_0$ подынтегральная функция не имеет столь выраженного пика, поэтому допустимые значения рабочего интервала и шага интегрирования будут выше.

С увеличением расстояния между проводниками для достижения репрезентативности результата требуемый рабочий интервал интегрирования при относительно малых h (например, для однослойных плат) должен заметно расширяться, что отражено на рис. 2, *б*. Увеличение интервала может, на первый взгляд, потребовать больше машинного времени для обеспечения той же погрешности интегрирования. Однако на практике этого не происходит, поскольку с увеличением расстояния между проводниками пик модуля подынтегральной функции становится менее острым, что позволяет увеличить шаг интегрирования.

Поскольку при заданном значении α и данном расстоянии $|y - y_0|$ требуемый интервал интегрирования является единым для всех проводников платы, вычисление указанного интервала целесообразно проводить в автоматическом режиме и для заданного уровня α аппроксимировать зависимость требуемого интервала от расстояния $|y - y_0|$ степенным многочленом второй степени.

Дискретизация с использованием компенсирующей поправки

С точки зрения повышения быстродействия и точности расчетов подход к дискретизации и выделению рабочего интервала интегрирования должен быть иным. При решении проектных задач большой размерности приоритетная ориентация на достижение максимально возможной точности, обусловливающая повышенный расход машинного времени, обычно нецелесообразна: предпочтительна ориентация на снижение времени счета при выполнении ограничений по уровню погрешности. Расчет перекрестных помех в электронном модуле — задача, предполагающая немалый объем вычислений, и

добиваться высокой точности на каждом этапе вычислений в данном случае нецелесообразно. Наоборот, оказалось целесообразным несколько "загрубить" вычисления, снизив число шагов интегрирования до единицы[1], используя формулу численного интегрирования низкого порядка на три узла и оптимизируя рабочий интервал интегрирования.

Очевидно, что погрешность вычисления интеграла имеет две основные составляющие. Одна — парциальная погрешность — обусловлена уменьшением формально необходимого интервала интегрирования $[x - l/2, x + l/2]$ до рабочего значения $[x - d, x + d]$, а другая — погрешностью используемой квадратурной формулы. Как показал вычислительный эксперимент с использованием квадратурной формулы Гаусса на три узла, погрешность вычисления модуля на участках относительно быстрого изменения модуля подынтегральной функции положительна, а парциальная погрешность, обусловленная сужением интервала интегрирования, отрицательна; погрешность округления при проведении вычислительного эксперимента заведомо была пренебрежимо мала. Как оказалось, рабочий интервал интегрирования можно выбирать таким образом, чтобы погрешность квадратурной формулы была хотя бы частично компенсирована погрешностью, которая вызвана сужением интервала. Для этого сужение формально необходимого интервала $[x - l/2, x + l/2]$ должно проводиться таким образом, чтобы оно давало стабильную ошибку до 4...6 %, занижающую значение модуля интеграла. Эта ошибка может рассматриваться как некая компенсирующая поправка, заметно повышающая точность вычислений; в некоторых случаях ее допустимое значение может быть повышено до уровня 15...20 %. Интервал интегрирования и значение α, соответствующие полностью взаимно компенсированным парциальным погрешностям, условно назовем оптимальными.

Поскольку оптимальный интервал интегрирования является единым для всех проводников[2] данного слоя коммутации, разнесенных на расстояние $|y - y_0|$, на каждой рабочей частоте его можно вычислять один раз. Оптимизация поправки χ легко формализуется и проводится в автоматическом режиме для каждого слоя коммутации с помощью специальной подпрограммы. Целью оптимизации при этом является не просто повышение точности расчетов, а прежде всего снижение времени счета при одновременном повышении точности или незначительном ее понижении. В результате оптимизации

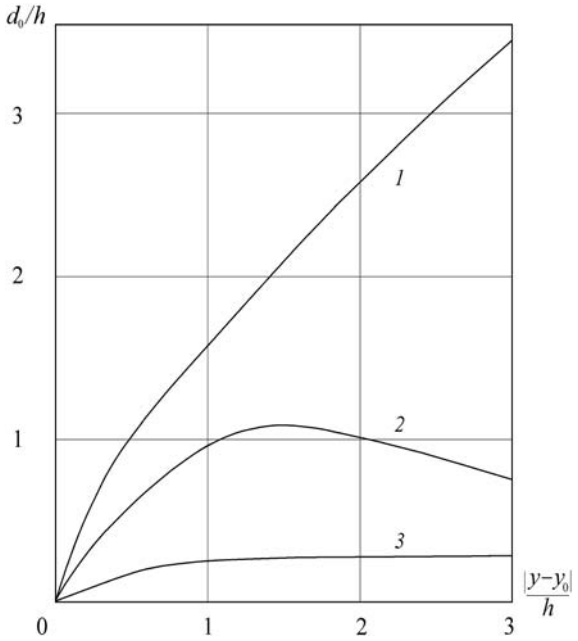


Рис. 3. Зависимость оптимального полушага интегрирования от нормированного расстояния между проводниками при толщине диэлектрика $h = 2$ мм и фиксированной его относительной диэлектрической проницаемости ε_2 на частоте $\omega = 10^{10}$ рад/с:
1 — $\varepsilon_2 = 2{,}5$; *2* — $\varepsilon_2 = 4{,}7$; *3* — $\varepsilon_2 = 5{,}5$

должна быть получена функция[3], аппроксимирующая зависимость указанного интервала $2d_0$ от расстояния $|y - y_0|$. При невозможности получения такой аппроксимирующей функции можно воспользоваться приведенными выше экспресс-оценками или же просто принять значение интервала интегрирования для достижения репрезентативности результата равным $\pm 1{,}5h$.

На рис. 3 представлена часть результатов вычислительного эксперимента по исследованию зависимости оптимального полуинтервала интегрирования d_0 от расстояния между проводниками $|y - y_0|$ на частоте 10^{10} рад/с при описанном выше способе задания интервала интегрирования с использованием компенсирующей поправки. В эксперименте использовалась квадратурная формула Гаусса на три узла. По результатам эксперимента можно сделать вывод о том, что оптимальное значение α, минимизирующее погрешность вычислений, как правило, является еще и вполне репрезентативным. От расстояния между проводниками это значение заметно зависит лишь на малых расстояниях $|y - y_0| < 5t$, что может оказаться существенным не только при расчете полей в микросхемах, но и в случае печатных плат. Однако с уменьшением расстояния между проводниками (на расстояниях порядка 0,2 мм и менее) оптимальное значение α заметно снижается и становится менее репрезентативным.

[1] Такое значение шага интегрирования не всегда удовлетворяют ограничения, налагаемые на основе теоремы Котельникова. Так, на частоте $\omega = 10^{10}$ рад/с запас составляет 1...2 десятичных порядка. С ростом частоты запас снижается, и на частотах $\omega > 10^{11}$ рад/с указанные ограничения могут вступить в силу, что потребует увеличения числа шагов.

[2] При условии, что длина проводника превосходит интервал интегрирования или равна ему.

[3] Фактически должны быть получены числовые значения коэффициентов аппроксимирующего многочлена второй степени.

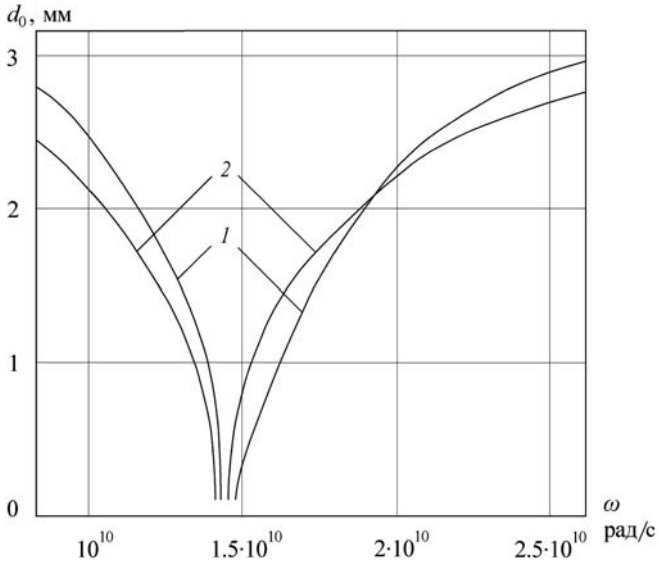


Рис. 4. Частотная зависимость оптимального полуинтервала интегрирования при $h = 2$ мм:
1 — для векторного потенциала магнитного поля; *2* — для потенциала электрического поля; $|y - y_0|/h = 1{,}5$; $\varepsilon_2 = 2{,}5$

Следует особо отметить, что значение оптимального интервала интегрирования является частотнозависимым. На рис. 4 представлен фрагмент типичной частотной зависимости оптимального полуинтервала. Из рисунка видно, что имеются частоты, вблизи которых точность квадратурной формулы, используемой для интегрирования пика, снижается и ширина оптимального полуинтервала падает. На таких частотах погрешность используемой квадратурной формулы "перекомпенсирует" вторую парциальную погрешность, обусловленную уменьшением формально необходимого интервала интегрирования до рабочего значения. Соответствующее значение α также падает, а процедура вычисления значения d_0 становится менее устойчивой. В этом случае возникает необходимость увеличения числа шагов интегрирования пика или перехода на квадратурную формулу более высокого порядка и/или другого типа. Возможен также переход на квазиоптимальное значение интервала интегрирования, при котором парциальные погрешности взаимно компенсируются лишь частично. Индикатором необходимости такого перехода может служить снижение значения α до некоторого критического уровня, заданного заранее и признаваемого нерепрезентативным. Выбор типа и порядка квадратурной формулы и необходимого числа шагов может проводиться для каждого слоя коммутации на дискретном рабочем поле один раз в автоматическом режиме до начала проектирования. Все это может свидетельствовать о некоторой искусственности и возможной нефизичности метода компенсирующей поправки; однако принципиально возможная нефизичность метода вполне искупается тем, что принимаемые при корректном использовании этого метода параметры интегрирования обеспечивают высокое быстродействие и высокую точность вычислений. Тем не менее, на стадии предпроектных исследований[4] требование надежной физической трактовки результатов расчета может оказаться доминирующим, и тогда выбор шага интегрирования по критерию репрезентативности может оказаться более надежным и поэтому предпочтительным. При любом способе выбора шага интегрирования необходимо контролировать рабочее значение шага с помощью теоремы Котельникова, как предлагается выше.

Интегрирование методом компенсирующей поправки на оптимальном интервале интегрирования за один шаг позволяет добиться точности, достигаемой при интегрировании с постоянным шагом на интервале $[x - l/2, x + l/2]$ за 250...300 шагов, если в обоих случаях используется квадратурная формула Гаусса на три узла[5]. Очевидно, что уменьшение числа шагов интегрирования влечет за собой соответствующее увеличение быстродействия программного обеспечения, причем без снижения точности расчетов.

Организация вычислений

Как показал вычислительный эксперимент, значения шага интегрирования, рассчитанные по различным критериям, могут значительно различаться, особенно учитывая широкий диапазон возможных рабочих частот, поэтому в качестве рабочего значения шага следует принять меньшее из указанных значений.

Учитывая успешную многолетнюю практику дискретизации в системах цифрового управления [11], при вычислении интеграла по x_0 в данной задаче значению шага дискретизации, вычисляемому на основе теоремы Котельникова, целесообразно сопоставить понятие "шаг интегрирования". При любом способе дискретизации на каждом шаге интегрирования целесообразно использовать квадратурную формулу, которая точна для алгебраических многочленов третьей или пятой степени. По мнению автора, при интегрировании по x_0 предпочтительной является известная квадратурная формула Гаусса на три узла.

Учитывая быстрое убывание модуля подынтегральной функции по мере удаления от точки $x_0 = x$ (см. рис. 1), при интегрировании в области $[x - d_0, x + d_0]$ в случае $x - d_0 \geqslant x_1$ и $x + d_0 \leqslant x_1 + l$ один из узлов квадратурной формулы должен совпадать с точкой $x_0 = x$, поскольку значение интег-

[4] Например, при исследовании влияния конструктивно-технологической реализации на уровень помех в целях выбора оптимальных значений параметров конструктива, в первую очередь — выбора материала и толщины диэлектрика.

[5] Эффективность других квадратурных формул не проверялась.

рируемой функции в этой точке дает превалирующий вклад в сумму квадратурной формулы и этот вклад должен быть учтен максимально точно.

Значение оптимального интервала интегрирования $2d_0$, полученное для симметричной относительно точки $x_0 = x$ области интегрирования по x_0, позволяет добиться указанных выше точности и выигрыша во времени счета также и при интегрировании на близких к концам проводника участках, где $|x - x_1| < d_0$ и $|x_1 + l - x| < d_0$. В этом случае область интегрирования оказывается несимметричной относительно точки $x_0 = x$, а интегрирование в зависимости от расположения точки x проводится в интервале оптимальной ширины $2d_0$ от начала проводника, т. е. в интервале $[x_1, x_1 + 2d_0]$, или до конца проводника, т. е. в интервале $[x_1 + l - 2d_0, x_1 + l]$. Парциальные погрешности при этом хорошо компенсируют друг друга и совмещать один из узлов квадратурной формулы с точкой $x_0 = x$ не следует.

При интегрировании потенциала магнитного поля источника численное интегрирование по x целесообразно заменить аналитическим интегрированием функции, полученной с помощью аппроксимации подынтегрального выражения. Как показал вычислительный эксперимент, подлежащий интегрированию потенциал магнитного поля на боковой кромке рецептора везде, кроме концов проводника-рецептора, с высокой точностью аппроксимируется функцией вида $A_x\exp[i(a + bx)]$, где A_x, a, b — действительные, не зависящие от переменной интегрирования x, коэффициенты аппроксимации, i — мнимая единица. Протяженность участков на концах рецептора, где зависимость аргумента комплексного числа от x становится заметно нелинейной, а модуль A_x заметно уменьшается, для печатных плат сравнительно невелика (приблизительно 2,5 мм с каждого конца на частоте $\omega = 10^{10}$ рад/с); от параметров конструктива она зависит слабо. Поэтому для проводников, длина которых $l \gg 5$ мм, указанным концевым эффектом можно пренебречь. Тогда интегрирование по x сводится к вычислению подынтегральной функции в двух точках вблизи середины проводника для определения трех коэффициентов аппроксимации A_x, a, b и вычислению значения табличного интеграла, имеющего хорошо известную первообразную.

Заключение

Описанные выше математическое обеспечение (методы, модели) и приемы организации вычислительного процесса позволяют существенно (в 300 раз, как минимум) повысить быстродействие по сравнению с данными, приведенными в работе [12] и сообщенными автором работы [13] по результатам реализации его метода. При реализации предлагаемого в данной работе математического обеспечения на языке Borland C++ (v. 3.1) и проведении основной части расчетов с учетом шести знаков мантиссы каждого операнда даже в случае использования компьютера широко доступного класса с тактовой частотой 800 МГц расход машинного времени на вычисление амплитуды помехи составляет менее 1 с на 100 пар проводников. Такое быстродействие при работе в интерактивном режиме вполне приемлемо, поскольку задача анализа электромагнитной совместимости цепей на плате или в микросхеме, как правило, решается на основе декомпозиции: обычно проводится последовательный анализ фрагментов конструктива и одномоментного расчета помех во всех цепях не требуется.

Таким образом, практическая реализация предлагаемого подхода приводит к математическим моделям, которые занимают промежуточное положение среди известных: они несколько уступают моделям, основанным на строго динамическом подходе и пространственной дискретизации, по широкополосности значительно превосходя их по экономичности (требуемому расходу машинного времени и требуемой емкости оперативной памяти). Вместе с тем, широко известные модели, использующие взаимные емкости и индуктивности, не имеют преимуществ перед предлагаемыми моделями ни по экономичности, ни по широкополосности.

Список литературы

1. **Конников И. А.** Метод эквивалентной постоянной распространения для моделирования электромагнитного поля в микроэлектронике // Сб. докл. "Научная сессия ГУАП. Технические науки". Ч. II. СПб.: ГУАП, 2008. С. 109—110.
2. **Конников И. А.** Математическая модель конструкции микросхемы // Математическое моделирование. 2007. № 4. С. 37—44.
3. **Конников И. А.** Оценка точности вычисления функции Грина в слоистой среде // Вычислительные технологии. 2006. № 5. С. 55—62.
4. **Конников И. А.** Влияние плотности распределения заряда на емкость прямоугольной пленки в слоистой среде // Электричество. 2007. № 3. С. 37—41.
5. **Максвелл Дж. К.** Трактат об электричестве и магнетизме. М.: Наука, 1989. Т. 2. 440 с.
6. **Калантаров П. Л., Цейтлин Л. А.** Расчет индуктивностей. Л.: Энергоатомиздат, 1986. 488 с.
7. **Конников И. А.** Емкость тонкого проводника прямоугольного сечения в микросхеме // Технология и конструирование в электронной аппаратуре. 2006. № 4. С. 18—23.
8. **Конников И. А.** Емкость тонкого проводника прямоугольного сечения // Авиакосмическое приборостроение. 2006. № 11. С. 19—25.
9. **Гоноровский И. С., Демин М. П.** Радиотехнические цепи и сигналы. Изд. 5-е. М.: Радио и связь, 1994. 480 с.
10. **Конников И. А.** Использование теоремы Котельникова для интегрирования функции Грина // Научная сессия ГУАП. Сб. докл. Ч. II. Технические науки. СПб.: ГУАП, 2012. С. 111—112.
11. **Клиначев Н. В.** Теория систем автоматического регулирования. URL: http://www.model.exponenta.ru/lectures/0130.htm.
12. **Конников И. А.** Модификация метода эквивалентной постоянной распространения для оценки перекрестных помех в электронном модуле // Научная сессия ГУАП. Сб. докл. Ч. II. Технические науки. СПб.: ГУАП, 2011. С. 126—127.
13. **Kochetov S. V.** PEEC-models based on dyadic Green's functions for structures in layered media // Proceedings 7-th International symposium on electromagnetic compatibility and electromagnetic ecology. June 26—29. SPb, 2007. P. 179—182.

УДК 519.17

И. А. Евсеенко, канд. техн. наук, доц.,
Белорусско-Российский университет, г. Могилев,
e-mail: 327igor@rambler.ru

Автоматизация формирования структуры трансформаторных элементов сложной конфигурации на основе теории графов

Предложен метод автоматизированного формирования структуры трансформаторных элементов сложной конфигурации на основе теории графов. Трансформаторные элементы разбиваются на подсистемы простых трансформаторов, представляемые в виде замкнутых подмножеств вершин с внутренними связями (внутренними ребрами). На вершины, принадлежащие различным подмножествам, накладываются внешние постоянные и переменные (случайные или управляемые) связи. Представлено применение предлагаемого метода применительно к планетарным коробкам передач для решения задач структурного синтеза и динамического анализа.

Ключевые слова: *граф, планетарная коробка передач, трансформаторный элемент, матрица инцидентности, матрица смежности, структурный синтез, автоматизация, матричное описание структуры планетарных коробок передач, теория графов, автоматизированное формирование структуры*

Введение

Трудоемкость автоматизации структурного синтеза и формирования динамических моделей трансформаторных элементов сложной конфигурации заключается в их многообразии. Для удобства работы с элементной базой необходимо, чтобы каждый трансформаторный элемент имел свое наглядное графическое представление. К настоящему времени запатентовано огромное число трансформаторных элементов (различного рода редукторов механической, гидравлической и электрической природы) и постоянно изобретаются новые.

Предусмотреть удобные для восприятия графические образы трансформаторных элементов для всех механизмов преобразования параметров потока мощности практически невозможно. Базовые элементы современных пакетов динамического моделирования предусматривают набор только наиболее распространенных трансформаторных элементов. Кроме того, увеличение разновидностей элементной базы создает определенные трудности для выбора нужных элементов и построения графического образа модели в целом.

Также имеется разное специализированное программное обеспечение для динамического моделирования применительно к конкретным объектам (гидромеханические трансмиссии, электрические силовые приводы и т. д.), в котором предусмотрен набор всех трансформаторных элементов, применяемых в рассматриваемом техническом объекте. Однако внедрение принципиально нового трансформаторного механизма создает сложности в таких программных продуктах и требует добавления новых элементов.

Актуальная задача заключается в создании универсальных схем замещения, либо универсального математического описания работы трансформаторного элемента с дальнейшей загрузкой пользователем графического изображения трансформатора (понятного и удобного для восприятия).

Универсальность математического описания базируется на аналогиях в динамических системах различной физической природы. Например, аналогия в поступательной и вращательной механических системах: модель зубчатой передачи и рычажного механизма описываются одинаковым уравнением. Передаточное число (параметр трансформаторного элемента) во вращательной системе определяется как отношение чисел зубьев ведущего и ведомого зубчатых колес, а в поступательной системе — отношением длин рычагов (качели). Такой же принцип положен в основу математического описания работы цепных и ременных передач.

Одним из возможных вариантов создания универсальных схем замещения является представление графической модели в виде графов.

Применение теории графов для представления структуры трансформаторных элементов сложной конфигурации недостаточно изучено, в частности применительно к планетарным коробкам передач.

Представление структуры трансформаторных элементов сложной конфигурации в виде графов

Одним из самых сложных трансформаторных элементов являются трансмиссии транспортных средств, где важную роль играют минимальные размеры и масса, и в то же время требуется обеспечить высокий КПД. Проблема особо остро стоит для трансмиссий с планетарными коробками передач (ПКП), которые требуют специфических подходов

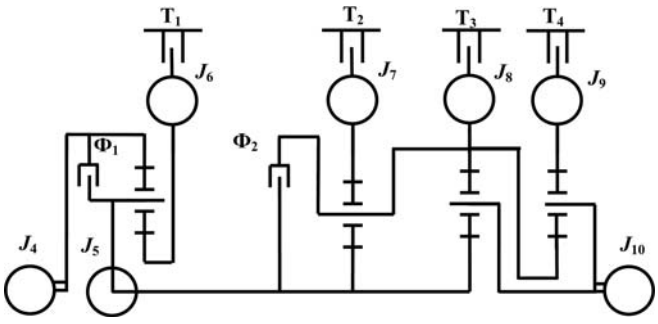


Рис. 1. Динамическая модель ПКП БелАЗ-7516

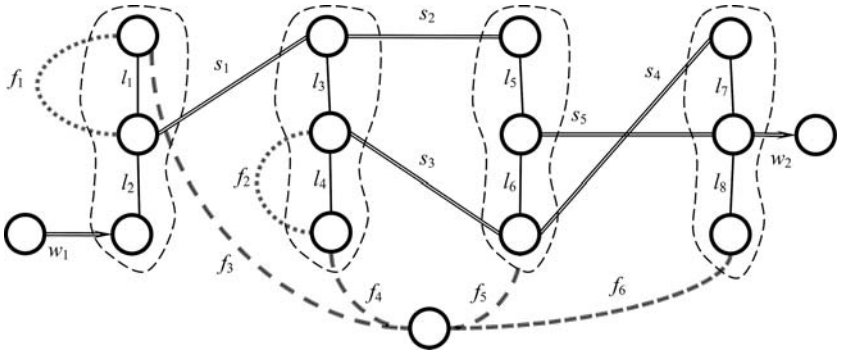


Рис. 2. ПКП БелАЗ-7516, представленная в виде графа

для проведения структурного синтеза и динамического анализа.

Рассмотрим пример представления структуры ПКП БелАЗ-7516 (рис. 1 и 2) на основе теории графов [1].

Предлагаемый метод представления структуры ПКП в виде графов заключается в следующем. Каждый *i*-й планетарный ряд, входящий в состав ПКП, представляется в виде подмножества, состоящего из трех вершин a_i, h_i, b_i (на рис. 2 подмножества выделены штриховыми одинарными линиями). Вершины соответствуют элементам планетарного ряда, т. е. центральным зубчатым колесам (*a*, *b*) и водилу (*h*).

Кроме вершин, соответствующих элементам планетарных рядов предусмотрены: базовая вершина "О", характеризующая неподвижную систему отсчета (корпус ПКП); вершина "Э", описывающая источник энергии; вершина "П", соответствующая потребителю или сопротивлению. Для соединения вершин между собой предложено использовать пять видов ребер: внутренние постоянные, внешние постоянные и внешние управляемые; входные; выходные.

Ребра, применяемые для постоянного соединения вершин в одном подмножестве, называют внутренними постоянными. Они характеризуют внутренние связи между элементами планетарного ряда, осуществляемые посредством зубчатых зацеплений или каким-либо иным способом. Вершины a_i и b_i внутри своего подмножества соединяются ребрами с вершиной h_i (на рис. 2 внутренние постоянные ребра показаны сплошными одинарными линиями l_1—l_8). Способ соединения вершин внутри подмножества зависит от специфики трансформаторного элемента (в частном случае планетарного ряда) и числа составных звеньев. Например, для зубчатой передачи с неподвижными осями (состоящей из двух зубчатых колес) подмножество будет состоять из двух вершин, соединенных между собой ребром.

Ребра, применяемые для постоянного соединения вершин, принадлежащих различным подмножествам (планетарным рядам) называют внешними постоянными. Они характеризуют способ и специфику соединения подмножеств (планетарных рядов) между собой, осуществляемых посредством шлицов или каким-либо иным способом, образуя каркас ПКП (на рис. 2 внешние постоянные ребра показаны сплошными двойными линиями s_1—s_5).

Ребра, применяемые для кратковременного соединения вершин, принадлежащих одному или разным подмножествам (планетарным рядам) в целях получения нужного передаточного числа, называют внешними управляемыми. Они характеризуют способ и специфику соединения элементов одного или разных подмножеств (планетарных рядов) между собой или с неподвижной системой отсчета для обеспечения требуемых параметров преобразования потока мощности, осуществляемых посредством элементов управления (фрикционные муфты и тормоза, муфты свободного хода и т. д.), лишая ПКП избыточного числа степеней свободы (на рис. 2 внешние управляемые ребра показаны прерывистыми тройными линиями f_1—f_6 (f_1, f_2 — фрикционные муфты, f_3—f_6 — фрикционные тормоза)).

Входные и выходные ребра служат для отображения входа и выхода ПКП (на рис. 2 входные и выходные ребра показаны двойными линиями со стрелками w_1, w_2).

Построение графа ПКП необходимо осуществлять в следующей последовательности: нанести базовую вершину, вершину источника энергии, вершину потребителя и подмножества (планетарные ряды) со входящими в них вершинами; построить внутренние постоянные ребра; построить внешние постоянные ребра; установить внешние управляемые ребра; задать входные и выходные ребра графа.

Матрицы смежности и инциденций, описывающие структуру графа, приведены в табл. 1 и 2.

Таблица 1

Матрица смежности ПКП БелАЗ-7516

	a_1	h_1	b_1	a_2	h_2	b_2	a_3	h_3	b_3	a_4	h_4	b_4	О	Э	П
a_1		$l_1; f_1$											f_3		
h_1	$l_1; f_1$		l_2	s_1											
b_1		l_2													w_1
a_2		s_1			l_3		s_2								
h_2				l_3		$l_4; f_2$			s_3						
b_2					$l_4; f_2$								f_4		
a_3				s_2				l_5							
h_3							l_5		l_6		s_5				
b_3					s_3			l_6		s_4			f_5		
a_4									s_4		l_7				
h_4								s_5		l_7		l_8			w_2
b_4											l_8		f_6		
О	f_3					f_4			f_5			f_6			
Э															
П			w_1								w_2				

Таблица 2

Матрица инциденций ПКП БелАЗ-7516

	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	s_1	s_2	s_3	s_4	s_5	f_1	f_2	f_3	f_4	f_5	f_6	w_1	w_2
a_1	1													1		1					
h_1	1	1							1					1							
b_1		1																		1	
a_2			1						1	1											
h_2			1	1							1				1						
b_2				1											1		1				
a_3					1					1											
h_3					1	1							1								
b_3						1					1	1						1			
a_4							1					1									
h_4							1	1					1								1
b_4								1											1		
О																1	1	1	1		
Э																				1	
П																					1

На основе изложенной методики представления структуры ПКП в виде графов можно осуществить структурный синтез ПКП путем генерирования всех возможных вариантов графов и, отбраковывая невозможные и неподходящие варианты.

Матричное представление структуры планетарных коробок передач

Рассмотрим матричное представление структуры трансформаторов сложной конфигурации на примере ПКП БелАЗ-7516 (см. рис. 1, табл. 3 и 4) для автоматизированного построения динамических моделей [2].

Таблица 3

Матрица жестких связей ПКП БелАЗ-7516

Инерционные элементы	a_1	h_1	b_1	a_2	h_2	b_2	a_3	h_3	b_3	a_4	h_4	b_4
J_4	0	0	−1	0	0	0	0	0	0	0	0	0
J_5	0	1	0	1	0	0	1	0	0	0	0	0
J_6	1	0	0	0	0	0	0	0	0	0	0	0
J_7	0	0	0	0	0	1	0	0	0	0	0	0
J_8	0	0	0	0	1	0	0	0	1	1	0	0
J_9	0	0	0	0	0	0	0	0	0	0	0	1
J_{10}	0	0	0	0	0	0	0	1	0	0	1	0

Таблица 4

Матрица фрикционных связей ПКП БелАЗ-7516

Инерционные элементы	Φ_1	Φ_2	T_1	T_2	T_3	T_4
J_4	−1	0	0	0	0	0
J_5	1	−1	0	0	0	0
J_6	0	0	−1	0	0	0
J_7	0	0	0	−1	0	0
J_8	0	1	0	0	−1	0
J_9	0	0	0	0	0	−1
J_{10}	0	0	0	0	0	0

Инциденторы в матрице жестких связей принимают значение "−1", если элемент планетарного ряда расположен на входном валу планетарного редуктора. Строка матрицы жестких связей представляет собой сосредоточенную массу, образованную путем жесткого соединения элементов планетарных рядов, инциденторы которых не равны нулю.

В матрице фрикционных связей учтено: "−1", если поток энергии отводится (ведущее звено фрикциона), и "1", если поток энергии подводится (ведомое звено фрикциона).

Таким образом, на основе матриц жестких и переменных связей (табл. 3 и 4) можно осуществить перебор всех возможных структур ПКП (с заданным числом планетарных рядов и включаемых элементов управления на каждой передаче), заполняя матрицы нулями и единицами по строго определенным правилам и отбраковывая невозможные варианты.

Заключение

Предложен метод представления структуры трансформаторных элементов сложной конфигурации на основе теории графов.

Новизна предлагаемого метода представления структуры трансформаторных элементов заключается в следующем:

- разбиение на подсистемы простых трансформаторов и представление их в виде замкнутых подмножеств вершин с внутренними связями (ребрами);
- на взаимодействие вершин, принадлежащих различным подмножествам, могут быть наложены внешние постоянные и переменные (случайные или управляемые) связи.

Преимуществом метода является универсальность, т. е. метод может быть применен для представления на ЭВМ структуры любого трансформаторного элемента. Предлагаемый метод хорошо приспособлен для автоматизации формирования структуры моделей с трансформаторными элементами при проведении статического и динамического анализа. Кроме того, изложенный подход может быть применен для решения задач структурного синтеза сложных трансформаторных элементов, когда необходимо осуществить оптимальный выбор структуры из огромного числа вариантов и проверить возможность реализации наиболее подходящих.

Список литературы

1. **Евсеенко И. А.** САПР планетарных коробок передач. Saarbrucken: LAP LAMBERT Academic Publishing GmbH & Co. KG, 2012. 200 с.

2. **Евсеенко И. А.** Методика автоматизированного построения динамических моделей планетарных коробок передач // Автомобильная промышленность. 2010. № 6. С. 36—39.

УДК 519.8

Ю. А. Зак, д-р техн. наук, науч. консультант,
Аахен, Германия
e-mail: yuriy_zack@hotmail.com

Методы локальных вариаций в решении задач теории расписаний

Сформулированы общие подходы, установлены свойства и параметры алгоритмов локальных вариаций, которые приемлемы для решения широкого класса задач теории расписаний в условиях наличия ограничений на сроки выполнения заданий. Разработанные методы иллюстрируются при построении алгоритмов решения различных классических задач разбиения на подмножества и упорядочения, имеющих прикладное значение в проблемах календарного планирования производства, маршрутизации перевозок и организации технического и сервисного обслуживания объектов.

Ключевые слова: *разбиение на подмножества и упорядочение заданий, локальные вариации, оптимальные расписания выполнения работ, ограничения на сроки выполнения заданий*

Введение

Методы локальных вариаций [2, 4, 5] сводятся, по сути дела, к случайному перебору на каждом k-м шаге векторов X из некоторой окрестности $\|X^k - X\| \leqslant \delta$, проверке их эффективности (пробного шага) $\{F(X) - F(X^k)\}$ и к выбору соответствующей реакции на исход эксперимента. Различные алгоритмы данного класса отличаются друг от друга реализацией отдельных этапов выбора экспериментальных точек и методами принятия решений. Наибольшее распространение в настоящее время получили монотонные алгоритмы локальной вариации полученного решения, в которых переход к следующему значению вектора переменных задачи происходит лишь в том случае, если $F(X^{k+1}) > F(X^k)$, в задаче максимизации целевой функции или в случае, если $F(X^{k+1}) < F(X^k)$, в задаче минимизации.

Задача теории расписаний в самом общем случае может быть сформулирована следующим образом. Множество заданий $\tilde{I} = \{i_1, i_2, ..., i_l, ..., i_N\}$ необходимо разбить на m непересекающихся подмножеств $\tilde{I}_k$, $k = 1, ..., m$, таким образом, чтобы $\bigcup_{k=1}^{m} \tilde{I}_k = \tilde{I}$, $\tilde{I}_k \cap \tilde{I}_p = \varnothing$, $k, p = 1, ..., m$, и определить последовательности выполнения в каждом из этих подмножеств $\overline{U}_k = \{u_{k1}, u_{k2}, ..., u_{kl}, ..., i_{kN_k}\}$. Для каждого из рассматриваемых заданий заданы соответственно t_i — время выполнения, θ_i и τ_i — наиболее ранний срок начала выполнения задания и граничные сроки его завершения. Могут быть также заданы ограничения на частичные порядки выполнения заданий. Построенные последовательности $\overline{U}_k$ определяют T_i — фактические сроки завершения выполнения каждого из заданий и $\overline{E}_k$ — время завершения выполнения каждой последовательности заданий $\overline{U}_k$. Решение задачи должно обеспечить выполнение всей заданной системы ограничений и оптимальное значение одного из критериев оптимальности

$$F_1 = \sum_{k=1}^{m} \alpha_k \overline{E}_k \to \min \text{ или } F_2 = \max_{1 \leqslant k \leqslant m} \alpha_k \overline{E}_k \to \min, \quad (1)$$

где $\alpha_k \geqslant 0$, нормированные значения весовых коэффициентов, $\sum_{k=1}^{m} \alpha_k = 1$. В рассматриваемых ниже задачах теории расписаний все граничные значения и времена выполнения операций предполагаются целочисленными величинами. Не допускается прерывание работ в процессе их выполнения.

Подробный обзор методов локальных вариаций для решения различных частных задач теории расписаний изложен в монографиях [2, 4, 5]. Применение этих алгоритмов для решения Flow-Shop-Problem [2, 4] рассмотрено в работе автора [6]. Цель данной работы — формирование некоторых общих подходов и установление общих свойств и параметров алгоритмов, которые приемлемы для решения широкого класса задач теории расписаний. Предложенные общие схемы решения детализируются при построении алгоритмов решения классических задач теории расписаний.

1. Общие подходы применения методов локальных вариаций для решения задач теории расписаний

При решении задач теории расписаний применение методов локальных вариаций сводится к выполнению следующих операций:

- перестановка местами двух операций (заданий) либо в последовательности выполнения их на одной и той же машине, либо в двух различных подмножествах, относящих их к различным видам ресурсов (машинам);
- выбор некоторого задания (или операции) и перестановка его на другое более раннее или более позднее место в последовательности выполнения заданий данного подмножества, либо включение его на какое-либо место в последовательности другого подмножества.

При перестановке местами двух операций или заданий одной и той же последовательности могут выбираться как рядом стоящие две операции, так и две операции, стоящие на некотором расстоянии друг от друга. При перестановке местами операций из разных подмножеств каждая заменяемая операция может устанавливаться на освободившееся место в новой для нее последовательности, либо после включения этой операции в новое для нее подмножество вновь производится упорядочение всех его членов. В случае наличия ограничений на времена начала и завершения выполнения заданий эти процедуры в ряде случаев позволяют сократить время завершения выполнения заданий, являющееся "узким местом", обеспечив выполнение ограничений и уменьшение значения критерия оптимальности задачи.

Эти методы стартуют с момента построения некоторой допустимой последовательности, которая строится на основе решающих правил или эвристик, учитывающих специфику решаемой задачи, и предусматривают выполнение следующих шагов и процедур:

- определение, в каком направлении проводить локальную вариацию (передвижение элемента в одну или другую сторону, перестановка двух элементов местами, детерминированный или случайный выбор элементов и т. п.);
- выбор элементов последовательности для перестановки их местами;
- выбор методов оценки допустимости и эффективности каждого шага;
- решение вопроса, осуществлять ли шаг локальной вариации, приведший к худшему результату;
- решение вопроса о целесообразности дальнейших продолжений.

Выбор перемещаемого элемента, определение направления перемещения (вверх или вниз), а также числа элементов в выбранном направлении, куда должен быть установлен элемент, может осуществляться детерминированным образом в соответствии с некоторым решающим правилом, или выбираться случайным образом.

Широкое распространение получили стратегии, предусматривающие выбор двух рядом стоящих элементов. Начиная с начала или конца последовательности, выбранный элемент перемещается в одном и том же направлении до тех пор, пока такое перемещение приводит к положительному результату. В случае если хотя бы одна из этих перестановок цикла привела к положительному результату, целесообразно повторить цикл перемещений, начиная с элемента, перестановка или перемещение которого была эффективной. Рассмотрим некоторую последовательность выполнения n заданий на одной машине — $\tilde{U} = \{u_1, u_2, ..., u_l, ..., u_p, ..., u_n\}$. Пусть t_i — время выполнения u_i-го задания. Если время начала выполнения всех работ расписания равно θ и на сроки начала выполнения каждого из заданий не наложено никаких ограничений, то время завершения выполнения задания u_i определяется выражением

$$T_{u_i} = \theta + \sum_{j=1}^{i} t_{u_j},\ i = 1, ..., n. \qquad (2)$$

Утверждение 1. При перестановке местами двух членов u_l и u_p последовательности выполнения заданий на одной машине $\tilde{U}_1 = \{u_1, u_2, ..., u_{l-1}, u_l, ..., u_{p-1}, u_p, ..., u_n\}$ и получения новой последовательности $\tilde{U}_2 = \{u_1, u_2, ..., u_{l-1}, u_p, ..., u_{p-1}, u_l, ..., u_n\}$ времена завершения выполнения всех заданий T_{u_i}, где $i = 1, ..., l-1, i = p+1, ..., n$, в последовательностях $\tilde{U}_1$ и $\tilde{U}_2$ равны, и подлежат изменению только времена выполнения заданий T_{u_i} подмножества индексов $i = l, l+1, ..., p$ рассматриваемой последовательности $\tilde{U}_1$, которые вычисляются по формулам

$$T_{u_i}(\tilde{U}_1) = T_{u_{l-1}} + t_{u_p} + \sum_{j=l+1}^{p-1} t_{u_j} + t_{u_l},$$
$$i = l, p+1, ..., n. \qquad (3)$$

Доказательство утверждения 1 следует из следующих соображений. Расчет значений $T_{u_i}(\tilde{U}_1)$ и $T_{u_i}(\tilde{U}_2)$, $i = 1, ..., l-1$, ведется по одним и тем же формулам. Для индексов $i = p+1, ..., n$ соответствующие значения вычисляются по формулам

$$T_{u_i}(\tilde{U}_1) = T_{u_{l-1}} + t_{u_l} + \sum_{j=l+1}^{p-1} t_{u_j} + t_{u_p} + \sum_{j=p+1}^{i} t_{u_j},$$
$$T_{u_i}(\tilde{U}_1) = T_{u_{t-1}} + t_{u_p} + \sum_{j=l+1}^{p-1} t_{u_j} + t_{u_l} + \sum_{j=p+1}^{i} t_{u_j}. \qquad (4)$$

Так как от перемены мест слагаемых значение суммы не меняется, то

$$T_{u_i}(\tilde{U}_1) = T_{u_i}(\tilde{U}_2),\ i = p+1, \ldots, n.$$

Следствия утверждения 1

1. При перестановке местами двух рядом стоящих членов u_{l-1} и u_l последовательности $\tilde{U}_1$ изменяются значения времени выполнения только двух рядом стоящих заданий $T_{u_{l-1}}(\tilde{U}_1)$ и $T_{u_l}(\tilde{U}_1)$.

2. Если значения $T_{u_l}(\tilde{U}_1)$ определяют условия выполнения ограничений на сроки выполнения заданий и значение критериев оптимальности построенного расписания, то проверка выполнения соответствующих ограничений и вычисление соответствующих показателей может проводиться только в тех членах, в которых присутствуют значения $T_{u_i}(\tilde{U}_1)$, $i = l, l+1, \ldots, p$.

Пусть заданы наиболее ранние сроки начала выполнения заданий $a_i \geqslant \theta$, $i = 1, \ldots, n$. Тогда в последовательности $\tilde{U}_1$ могут появиться и допускаются промежутки времени между завершением выполнения u_{l-1}-го $T_{u_{l-1}}(\tilde{U}_1)$ и началом выполнения u_l-го задания $\theta_{u_l}(\tilde{U}_1) = a_{u_l}$, в течение которых не проводится никаких работ. В этих случаях перенос отдельных заданий на более позднее место в $\tilde{U}_1$ может привести к повышению эффективности расписания.

Приведем несколько приложений использования методов локальной вариации при построении расписаний выполнения работ на одной и нескольких машинах.

2. Методы локальных вариаций при построении расписаний выполнения заданий на одной машине

2.1. *Минимизация суммы штрафов, связанных со сроками завершения выполнения заданий.* Пусть заданы время начала выполнения расписания, время выполнения каждого из заданий t_i, а также произвольные нелинейные функции штрафов, связанных со сроками завершения выполнения заданий $f_i(T_i)$. В качестве примеров таких функций могут встречаться как линейные функции, так и функции вида

$$f_i(T_i) = \varphi_i(T_i),\ f_i(T_i) = \begin{cases} 0, \text{ если } T_i \leqslant B_i, \\ \varphi_i(T_i) \text{ если } T_i > B_i, \end{cases}$$

$$f_i(T_i) = \begin{cases} 0, \text{ если } T_i \leqslant B_{1i}, \\ c_{1i}, \text{ если } B_{1i} < T_i \leqslant B_{2i}, \\ \ldots\ldots \\ c_{ri}, \text{ если } B_{r,\,i},\ i < T_i. \end{cases}$$

Здесь $\varphi_i(T_i)$ — произвольные нелинейные функции, которые могут быть недифференцируемыми и разрывными. Необходимо найти последовательность выполнения заданий, минимизирующую сумму штрафов.

В случае нелинейных функций штрафов не существует простого решающего правила (см. [2, 5] и др.), позволяющего построить оптимальную последовательность выполнения заданий за полиномиальное время. Рассматриваемая задача относится к классу *NP*-полных задач. Для решения этой задачи могут быть использованы описанные выше методы локальных вариаций. Начинаем с некоторой произвольной начальной последовательности выполнения заданий $\tilde{U}_0 = \{u_1, u_2, \ldots, u_{l-1}, u_l, \ldots, u_{p-1}u_p, \ldots, u_n\}$, для которой значение критерия оптимальности определяется выражением $F(\tilde{U}_0) = \sum_{i=1}^{n} f_{u_i}(T_{u_i})$. При этом значения T_{u_i} вычисляются по формулам (2). Поменяв местами два рядом стоящих члена последовательности u_{l-1} и u_l, получим новую последовательность $\tilde{U}_1$, в которой пересчету подлежат только два члена $f_{u_{l-1}}(T_{u_{l-1}})$ и $f_{u_l}(T_{u_l})$, где

$$\begin{aligned} T_{u_l}(\tilde{U}_1) &= \theta + \sum_{i=1}^{l-2} t_{u_i} + t_{u_l}, \\ T_{u_{l-1}}(\tilde{U}_1) &= T_{u_l}(\tilde{U}_0) = \theta + \sum_{i=1}^{l-2} t_{u_i} + t_{u_l} + t_{u_{l-1}}. \end{aligned} \tag{5}$$

При перемене мест в последовательности $\tilde{U}_0$ двух произвольных заданий u_l и u_p, где $p > l$, и получении новой последовательности $\tilde{U}_2 = \{u_1, u_2, \ldots, u_{l-1}, u_p, \ldots, u_{p-1}, u_l, \ldots, u_n\}$ пересчету подлежат значения времени завершения заданий и слагаемые суммы критерия эффективности членов $\tilde{U}_0$, индексы которых $i = l, l+1, \ldots, p$. Время завершения выполнения этих заданий определяется по формулам (3).

Рассмотрим один из возможных алгоритмов решения этой задачи. Начиная с первой пары членов последовательности и двигаясь вниз, проводится проверка возможности улучшить значение критерия эффективности путем перестановки двух рядом стоящих членов. Если для какой-либо пары такая перестановка окажется успешной, то на следующем шаге большой итерации, начиная со стоящего первым в последовательности перемещенного члена, проводится проверка эффективности перемещения его на более раннее место в последовательности.

2.2. *One-machine sequencing problem.* Рассмотрим известную в литературе задачу построения расписаний выполнения заданий на одной машине (One-machine sequencing problem [2—5]).

Заданы времена выполнения заданий t_i, ограничения на начальные сроки их выполнения a_i, а также заключительные времена g_i, необходимые для завершения выполнения каждого задания после окончания его обработки на данной машине. Необходимо найти оптимальную последовательность $\tilde{U} = \{u_1, u_2, ..., u_l, ..., u_p, ..., u_n\}$ обработки всех заданий на этой машине, а также время начала x_i и завершения σ_i выполнения каждого из заданий, обеспечивающее выполнение всех сформулированных выше ограничений при условии, что не допускается прерывание времени выполнения заданий. При этом необходимо минимизировать сроки завершения наиболее позднего из всех заданий с учетом добавления заключительного времени обработки g_i каждого из этих заданий на других машинах, т. е.

$$F = \min \max_{1 \leqslant i \leqslant n} (T_{u_i} + g_{u_i}). \qquad (6)$$

Здесь T_{u_i} — время завершения выполнения u_i-го задания, стоящего на i-м месте в построенной последовательности,

$$T_{u_i} = \max(a_{u_i}, T_{u_{i-1}}) + t_{u_i}, \; i = 1, ..., n. \qquad (7)$$

Рассматриваемая задача также относится к *NP*-сложным проблемам. Эффективные методы решения этой задачи впервые были предложены в работе [3], а с учетом ограничений на частичные порядки и граничные сроки завершения выполнения отдельных заданий — в работе [5].

Пусть построена некоторая последовательность выполнения заданий $\tilde{U}_1 = \{u_1, u_2, ..., u_l, ..., u_p, ..., u_n\}$, для которой по формулам (7) вычислены значения времени завершения выполнения всех заданий T_{u_i}, $i = 1, ..., n$. Пусть для этой последовательности $F(\tilde{U}_1) = (T_{u_p} + g_{u_p}) = \max\limits_{1 \leqslant i \leqslant n} (T_{u_i} + g_{u_i})$ и определяется p-м членом последовательности $\tilde{U}_1$.

Утверждение 2. Если $a_{u_p} = (T_{u_p} - t_{u_p})$ и $a_{u_p} = (T_{u_{p-1}} + 1)$, то построенная последовательность выполнения заданий является оптимальной, и никакие локальные вариации не могут привести к уменьшению значения критерия оптимальности.

Следствия утверждения 2

1. Если $a_{u_p} < (T_{u_p} - t_{u_p})$, то к уменьшению значения $F(\tilde{U}_1)$ может привести только перемещение задания с индексом u_p на какое-либо более раннее l-е место в последовательности $\tilde{U}_1 \Rightarrow \tilde{U}_2 = \{u_1, ..., u_{l-1}, u_p, u_{l+1}, ..., u_{p-1}, u_l, u_{p+1}, ..., u_n\}$ при выполнении условий

$$a_{u_p} \geqslant (T_{u_{l-1}} + \tau_{u_p} + 1),$$
$$(T_{u_l} + g_{u_l} | \tilde{U}_2) < (T_{u_p} + g_{u_p} | \tilde{U}_1), \qquad (8)$$

где $\tau_{u_p} \geqslant 0$ — необходимое минимальное число временных интервалов, обеспечивающее выполнение первого неравенства из условий (8).

2. Если в последовательности $\tilde{U}_2$ выполняется условие $a_{u_p} = (T_{u_{l-1}} + 1)$, т. е. $\tau_{u_p} = 0$, то значения

$$(T_{u_i} + g_{u_i} | \tilde{U}_2) = (T_{u_i} + g_{u_i} | \tilde{U}_1),$$
$$i = 1, ..., l-1; \; i = p+1, p+2, ..., n, \qquad (9)$$

и перестановка местами в $\tilde{U}_1$ членов u_l и u_p не может повлиять на изменение значения $F(\tilde{U})$. При вычислении $F(\tilde{U}_2)$ требуется лишь вычисление значений

$$(T_{u_i} + g_{u_i} | \tilde{U}_2),$$
$$i = (l-1), p, (l+1), (l+2), ..., (p-1), l. \qquad (10)$$

3. Если в последовательности $\tilde{U}_2$ значение $\tau_{u_p} > 0$, то при вычислении $F(\tilde{U}_2)$ требуется расчет всех значений

$$(T_{u_i} + g_{u_i} | \tilde{U}_2), \; i = (l-1), p, (l+1), (l+2), ...,$$
$$(p-1), l, (p+1), ..., n. \qquad (11)$$

4. Если условиями задачи предусмотрены граничные сроки завершения заданий и в последовательности $\tilde{U}_1$ все эти ограничения выполняются, то при переходе к последовательности $\tilde{U}_2$ и при условии $\tau_{u_p} = 0$ эти ограничения должны выполняться для всех членов, определяемых выражением (10), а при условии $\tau_{u_p} > 0$ — для всех членов, определяемых выражением (11).

5. В ряде случаев полученное в последовательности $\tilde{U}_1$ оптимальное решение не является единственным, и перестановкой местами двух членов этой последовательности u_l и u_r при условиях $r < p$ и $(T_{u_p} | \tilde{U}_1) = (T_{u_p} | \tilde{U}_2) = \text{const}$, либо u_t и u_k, где $p < t < k$, $(T_{u_j} | \tilde{U}_2) \leqslant (T_{u_p} | \tilde{U}_1)$, $j = (p+1), ..., n$, могут быть получены другие альтернативные оптимальные последовательности, обеспечивающие более эффективные значения других критериев оптимальности.

3. Синхронизация работы сборочных конвейеров

Пусть последовательность технологических операций сборки задана некоторым графом. Вершины графа $i = 1, ..., n$ определяют технологические опе-

рации, длительность которых равна t_i, где t_i — целые числа, а дуги — последовательность их выполнения. Рассматриваемая задача заключается в разбиении всего множества выполняемых операций $\tilde{I}$ на m ($m < n$) непересекающихся подмножеств $\tilde{I}_1, \tilde{I}_2, ..., \tilde{I}_m$ (здесь m — число рабочих станций) таким образом, что

$$\bigcup_{k=1}^{m} \tilde{I}_k = \tilde{I},\ \tilde{I}_k \cap \tilde{I}_p = \varnothing,\ k, p = 1, ..., m, \quad (12)$$

а также в определении последовательности выполнения операций на каждой рабочей станции, обеспечивающих допустимую последовательность выполнения операций сборки и экстремальное значение некоторого критерия оптимальности. В качестве критериев оптимальности могут быть выбраны минимальное число рабочих станций при условии выполнения ограничений, связанных с обеспечением заданной производительности конвейерной линии, либо минимальное время такта или цикла сборки (максимальной производительности) при заданном числе рабочих станций.

Время такта сборки θ, т. е. промежуток времени между завершением сборки r-го и $(r+1)$-го изделия, или обратная ему величина — производительность сборочного конвейера (число изделий, выпущенных линией за время T), определяется максимальной длительностью выполнения всех технологических операций на наиболее напряженном рабочем посту:

$$\theta = \max_{1 \leqslant k \leqslant m} \sum_{i \in \tilde{I}_k} t_i + \tau, \quad (13)$$

где τ — время транспортировки собираемого изделия с одного поста на другой; ($\tau_k = \tau_{k+1} = \tau$, $k = 1, ..., m-1$), $\tilde{I}_k$ — подмножество операций, выполняемых на k-м сборочном посту.

Введем следующие обозначения:

$\tilde{U}$ — множество всех дуг графа;

$A(i)$ — (непосредственно после "i") множество всех дуг $j \in \tilde{I}$, для которых

$$(i,j) \in \tilde{U},\ A(i) = \{j \in \tilde{I} \,|\, (i,j) \in \tilde{U}\};$$

$B(i)$ — (непосредственно перед "i") множество всех дуг $j \in \tilde{I}$, для которых

$$(j,i) \in \tilde{U},\ B(i) = \{j \in \tilde{I} \,|\, (j,i) \in \tilde{U}\}.$$

Каждое допустимое решение должно обеспечить выполнение следующих условий

$$\bigcup_{k=1}^{m} \tilde{I}_k = \tilde{I},\ \tilde{I}_k \cap \tilde{I}_p = \varnothing,\ k, p = 1, ..., m,\ k \neq p; \quad (14)$$

$$\bar{B}(i) \cap \left\{ \bigcup_{p=k+1}^{m} \tilde{I}_p \right\} = \varnothing,\ i \in \tilde{I}_k,\ k = 1, ..., m. \quad (15)$$

То есть если выполнение i-й операции назначено на рабочей станции k, то ни на одной из рабочих станций с индексом $p = k+1, ..., m$ не может выполняться ни одна операция из подмножества $\bar{B}(i)$, т. е. принадлежащая к одному из путей, ведущих из начальной фиктивной вершины $j = 0$ в данную вершину i. Если построено некоторое допустимое решение $\tilde{P}_1$, удовлетворяющее условиям технологической последовательности выполнения операций (14), (15), то процедура локальной вариации, обеспечивающая переход к другому допустимому решению $\tilde{P}_2$ с лучшим значением критерия оптимальности, может заключаться в следующем:

а) выбор некоторой операции $j \in \tilde{I}_{k-1}$, для которой $\bar{A}(i) \cap \{\tilde{I}_{k-1}/j\} = \varnothing$, и перенос ее в подмножество операций $i \in \tilde{I}_k$. Выполняем преобразования $\tilde{I}_k = \tilde{I}_{k-1}/j;\ \tilde{I}_k = \tilde{I}_k \cup j$;

б) выбор некоторой операции $j \in \tilde{I}_k$, для которой $\bar{B}(i) \cap \{\tilde{I}_k/j\} = \varnothing$, и перенос ее в подмножество операций $\tilde{I}_{k-1}$. Выполняем преобразования $\tilde{I}_{k-1} = \tilde{I}_{k-1} \cup j;\ \tilde{I}_k = \tilde{I}_k/j$;

в) перенос одной операции $j \in \tilde{I}_{k-1}$, удовлетворяющей условиям а), в подмножество операций $\tilde{I}_k$, и корректировка подмножеств $\tilde{I}_{k-1} = \tilde{I}_{k-1}/j$, $\tilde{I}_k = \tilde{I}_k \cup j$. Затем перенос некоторой операции $l \in \tilde{I}_k$, удовлетворяющей условиям "б)", в подмножество операций $\tilde{I}_{k-1}$, и корректировка подмножеств $\tilde{I}_{k-1} = \tilde{I}_{k-1} \cup l,\ \tilde{I}_k = \tilde{I}_k/l$.

После выполненного шага локальной вариации должны быть построены скорректированные последовательности выполнения операций на $(k-1)$-й и k-й рабочих станциях.

В результате описанных выше преобразований изменяются суммарные длительности выполнения операций на $(k-1)$-й и k-й рабочих станциях, что может привести к уменьшению времени такта сборочного конвейера.

Алгоритм локальных вариаций для решения задач определения оптимального распределения и установления последовательностей выполнения технологических операций на постах сборочного конвейера начинает работу после получения некоторого допустимого решения построенным любым из описанных в литературе приближенным методом (см., например, [2, 5]).

Пусть на некотором этапе решения некоторое допустимое решение $\tilde{P}_s$, $s = 0,1, ..., S$, со значением критерия оптимальности, равным

$$F(\tilde{P}_s) = \max_{1 \leqslant q \leqslant m} \sum_{i \in \tilde{I}_q} t_i + \theta = \sum_{i \in \tilde{I}_k} t_i + \theta, \quad (16)$$

где k — номер рабочей станции, суммарная длительность выполнения операций на которой максимальна.

Каждый этап локальной вариации включает выполнение следующих шагов.

Шаг 1. Находим номера рабочих станций k в соответствии с выражением (16). Если $\sum_{i \in \tilde{I}_k(\tilde{P}_s)} t_i = \sum_{i \in \tilde{I}_{k-1}(\tilde{P}_s)} t_i = \sum_{i \in \tilde{I}_{k+1}(\tilde{P}_s)} t_i$, то алгоритм завершает свою работу. В противном случае переходим к шагу 2.

Шаг 2. Если $\sum_{i \in \tilde{I}_k(\tilde{P}_s)} t_i > \sum_{i \in \tilde{I}_{k-1}(\tilde{P}_s)} t_i$, то определяем некоторые две операции $j \in \tilde{I}_{k-1}$ и $l \in \tilde{I}_k$, где $t_l > t_j$ (в частном случае операция $j \in \tilde{I}_{k-1}$ может не быть выбрана, т. е. $t_j = 0$), и выполняем одну из описанных выше процедур локальной вариации б) или в), соответствующим образом скорректировав подмножества операций $\tilde{I}_{k-1}$ и $\tilde{I}_k$. Если в результате выполненных преобразований получим

$$\sum_{i \in \tilde{I}_k(\tilde{P}_s)} t_i - t_l + t_j < F(\tilde{P}_s);$$

$$\sum_{i \in \tilde{I}_{k-1}(\tilde{P}_s)} t_i - t_j + t_l < F(\tilde{P}_s), \qquad (17)$$

то определяем новый план $\tilde{P}_{s+1}$, полагаем $\tilde{P}_s := \tilde{P}_{s+1}$. Переходим к шагу 1. Если в результате выполненных преобразований на шаге 2 не может быть найдена пара операций $j \in \tilde{I}_{k-1}$ и (или) $l \in \tilde{I}_k$, обеспечивающая выполнение условий (14), либо $\sum_{i \in \tilde{I}_k(\tilde{P}_s)} t_i = \sum_{i \in \tilde{I}_{k-1}(\tilde{P}_s)} t_i$, либо $k = 1$, то переходим к шагу 3.

Шаг 3. Если $\sum_{i \in \tilde{I}_k(\tilde{P}_s)} t_i > \sum_{i \in \tilde{I}_{k+1}(\tilde{P}_s)} t_i$, то определяем некоторые две операции $j \in \tilde{I}_{k+1}$ и (или) $l \in \tilde{I}_k$, где $t_l < t_j$, (в частном случае операция $j \in \tilde{I}_{k+1}$ может не выбираться, т. е. $t_j = 0$), и выполняем одну из описанных выше процедур локальной вариации а) или в), соответствующим образом скорректировав подмножества операций $\tilde{I}_{k-1}$ и $\tilde{I}_k$. Если в результате выполненных преобразований получим

$$\sum_{i \in \tilde{I}_k(\tilde{P}_s)} t_i - t_l + t_j < F(\tilde{P}_s);$$

$$\sum_{i \in \tilde{I}_{k+1}(\tilde{P}_s)} t_i - t_j + t_l < F(\tilde{P}_s), \qquad (18)$$

то определяем новый план $\tilde{P}_{s+1}$, полагаем $\tilde{P}_s := \tilde{P}_{s+1}$, и переходим к шагу 1.

Применение описанного выше алгоритма локальной вариации в ряде случаев позволяет сократить время такта работы сборочного конвейера при рассмотрении расписаний, построенных различными приближенными и эвристическими методами.

4. Построение расписаний выполнения заданий на параллельных машинах

На K, $k = 1, ..., K$, рабочих станциях (машинах) должны быть выполнены N различных работ (заданий), $\bar{I} = \{1, ..., i, ..., j, ..., N\}$ $i, j = 1, ..., N$. Каждое из заданий должно выполняться только на одной машине и без разрывов времени в процессе его выполнения. Заданы:

- директивные сроки завершения каждого из заданий T_i, $i = 1, ..., N$;
- матрица времен выполнения каждого из заданий на всех машинах $\bar{t}^k = (t_1^k, t_2^k, ..., t_i^k, ..., t_N^k)$, $k = 1, ..., K$;
- θ^k — наиболее ранние допустимые сроки начала выполнения работ на k-й машине;
- $A^k = |a_{ij}^k|$, $i, j = 0, 1, ..., N$, — матрицы времен потерь времени на переналадки при переходе k-й машины от выполнения одного задания к другому. На пересечении i-й строки и j-го столбца этих матриц стоят потери времени на переналадки k-й машины при переходе после выполнения i-го задания к j-му. В 0-й строке каждой матрицы A^k заданы времена настройки машины из состояния, в котором находится в момент начала выполнения расписания работ, в режим выполнения i-го задания. Элементы 0-го столбца определяют затраты времени на переход k-й машины после завершения выполнения j-го задания в режим простоя (или возвращения машины из j-го пункта на базу).

Необходимо найти распределение всего множества заданий по машинам

$$\bar{I}^k = \{i_1^k, ..., i_n^k\},\ k = 1, ..., K;\ \bigcup_{k=1}^{K} \bar{I}^k = \bar{I};$$

$$\bar{I}^k \cap \bar{I}^q = \varnothing,\ k, q = 1, ..., K,$$

а также определить последовательности выполнения всех назначенных на каждой машине заданий $\tilde{U}^k = \{u_1^k, ..., u_n^k\}$, $k = 1, ..., K$, обеспечивающие все установленные сроки их завершения T_i, и минимизировать время окончания всего комплекса работ (критерий оптимальности F_1). В качестве другого критерия оптимальности F_2 может быть выбрано минимальное средневзвешенное время работы машин, необходимое для выполнения всего комплекса работ. В работах автора [5, 7, 8] рассматривались свойства допустимых и оптимальных решений этих задач и

Задача	Размерность задач	Метод получения улучшаемого приближенного решения	Эффективность метода локальных вариаций, %
Минимизация суммы штрафов на 1-й машине	$n = 20...35$	Генетические алгоритмы	8...10
One-machine sequencing problem	$n = 200...250$	Полиномиальный алгоритм построения 1-го приближения в методе [3]	3...4
Синхронизация работы сборочного конвейера	$n = 150...200$ $m = 8...12$	Эвристический алгоритм [1]	7...8
Расписание выполнения работ на параллельных машинах	$n = 40...50$ $K = 5...10$	1. Эвристический алгоритм [1] 2. Генетические алгоритмы	6...9 4...5

предложены алгоритмы их решения методами динамического программирования [1] и ветвей и границ [2, 5]. Ниже рассматриваются алгоритмы решения этой задачи методами локальных вариаций.

В качестве начального приближения используется некоторое допустимое решение, полученное любым из приближенных или эвристических методов $\tilde{P}_0$, определяющее подмножества и последовательности $\bar{I}^k(\tilde{P}_0)$ и $\tilde{U}^k(\tilde{P}_0)$, $k = 1, ..., K$, в которых выполняются все ограничения на сроки выполнения заданий. Построим матрицы $B^k = |b_{ij}^k|$, $i, j = 0, 1, ..., N$, $k = 1, ..., K$, суммарных затрат времени на выполнение заданий на каждой машине, где

$$b_{ij}^k = \begin{cases} t_j^k + a_{ij}^k, \text{ если } j = 1, ..., N, \\ a_{ij}^k, \text{ если } j = 0;\ i = 0, 1, ..., N;\ k = 1, ..., K. \end{cases}$$

На s-й итерации алгоритма в качестве процедур локальной вариации при решении данной задачи могут быть использованы следующие:

а) перестановка местами двух каких-либо членов u_l^k и u_p^k в любой из последовательностей $\tilde{U}^k(\tilde{P}_s)$, $k = 1, ..., K$;

б) перенос некоторого задания u_l^k из подмножества $\tilde{I}_k(\tilde{P}_s)$ в некоторое другое подмножество $\bar{I}^q(\tilde{P}_s)$, при этом должны быть скорректированы последовательности выполнения заданий $\tilde{U}^k(\tilde{P}_s)$ и $\tilde{U}^q(\tilde{P}_s)$;

в) некоторое задание u_l^k из последовательности $\tilde{U}^k(\tilde{P}_s)$ ставится на место задания u_p^q на p-е место в последовательности $\tilde{U}^q(\tilde{P}_s)$; задание u_p^q, стоявшее на p-м месте в $\tilde{U}^q(\tilde{P}_s)$, переносится на l-е место в последовательности $\tilde{U}^k(\tilde{P}_s)$, т. е. заменяет в ней задание u_l^k;

г) некоторое задание i_l из подмножества $\tilde{I}_k(\tilde{P}_s)$ переносится в другое подмножество $\bar{I}^q(\tilde{P}_s)$, а задание i_p из $\bar{I}^q(\tilde{P}_s)$ включается в подмножество $\bar{I}^k(\tilde{P}_s)$. При этом должны быть скорректированы последовательности выполнения заданий $\tilde{U}^k(\tilde{P}_s)$ и $\tilde{U}^q(\tilde{P}_s)$ скорректированных подмножеств $\bar{I}^k(\tilde{P}_s)$ и $\bar{I}^q(\tilde{P}_s)$.

В процессе выполнения вычислительных экспериментов, результаты которых приведены в таблице, эффективность предлагаемых методов оценивалась показателем $\delta = \dfrac{F - F_{LV}}{F} 100\ \%$, где F и F_{LV} — соответственно значение критерия оптимальности решения, полученного приближенным методом, и улучшенное значение критерия в результате корректировки расписания методами локальных вариаций.

Заключение

1. Все рассмотренные в работе алгоритмы локальных вариаций имеют полиномиальную сложность, просты в программной реализации и требуют небольшого объема вычислений.

2. Применение методов локальных вариаций для расписаний, построенных приближенными или эвристическими методами, либо с помощью генетических алгоритмов, в ряде случаев позволяет осуществить спуск в точку локального или глобального минимума и тем самым повысить эффективность расписаний, построенных описанными выше методами.

3. Применение алгоритмов локальных вариаций для различных расписаний, построенных эвристическими или приближенными методами, позволяет получить несколько различных решений, очень близких по значению выбранного критерия оптимальности, которые отличаются друг от друга составом подмножеств и (или) последовательностями выполнения заданий в каждом из них. Среди полученных решений может быть осуществлен выбор наиболее приемлемого исходя из других показателей эффективности.

Список литературы

1. **Беллман Р.** Динамическое программирование. М.: Изд-во иностранной литературы, 1960.
2. **Domschke W., Scholl A., Vob S.** Produktionsplanung. Ablauforganisatorische Aspekte. Berlin, Heidelberg: Springer Verlag, 2005. 456 s.
3. **Carlier J.** The one-machine sequencing problem // European Journal of Operational Research. 1982. N 11. P. 42—47.
4. **Brucker P.** Scheduling Algorithms. Berlin, Heidelberg und New York: Springer-Verlag, 1998.
5. **Зак Ю. А.** Прикладные задачи теории расписаний и маршрутизации перевозок. М.: URSS, 2011. 394 с.
6. **Зак Ю. А.** Решение обобщенной задачи Джонсона с ограничениями на сроки выполнения заданий и времена работы машин. Ч. 2. Приближенные методы решения // Проблемы управления. 2010. № 4. С. 12—19.
7. **Зак Ю. А.** Разбиение на подмножества и построение допустимых и оптимальных последовательностей выполнения множества заданий на нескольких машинах // Системні дослідження та інформаційні технології. (Киев). 2012. № 2. С. 87—101.
8. **Зак Ю. А.** Распределение множества заданий и определение оптимальных очередностей их выполнения на параллельных машинах методами динамического программирования // Информационные технологии. 2012. № 8. С. 14—20.

УДК 621.396.6:621.391.827; 004.056:061.68

З. М. Гизатуллин, канд. техн. наук, доцент,
Р. М. Гизатуллин, аспирант,
Казанский национальный исследовательский технический университет им. А. Н. Туполева,
e-mail: gzm_zinnur@mail.ru

Моделирование электромагнитной обстановки на основе теории масштабного эксперимента для задач электромагнитной совместимости и защиты информации

Предложена методика и приведены результаты моделирования магнитных полей внутри здания при воздействии источника тока на элементы металлоконструкции здания на основе теории масштабного эксперимента.

Ключевые слова: *электромагнитная совместимость, защита информации, моделирование, масштабный эксперимент*

Введение

Анализ электромагнитной совместимости и защиты информации при внешних электромагнитных воздействиях неразрывно связан с точным определением электромагнитной обстановки вокруг электронных средств (ЭС). В формировании электромагнитной обстановки вокруг ЭС непосредственно участвуют макрообъекты (проводящие элементы конструкции зданий, проводящие элементы вокруг зданий, элементы конструкции транспортных средств и т. п.), которые имеют геометрические размеры намного больше, чем размеры самих ЭС. В данных условиях возникают определенные трудности с изготовлением макетов, имитаторов электромагнитного поля в реальном масштабе или с измерением быстротекущих электромагнитных процессов. Решением данной задачи является использование макетов или имитаторов, чьи размеры, материалы и временные характеристики приемлемы для экспериментатора. Задача сводится к определению критерия подобия при протекании электромагнитных процессов на макрообъектах с различными электромагнитными характеристиками, в предположении, что электромагнитные процессы на оригинале и модели описываются феноменологическими уравнениями Максвелла [1, 2].

1. Постановка задачи

Наиболее часто макрообъектом, участвующим в формировании электромагнитной обстановки вокруг ЭС, является здание (помещение здания), где ЭС устанавливается для последующей эксплуатации. В рамках данной работы рассматривается применение моделирования электромагнитной обстановки внутри здания на основе теории масштабного эксперимента при воздействии электромагнитного источника на элементы металлоконструкции здания. Элементы металлоконструкции здания — технические коммуникации (металлические трубопроводы горячей и холодной воды, отопления и т. д.), конструктивные элементы (арматура и др.), заземляющие устройства (контуры рабочего и защитного заземления, проводники молниеотводов) и т. д. Данная задача является частью общей проблемы обеспечения электромагнитной совместимости и защиты информации в ЭС при внешних непреднамеренных [3] и преднамеренных электромагнитных воздействиях [4].

2. Методика моделирования электромагнитной обстановки на основе теории масштабного эксперимента

В работе [2] приведены соотношения коэффициентов подобия для подбора параметров масштабной модели для анализа электромагнитной обстановки внутри макрообъектов:

$$f = a\sqrt{km};\ a = \frac{1}{p}\sqrt{\frac{k}{m}};\ c = \frac{1}{\sqrt{km}},$$

где a — коэффициент масштабирования физических размеров оригинала и модели; f — коэффициент масштабирования временной орты; k — коэффициент масштабирования электрической характеристики материала (ε — диэлектрическая проницаемость); m — коэффициент масштабирования магнитной характеристики материала (μ — магнитная проницаемость); p — коэффициент масштабирования электрической характеристики материала (σ — проводимость); c — коэффициент масштабирования скоростей. Если допустить $c = 1$, т. е. средняя скорость движения носителей в оригинале и в модели одинакова в подобных точках, то приходим к соотношениям:

$$f = a,\ k = \frac{1}{m},\ a \cdot p \cdot m = 1.$$

Если учитывать, что коэффициенты $k = m = 1$, так как в реальных условиях проведения масштабного эксперимента практически нельзя подобрать материалы с определенными, сильно отличающимися от оригинала диэлектрическими и магнитными свойствами, то получаем следующие выражения:

$$f = a,\ a = \frac{1}{p}.$$

Как видим, изменение коэффициента масштабирования временной орты f приводит к пропорциональному изменению коэффициента масштабирования физических размеров оригинала и модели a. При этом коэффициент масштабирования электрической проводимости материала p должен измениться обратно пропорционально данным коэффициентам. Трудности, связанные с масштабированием активного сопротивления, теоретически можно преодолеть, если заменить материалы оригинала другим материалом, с уменьшенной в p раз проводимостью. При такой замене активное сопротивление будет сохраняться неизменным, независимо от изменения физических размеров макрообъекта (что и требуется). Например, при уменьшении физических размеров макрообъекта с проводниками из латуни можно было бы их заменить на медь. Полученная таким образом масштабная модель полностью бы соответствовала оригиналу. Однако на практике, где правила масштабирования физических размеров являются весьма полезными, активное сопротивление играет не существенную роль. Например, как и в случае анализа электромагнитных полей при воздействии электромагнитного источника на элементы металлоконструкции здания (металлические трубопроводы, стальная арматура), активное сопротивление пренебрежимо мало. Следовательно, пропорциональное изменение физических размеров модели при сохранении ее характеристик по электрической и магнитной проницаемости приводит к пропорциональному изменению значений всех индуктивностей и емкостей, которые и являются определяющими при формировании электромагнитных процессов в оригинале и модели.

Таким образом, в рамках данной работы предлагается следующая методика моделирования электромагнитной обстановки вокруг ЭС внутри зданий на основе теории масштабного эксперимента.

1. Выбрать первичные и рассчитать вторичные масштабные коэффициенты для моделирования (табл. 1). Данные коэффициенты зависят от следующих факторов: условий проведения физического эксперимента (например, размеров лаборатории, параметров генераторов электромагнитного воздействия и т. п.); параметров здания — геометрических размеров, формы; параметров электромагнитного источника, воздействующего на элементы металлоконструкции здания (например генератора тока).

В качестве примера рассмотрим здание с размерами 10,8 × 10,8 × 14,4 м и стенами с армирующей сеткой (железобетонные стены). Размеры ячеек армирующей сетки стен здания — 0,25 × 0,25 м. Также здание имеет внешнюю систему молниезащиты с четырьмя токоотводами, расположенными в углах здания, что соответствует инструкции по устройству молниезащиты зданий [5].

2. Рассчитать значения реальных, расчетных масштабных и экспериментальных параметров источника тока (табл. 2).

3. Разработать масштабный физический макет здания, который должен учитывать следующие его

Таблица 1

Масштабные коэффициенты физических величин

Физическая величина	Масштабный коэффициент
Первичные	
Геометрические размеры	1:12
Время	1:12
Частота	12:1
Ток источника	1:6
Вторичные	
Производная от тока по времени	2:1
Напряженность магнитного поля	1:6
Производная от напряженности магнитного поля по времени	2:1

Таблица 2

Значение параметров источника тока

Параметры источника тока	Ток I (максимальный в импульсе), кА	Время фронта на уровне 10—90 % $t_ф$, мкс	Время спада на уровне 50 % $t_{50\,\%}$, мкс
Реальный [4, 6]	1,25	80	200
Масштабный	0,21	6,6	16,6
Экспериментальный	0,21	6,4	16

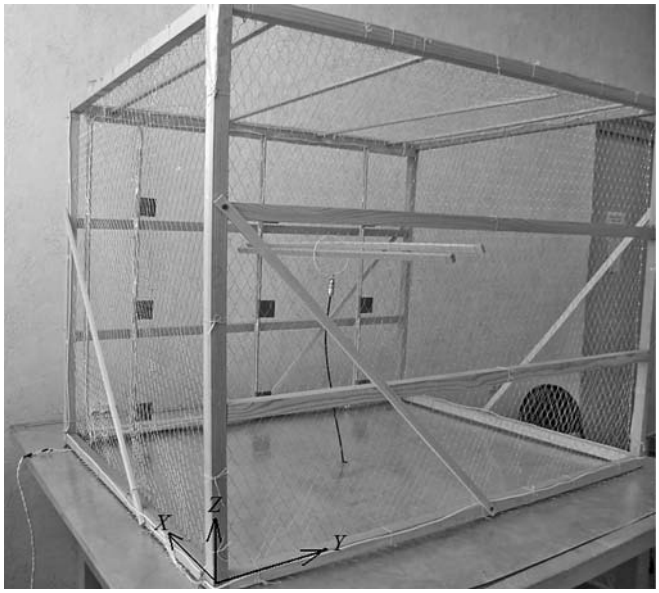


Рис. 1. Экспериментальный стенд для анализа электромагнитной обстановки внутри здания при воздействии источника тока на систему отопления здания

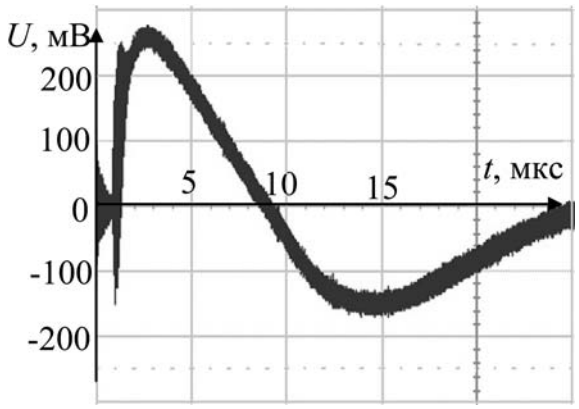


Рис. 2. Напряжение на рамочной антенне внутри масштабного макета здания (геометрическая ось антенны совпадает с осью *X*)

особенности: материалы элементов металлоконструкции здания; конфигурацию металлоконструкций здания; материал стен здания; точку и способ подключения источника электромагнитного воздействия к элементу металлоконструкции здания.

На рис. 1 представлен специальный экспериментальный стенд для анализа магнитных полей внутри здания при воздействии источника тока на систему отопления. Источник подключается параллельно между подающим и обратным трубопроводами системы водяного отопления [6, 7]. Измерение напряженности магнитного поля внутри здания осуществляется рамочной антенной (∅100 мм) по трем осям. Точки измерения (9 шт.) расположены на одной плоскости (расстояние от стен 0,2 м; между собой 0,25 м (по оси *X*); 0,4 м (по оси *Y*)), на высоте 0,4 м (по оси *Z*) от уровня земли и позволяют в целом оценить распределение напряженности магнитного поля внутри здания. Координаты данных точек условно обозначаются по осям (*X*, *Y*): (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3).

4. Провести серию экспериментальных исследований по измерению магнитных полей внутри масштабного макета здания (табл. 3).

На рис. 2 в качестве примера представлено измеренное внутри масштабного макета здания напряжение на рамочной антенне.

5. Провести моделирование электромагнитной обстановки (напряженности магнитного поля) внутри реального здания при воздействии генератора тока на систему отопления здания с использованием вторичного масштабного коэффициента, в данном случае равного 1:6 (табл. 4 и рис. 3, см. вторую сторону обложки).

Таким образом, по результатам моделирования электромагнитной обстановки внутри здания на основе теории масштабного эксперимента можно сделать следующие выводы: напряженность магнит-

Таблица 3

Значения напряженности магнитного поля внутри масштабного макета здания (измеренное)

Точки наблюдения	Напряженность магнитного поля, А/м			
	По оси *X*	По оси *Y*	По оси *Z*	Абсолютное значение
1	47,3	49,22	18,42	70,70
2	19,25	4,81	6,05	20,74
3	6,05	3,54	4,26	8,20
4	6,05	35,39	6,6	36,50
5	5,36	12,18	2,28	13,50
6	1,78	6,05	1,78	6,55
7	11,33	14,16	2,28	18,27
8	5,36	4,95	0,88	7,35
9	2,69	3,96	0,52	4,81

Таблица 4

Значения напряженности магнитного поля внутри реального здания (моделирование)

Точки наблюдения	Напряженность магнитного поля внутри здания, А/м			
	По оси *X*	По оси *Y*	По оси *Z*	Абсолютное значение
1	283,8	295,35	110,55	424,25
2	115,5	28,875	36,3	124,46
3	36,3	21,28	25,57	49,242
4	36,3	212,35	39,6	219,04
5	32,17	73,09	13,69	81,02
6	10,72	36,3	10,72	39,34
7	67,98	84,97	13,69	109,67
8	32,17	29,7	5,28	44,10
9	16,17	23,76	3,13	28,91

ного поля внутри здания при воздействии источника тока на систему отопления здания (при рассмотренных исходных данных) составляет 28,9...424,3 А/м; вектор напряженности магнитного поля имеет произвольное направление; наибольший уровень напряженности магнитного поля наблюдается поблизости от точки подключения источника.

Заключение

Предложена методика для моделирования электромагнитной обстановки внутри здания при воздействии электромагнитного источника на металлоконструкцию здания на основе теории масштабного эксперимента.

В качестве примера рассмотрена задача моделирования магнитных полей внутри здания с железобетонными стенами при воздействии генератора тока на систему отопления здания. Данная задача является частью проблемы защиты информации в ЭС при преднамеренных силовых электромагнитных воздействиях по металлоконструкциям [4, 6].

Приведенный пример измерения напряженности магнитного поля внутри масштабного макета здания и моделирование напряженности магнитного поля внутри реального здания через масштабные коэффициенты показывает эффективность данного подхода для решения задач электромагнитной совместимости и защиты информации, в условиях трудностей с изготовлением имитаторов и макетов в реальном масштабе.

Работа выполнена по ФЦП "Научные и научно-педагогические кадры инновационной России" на 2009—2013 годы

Список литературы

1. **Говард В. Д.** Высокоскоростная передача цифровых данных: высший курс черной магии. М.: Вильямс, 2005. 1024 с.

2. **Невзоров В. Н., Шувалов Л. Н.** Дополнительные возможности для моделирования электромагнитных процессов на макрообъектах. Деп. в ВИНИТИ. 05.11.90. № 5649-В90. 5 с.

3. **Гизатуллин З. М.** Анализ магнитных полей при воздействии разряда молнии на внешнюю систему молниезащиты здания // Технологии электромагнитной совместимости. 2010. № 3. С. 30—36.

4. **ГОСТ Р 52863—2007.** Защита информации. Автоматизированные системы в защищенном исполнении. Испытания на устойчивость к преднамеренным силовым электромагнитным воздействиям. Общие требования. М.: Изд-во стандартов, 2008. 33 с.

5. **Инструкция** по устройству молниезащиты зданий, сооружений и промышленных коммуникаций. СО 153-34.21.122—2003. М., 2003. 29 с.

6. **Сухоруков С. А.** Комментарии к ГОСТ 52863—2007 // Технологии электромагнитной совместимости. 2012. № 3. С. 13—25.

7. **СНиП. 41-01—2003.** Отопление, вентиляция и кондиционирование / Госстрой России. — М.: ФГУП ЦПП, 2004. 35 с.

УДК 519.23/25

Е. А. Малеев, аспирант,
e-mail: scoch_67@bk.ru,
В. А. Чепурко, канд. физ.-мат. наук, доц.,
e-mail: chepurko@iate.obninsk.ru,
Обнинский институт атомной энергетики
(ИАТЭ НИЯУ МИФИ), г. Обнинск

Корневая оценка плотности распределения по неполным данным

Предложены две модификации непараметрической корневой оценки плотности распределения в ситуации наличия неполных данных в виде группированных частот отказов. Первый (интегральный) метод связан с соответствующим изменением функции правдоподобия. Второй (resampling) метод восстановления отказов основан на итерационном восстановлении моментов отказов. Исследованы точности предложенных методов оценивания.

Ключевые слова: *пси-функция, метод квадратного корня, функция правдоподобия, псевдоотказы*

Введение

Объем статистической информации, поступающей на обработку, как правило, ограничен. Приходится сталкиваться с информацией, в которой наряду с наработками отказавших объектов присутствуют наработки объектов, продолжающих работать, но наблюдения за функционированием которых были по различным причинам приостановлены. Кроме этого, часто приходится иметь дело с группированной информацией об отказах, в которой потеряна информация о наработках отказавших объектов, а известна лишь частота их появления. Информацию подобной неопределенности называют цензурированной.

Как известно, методы анализа статистической информации делятся на параметрические и непараметрические. Для анализа данных об отказах малого объема рациональнее использовать непараметрические методы, не требующие, чтобы распределение вероятностей было описано каким-либо параметрическим законом распределения [1].

Наиболее общей характеристикой, описывающей поведение одномерной случайной величины, является ее плотность распределения $f(t)$. Задача оценки

плотности распределения наблюдаемой случайной величины по конечному числу ее реализаций при наличии неопределенностей является одной из ключевых задач статистического анализа, что и определяет актуальность настоящей статьи.

Известно множество методов оценивания плотности распределения полных и цензурированных данных: гистограммные, проекционные, ядерные, корневые оценки. Все эти методы имеют как достоинства, так и недостатки.

Метод гистограмм прост в реализации, однако не слишком нагляден, и гистограмма, построенная по малым выборкам, не позволяет сделать правильных выводов. Недостатком проекционной оценки является то, что на краях рассматриваемого интервала она может принимать отрицательные значения, тогда как плотность по определению неотрицательна. Качество ядерной оценки сильно зависит от выбора "ядра".

Корневая оценка представляет собой квадрат разлагаемой по ортонормированному базису функции и заведомо задает плотность. Оценка хорошо изучена для полных данных. В статье рассматривается корневая оценка для данных, имеющих неопределенность в моменте реализации исследуемого признака, т. е. для цензурированных данных.

В данной работе корневой метод оценки плотности условно разделен на два метода — интегральный и итеративный.

Корневая оценка плотности

Как известно, в классическом методе корневой оценки искомая плотность распределения $f_\xi(x)$ случайной величины ξ находится как квадрат так называемой пси-функции:

$$\hat{f}_\xi(x) = |\hat{\psi}(x)|^2. \qquad (1)$$

Разложим пси-функцию в ряд:

$$\hat{\psi}(x) = \sum_{i=1}^{m} c_i \varphi_i(x),$$

где $\{\varphi_i(x)\}$ — ортонормированная система; $\{c_i\}$ — коэффициенты разложения, подлежащие оценке (см. [2, 3]). После подстановки разложения в (1) получим корневую оценку плотности $\hat{f}_\xi(x)$.

В дальнейшем предполагается, что функции $\varphi_i(x)$, $\psi(x)$ и коэффициенты c_i действительны. Из условия нормировки, $\int \hat{f}_\xi(x)dx = 1$ следует равенство

$$\sum_{i,j=1}^{m} c_i c_j \int \varphi_i(x)\varphi_j(x)dx = \sum_{i=1}^{m} c_i^2 = 1. \qquad (2)$$

Следовательно, необходимо оценить $m - 1$ независимых коэффициентов. Для их оценки используется метод максимального правдоподобия.

Если выборка повторная — $\boldsymbol{\xi} = (\xi_1, ..., \xi_p)$, то функция правдоподобия (ФП) имеет следующий вид:

$$L_n(\mathbf{c}) = \prod_{k=1}^{p} \hat{f}_\xi(\xi_k) = \prod_{k=1}^{p} \left(\sum_{i=1}^{m} c_i \varphi_i(\xi_k) \right)^2.$$

Логарифмическая функция правдоподобия (ЛФП):

$$l_n(\mathbf{c}) = \ln L_n(\mathbf{c}) = \sum_{k=1}^{p} \ln \widehat{f}_\xi(\xi_k) =$$

$$= \sum_{k=1}^{p} \ln \sum_{i=1}^{m} \sum_{j=1}^{m} c_i c_j \varphi_i(\xi_k)\varphi_j(\xi_k).$$

Ее частные производные

$$\frac{\partial l_n(\mathbf{c})}{\partial c_i} = \sum_{k=1}^{p} \frac{\partial}{\partial c_i} \ln \widehat{f}_\xi(\xi_k) =$$

$$= \sum_{k=1}^{p} \frac{1}{\widehat{f}_\xi(\xi_k)} \frac{\partial}{\partial c_i} \left(\sum_{j=1}^{p} c_j \varphi_j(\xi_k) \right)^2 =$$

$$= \sum_{k=1}^{p} \frac{2\varphi_i(\xi_k) \left(\sum_{j=1}^{m} c_j \varphi_j(\xi_k) \right)}{\widehat{f}_\xi(\xi_k)} = 2 \sum_{k=1}^{p} \sum_{j=1}^{m} \frac{\varphi_j(\xi_k)\varphi_i(\xi_k)}{\widehat{f}_\xi(\xi_k)} c_j.$$

Возникает задача на условный экстремум:

$$L_n(\mathbf{c}) = \prod_{i=k}^{p} \hat{f}_\xi(\xi_k) = \prod_{i=k}^{p} \left(\sum_{i=1}^{m} c_i \varphi_i(\xi_k) \right)^2 \to \max_{\mathbf{c}}$$

с ограничением типа равенства $\sum_{i=1}^{m} c_i^2 = 1$. Коэффициенты c_i подбирают таким образом, чтобы ФП была максимальна, при этом их сумма квадратов равна 1.

Для нахождения максимального значения ЛФП $l_n(\mathbf{c})$ с учетом ограничения (2) формируется функция Лагранжа $L(\mathbf{c}) = l_n(\mathbf{c}) + \lambda \left(1 - \sum_{i=1}^{m} c_i^2 \right)$.

Производная функции Лагранжа приравнивается к нулю:

$$\frac{\partial}{\partial c_i} L(\mathbf{c}) = 2 \sum_{k=1}^{p} \sum_{j=1}^{m} \frac{\varphi_j(\xi_k)\varphi_i(\xi_k)}{\widehat{f}_\xi(\xi_k)} c_j - 2\lambda c_i = 0. \qquad (3)$$

Умножая обе части (3) на c_i и суммируя по i, получим $\lambda = p$. Подставив это в выражение (3), получим следующую систему нелинейных уравнений:

$$c_i = \frac{1}{p} \sum_{k=1}^{p} \varphi_i(\xi_k) \frac{\sum_{j=1}^{m} c_j \varphi_j(\xi_k)}{\widehat{f}_\xi(\xi_k)} =$$

$$= \frac{1}{p} \sum_{k=1}^{p} \varphi_i(\xi_k) \left(\sum_{j=1}^{m} c_j \varphi_j(\xi_k) \right)^{-1}, \; i = 1, ..., m.$$

Далее для ее решения, нахождения коэффициентов c_i, используются различные итерационные численные методы.

Корневая интегральная оценка плотности

Теперь решим задачу корневого оценивания плотности при наличии неполных (цензурированных) данных. При наличии такого рода неопределенности, в качестве исходной информации для построения оценки выступают элементы массива интервалов $\mathbf{LR} = [(l_1, r_1); (l_2, r_2); ...; (l_s, r_s)]$ и значений скачков эмпирической функции распределения в начале каждого интервала, которые, как известно, пропорциональны числу элементов выборки (случайному числу отказов), попавших на данный интервал $\boldsymbol{\nu} = (\nu_1, \nu_2, ..., \nu_s)$.

ФП для такого рода данных примет вид:

$$L_n(\mathbf{c}) = \prod_{k=1}^{p} \hat{f}_\xi(\xi_k) \prod_{m=1}^{s} (\hat{F}_\xi(r_m) - \hat{F}_\xi(l_m))^{\nu_m}, \quad (4)$$

т. е. к ФП полных данных добавится сомножитель $\prod_{m=1}^{s} (\hat{F}_\xi(r_m) - \hat{F}_\xi(l_m))^{\nu_m}$, отвечающий за цензурированные данные.

Далее будем действовать согласно классической схеме. Частные производные ЛФП будут равны следующим суммам:

$$\frac{\partial l_n(\mathbf{c})}{\partial c_i} = 2 \sum_{k=1}^{p} \sum_{j=1}^{m} \frac{\varphi_j(\xi_k)\varphi_i(\xi_k)}{\widehat{f}_\xi(\xi_k)} c_j + \sum_{m=1}^{s} \nu_m \frac{\partial \ln(\widehat{F}_\xi(r_m) - \widehat{F}_\xi(l_m))}{\partial c_i}.$$

Рассмотрим отдельно слагаемое второй суммы

$$\frac{\partial \ln(\hat{F}_\xi(r_m) - \hat{F}_\xi(l_m))}{\partial c_i}. \quad (5)$$

Если оценка плотности распределения

$$\hat{f}_\xi(x) = \left(\sum_{i=1}^{m} c_i \varphi_i(x) \right)^2, \quad (6)$$

то целесообразно взять в качестве оценки функции распределения интеграл

$$\hat{F}_\xi(x) = \int_{-\infty}^{x} \hat{f}_\xi(u)du. \quad (7)$$

В дальнейших выкладках предположим, что плотность распределения имеет носитель — отрезок [0, 1]. Для плотностей с другим носителем распределения можно сделать необходимое линейное преобразование случайной величины, отображающее множество ее значений в отрезок [0, 1] или использовать иные ортонормированные базисы.

Как известно, $\varphi_k(x) = \sqrt{2}\sin(k\pi x)$, $k = 1, 2, ...$ — ортонормированный базис Фурье на отрезке [0, 1]. Найдем интеграл (7), используя разложение (6):

$$\hat{F}_\xi(x) = \sum_{i=1}^{m} \sum_{j=1}^{m} c_i c_j \int_0^x \varphi_i(u)\varphi_j(u)du = \sum_{i=1}^{m} \sum_{j=1}^{m} c_i c_j 2 \int_0^x \sin(i\pi u)\sin(j\pi u)du =$$

$$= x - \frac{1}{2\sqrt{2}\pi} \left[\sum_{i=1}^{m} c_i^2 \frac{\varphi_{2i}(x)}{i} + \sum_{i=1}^{m} \sum_{j=1, j\neq i}^{m} c_i c_j \left(\frac{\varphi_{i+j}(x)}{i+j} - \frac{\varphi_{i-j}(x)}{i-j} \right) \right].$$

Частные производные функции распределения равны

$$\frac{\partial \hat{F}_\xi(x)}{\partial c_i} = \frac{1}{\sqrt{2}\pi} \left[2 \sum_{j=1, j\neq i}^{m} c_j \left(\frac{\varphi_{i-j}(x)}{i-j} - \frac{\varphi_{i+j}(x)}{i+j} \right) - \frac{c_i \varphi_{2i}(x)}{i} \right] = \frac{\sqrt{2}}{\pi} \left[\sum_{j=1, j\neq i}^{m} \frac{c_j \varphi_{i-j}(x)}{i-j} - \sum_{j=1}^{m} \frac{c_j \varphi_{i+j}(x)}{i+j} \right].$$

После подстановки полученных результатов (частных производных) в выражение (5) получим следующие уравнения:

$$\frac{\partial \ln(\hat{F}_\xi(r_m) - \hat{F}_\xi(l_m))}{\partial c_i} = \frac{\sqrt{2}}{\pi(\hat{F}_\xi(r_m) - \hat{F}_\xi(l_m))} \left[\sum_{j=1, j\neq i}^{m} \frac{c_j(\varphi_{i-j}(r_m) - \varphi_{i-j}(l_m))}{i-j} - \sum_{j=1}^{m} \frac{c_j(\varphi_{i+j}(r_m) - \varphi_{i+j}(l_m))}{i+j} \right].$$

Тогда

$$\frac{\partial l_n(\mathbf{c})}{\partial c_i} = 2 \sum_{k=1}^{p} \sum_{j=1}^{m} \frac{\varphi_j(\xi_k)\varphi_i(\xi_k)}{\hat{f}_\xi(\xi_k)} c_j + \sum_{m=1}^{s} \frac{\sqrt{2}\nu_m}{\pi(\hat{F}_\xi(r_m) - \hat{F}_\xi(l_m))} \times \left[\sum_{j=1, j\neq i}^{m} \frac{c_j(\varphi_{i-j}(r_m) - \varphi_{i-j}(l_m))}{i-j} - \sum_{j=1}^{m} \frac{c_j(\varphi_{i+j}(r_m) - \varphi_{i+j}(l_m))}{i+j} \right].$$

Необходимое условие экстремума, как и для полных данных, сводится к условиям равенства нулю частных производных функции Лагранжа:

$$\frac{\partial}{\partial c_i} L(\mathbf{c}) = 2 \sum_{k=1}^{p} \sum_{j=1}^{m} \frac{\varphi_j(\xi_k)\varphi_i(\xi_k)}{\hat{f}_\xi(\xi_k)} c_j - 2\lambda c_i + \\ + \sum_{m=1}^{s} \frac{\sqrt{2}\nu_m}{\pi(\hat{F}_\xi(l_m) - \hat{F}_\xi(r_m))} \times \\ \times \left[\sum_{j=1, j\neq i}^{m} \frac{c_j(\varphi_{i-j}(r_m) - \varphi_{i-j}(l_m))}{i-j} - \right. \\ \left. - \sum_{j=1}^{m} \frac{c_j(\varphi_{i+j}(r_m) - \varphi_{i+j}(l_m))}{i+j} \right] = 0. \qquad (8)$$

Умножим обе части (8) на c_i, просуммируем по i, получим следующее уравнение для λ:

$$\lambda = p + \sum_{m=1}^{s} \frac{\nu_m}{\pi\sqrt{2}(\hat{F}_\xi(r_m) - \hat{F}_\xi(l_m))} \sum_{i=1}^{m} c_i \times \\ \times \left[\sum_{j=1, j\neq i}^{m} \frac{c_j(\varphi_{i-j}(r_m) - \varphi_{i-j}(l_m))}{i-j} - \right. \\ \left. - \sum_{j=1}^{m} \frac{c_j(\varphi_{i+j}(r_m) - \varphi_{i+j}(l_m))}{i+j} \right].$$

Для нахождения решения системы уравнений (7) можно предложить следующий итерационный алгоритм:

$$c_i^{(q+1)} = \alpha c_i^{(q)} + \frac{1-\alpha}{2\lambda} \left\{ 2 \sum_{k=1}^{p} \sum_{j=1}^{m} \frac{\varphi_j(\xi_k)\varphi_i(\xi_k)}{\hat{f}_\xi^{(q)}(\xi_k)} c_j^{(l)} + \right. \\ + \sum_{m=1}^{s} \frac{\sqrt{2}\nu_m}{\pi(\hat{F}_\xi^{(q)}(r_m) - \hat{F}_\xi^{(q)}(l_m))} \times \\ \times \left[\sum_{j=1, j\neq i}^{m} \frac{c_j^{(q)}(\varphi_{i-j}(r_m) - \varphi_{i-j}(l_m))}{i-j} - \right. \\ \left. \left. - \sum_{j=1}^{m} \frac{c_j^{(q)}(\varphi_{i+j}(r_m) - \varphi_{i+j}(l_m))}{i+j} \right] \right\}.$$

Теперь приведем формулы для корневой оценки на произвольном отрезке локализации. В произвольном случае (случайная величина распределена на отрезке $[a, b]$) базис будет иметь вид $\varphi_k(x) = \sqrt{\frac{2}{b-a}} \sin\left(k\pi \frac{x-a}{b-a}\right)$.

В качестве $[a, b]$ могут быть взяты первая и последняя порядковые статистики, т. е. наименьшее и наибольшее значение.

Итерационный процесс для нахождения коэффициентов разложения в этом случае приведет к следующей схеме:

$$c_i^{(q+1)} = \alpha c_i^{(q)} + \frac{1-\alpha}{2\lambda} \left\{ 2 \sum_{k=1}^{p} \sum_{j=1}^{m} \frac{\varphi_j(\xi_k)\varphi_i(\xi_k)}{\hat{f}_\xi^{(q)}(\xi_k)} c_j^{l} + \right. \\ + \sum_{m=1}^{s} \frac{\sqrt{2(b-a)}\nu_m}{\pi(\hat{F}_\xi^{(q)}(r_m) - \hat{F}_\xi^{(q)}(l_m))} \times \\ \times \left[\sum_{j=1, j\neq i}^{m} \frac{c_j^{l}(\varphi_{i-j}(r_m) - \varphi_{i-j}(l_m))}{i-j} - \right. \\ \left. \left. - \sum_{j=1}^{m} \frac{c_j^{l}(\varphi_{i+j}(r_m) - \varphi_{i+j}(l_m))}{i+j} \right] \right\}. \qquad (9)$$

Метод корневого оценивания, основанный на ФП (4) и представлении (7), а также использующий итерационный процесс (9), назовем *интегральным*.

Корневая итеративная оценка плотности

Следующая модификация метода корневого оценивания по цензурированным данным основана на методе восстановления отказов (получении псевдоотказов) будет нами условно определена как *итеративная*. В западной литературе эта процедура носит название *Resampling-method*. Принцип моделирования псевдонаблюдений опирается на следующее известное свойство монотонной функции распределения: случайная величина $F_\xi(\xi)$ является равномерно распределенной на отрезке [0; 1]. Для каждого источника с цензурированными данными проводится поиск значения функции распределения в точках границ интервалов цензурирования, затем в каждом из построенных интервалов функций моделируется некоторое число точек, равное числу отка-

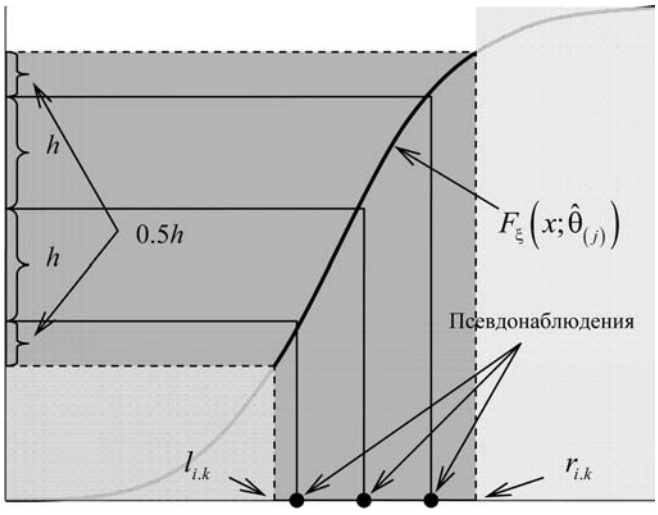


Рис. 1. Моделирование псевдонаблюдений на участке неопределенности

зов на данном интервале. С помощью интерполяции выполняется обратное отображение смоделированных псевдоотказов на ось наработок (рис. 1) [4].

На следующем шаге восстановленные отказы от каждого источника с группированной информацией собираются в один массив, далее с ними работаем, как с полными данными, по ним строим корневую (интегральную) оценку по формуле (1). Новая функция распределения находится как интеграл от этой оценки, по новой функции опять восстанавливаются отказы, так до тех пор, пока итерационный процесс не сойдется.

Отказы восстанавливались по функции $F_{ср}$, представляющей собой среднее значение эмпирических функций распределения источников с цензурированными данными и функции $F_{пр}$, взятой как интеграл от корневой оценки плотности полных данных.

Исследования оценок

Оценки плотности распределения исследовали на примере закона распределения Вейбулла с параметром формы $\alpha = 2$ и параметром масштаба $\lambda = 2$. Моделирование случайной величины проводили с помощью метода обратных функций. К полученным выборкам применяли корневую оценку плотности распределения для проверки соответствия оценки истинной плотности. Исследования проводили для пяти источников информации: одного с полной информацией и четырех с цензурированными данными. Схема моделирования цензурирования изображена на рис. 2.

Принятое число наблюдений n для каждого источника — 100. Длина интервала цензурирования по умолчанию — 0,1.

Корневые оценки цензурированной информации интегральным методом изображены на рис. 3. Графики представлены для числа гармоник m, равного 2, 4 и 6.

Качество оценивания зависит от числа гармоник и от длины интервала группирования данных. В ходе исследования выяснилось, что чем меньше интервал группирования, тем точнее оценка. Существует число гармоник, при котором достигается оптимальная оценка плотности. В данном случае $m = 4$ — оптимальное число гармоник.

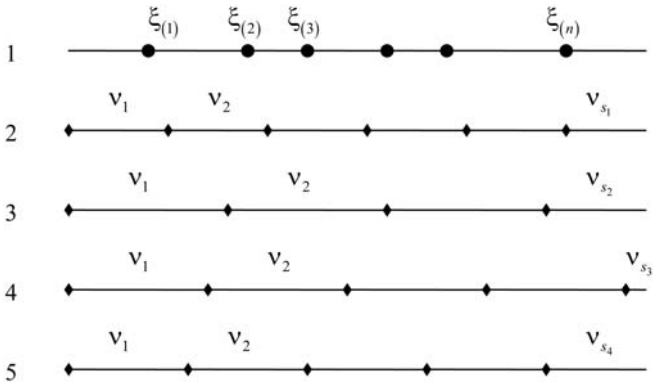


Рис. 2. Моделирование цензурированных данных

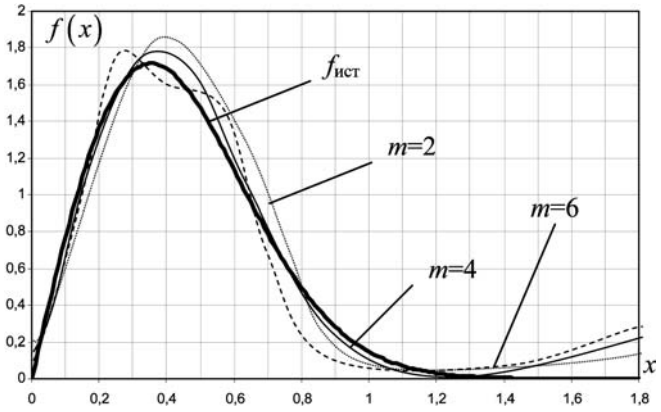


Рис. 3. Корневая интегральная оценка плотности распределения цензурированных данных для разного числа гармоник

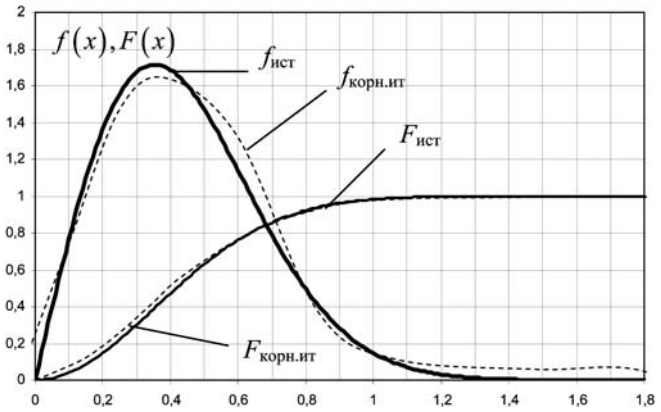


Рис. 4. Корневая итерационная оценка плотности методом псевдоотказов

Корневая оценка плотности методом восстановления отказов изображена на рис. 4, число гармоник $m = 3$.

На рисунке $f_{ист}$ и $F_{ист}$ — истинные значения плотности и функции распределения, $f_{корн.ит}$ и $F_{корн.ит}$ — их оценки методом восстановления отказов.

Длина интервала цензурирования сильно влияет на качество оценки. Оценка тем точнее, чем меньше длина интервала цензурирования.

Заключение

В работе приведены и исследованы две модификации корневой оценки плотности распределения случайной величины при наличии цензурированных данных. В интегральном методе используется классическая форма ФП при наличии цензу-

ры. В этом случае методика корневой оценки существенно усложняется.

Идея итерационного алгоритма заключается в моделировании псевдонаблюдений, т. е. восстановлении неизвестных значений согласно распределению, оцененному на предыдущем шаге. В этом случае схема корневой оценки эквивалентна алгоритму оценивания по полным статистическим данным.

Результаты исследований показали, что качество оценивания итеративным методом немногим уступает качеству интегрального метода, при этом его несомненное преимущество состоит в простоте применения.

Список литературы

1. Антонов А. В., Чепурко В. А. Построение непараметрической плотности распределения на основании цензурированной информации // Надежность. 2005. № 2. С. 3—13.

2. **Богданов Ю. И.** Основная задача статистического анализа данных: корневой подход. М: Изд. МИЭТ, 2002. 96 с.

3. **Крянев А. В., Лукин Г. В.** Математические методы обработки неопределенных данных. М.: ФИЗМАТЛИТ, 2003. 216 с.

4. **Ершов А. Н., Чепурко В. А.** Итерационная оценка параметров закона распределения случайной величины при наличии цензурированных данных // Диагностика и прогнозирование состояния сложных систем: сборник научных трудов № 18. Обнинск: Изд. ИАТЭ, 2009. С. 14—22.

Информация

УДК 519.246

Б. Г. Кухаренко, канд. физ.-мат. наук,
ст. науч. сотр., вед. науч. сотр.,
Институт машиноведения РАН, г. Москва,
e-mail: kukharenko@imash.ru,
Д. И. Пономарев, аспирант,
Московский физико-технический институт (ГУ),
e-mail: ponomarev-102@mail.ru

Обнаружение паттернов многомерных временных рядов на основе абстракции данных

Рассматриваются многомерные временные ряды, в которых присутствуют паттерны. Кластеризация временных сечений многомерных временных рядов обеспечивает абстракцию их данных в виде последовательности кодирующих символов — меток кластеров. В этой последовательности символов паттерны многомерных временных рядов представляются как повторяющиеся эпизоды. Для их обнаружения используется алгоритм подсчета не перекрывающихся эпизодов с ограничением продолжительности. Демонстрируется обнаружение паттернов (жестов оператора), представленных в трехмерном управляющем сигнале дистанционного манипулятора.

Ключевые слова: *анализ временных рядов, паттерны, кластеризация на заданное число кластеров, модель смеси Гауссовых распределений, Алгоритм ожидания и максимизации правдоподобия, абстракция данных, повторяющиеся эпизоды, дистанционный манипулятор*

Введение

Для многомерного временного ряда присутствие паттернов проявляется в приблизительно повторяющихся последовательностях элементов, синхронных для компонент этого временного ряда. Паттерн обнаруживается в результате обработки всех компонент временного ряда, и это гарантирует точность его обнаружения. В настоящей работе используется абстракция данных многомерного временного ряда, при которой его временные сечения отображаются в последовательность кодирующих символов — меток кластеров. При этом паттерны многомерного временного ряда отображаются в повторяющиеся эпизоды этой последовательности символов, которые обнаруживаются алгоритмом поиска повторяющихся эпизодов с ограничением их продолжительности. Проекция временных интервалов для повторяющихся эпизодов на исходный многомерный временной ряд упрощает поиск паттернов этого многомерного временного ряда и повышает точность их обнаружения.

Паттерны многомерного временного ряда и повторяющиеся эпизоды последовательностей символов

Обнаружение паттернов многомерных временных рядов состоит в поиске почти совпадающих последовательностей элементов, синхронных для компонент временного ряда $x[\overline{1, M}, \overline{1, N_0}]$, $N_0 \gg 1$. Для заданного временного ряда $x[j, \overline{1, N_0}]$, $j = \overline{1, M}$, последовательность S — это выборка длиной $N \ll N_0$, являющаяся непрерывной частью $x[j, \overline{1, N_0}]$. Все последовательности $S_k[\overline{1, N}]$, $k = \overline{1, (N_0 - N + 1)}$, получаются из временного ряда $x[j, \overline{1, N_0}]$ сдвигом временного окна длины N. Подобие двух последовательностей S_k, S_l с заданными k, $l = \overline{1, (N_0 - N + 1)}$ определяется величиной некоторой меры расстояния $\mathrm{dist}(S_k, S_l)$, т. е. они представляют паттерн, если $\mathrm{dist}(S_k, S_l) < R$, где R — значение порога [1]. Для обнаружения паттернов компонент временного ряда $x[j, \overline{1, N_0}]$, $j = \overline{1, M}$, требуется варьирование длины N временных последовательностей [2]. Для компонент временного ряда присутствие паттернов, представленных приблизительно повторяющимися последовательностями элементов, проявляется в ее колебательном изменении. Одночастотная аппроксимация компоненты временного ряда по методу Прони оценивает ее характерный период колебаний — временной масштаб для поиска паттернов [3]. Если паттерны компоненты временного ряда повторяются без промежутков, то у нее наблюдаются режимы колебаний [4].

При использовании абстракции данных появляется возможность обнаружить синхронные паттерны для всех компонент временного ряда [5—6]. Абстракция данных обеспечивается, например, при кластеризации временных сечений $\{x[\overline{1, M}; i], i = \overline{1, N_0}\}$ многомерного временного ряда $x[\overline{1, M}; \overline{1, N_0}]$ на заданное число кластеров посредством алгоритма k

средних [7]. Общий подход к кластеризации временных сечений многомерного временного ряда основан на модели смеси Гауссовых распределений [8]. Он использует алгоритм ожидания и максимизации правдоподобия [9]. Метки кластеров используются для классификации и кодирования временных сечений многомерного временного ряда в виде событий. В результате многомерный временной ряд дискретно представляется как последовательность событий $\langle (E[1], t[1]), (E[2], t[2]), ..., (E[N_0], t[N_0])\rangle$ (N_0 — число элементов для компонент временного ряда), показывающих переходы между кластерами. Эпизод — это некоторая частичная последовательность событий (во времени). Паттерны, которые обнаруживаются в последовательности событий, называются эпизодами. Отображение временных интервалов, соответствующих повторяющимся эпизодам последовательности событий, на исходный многомерный временной ряд определяет паттерны этого ряда (синхронные для всех его компонент).

В каждом событии $(E[i], t[i])$, $i = \overline{1, N_0}$, $E[i]$ обозначает тип события, а $t[i]$ — время, когда происходит событие. Типы событий $E[i]$ выбираются из конечного набора. Последовательность событий упорядочена относительно времени возникновения каждого события, т. е. $\forall i = \overline{1, N_0 - 1}$, $t[i] < t[i + 1]$. Эпизод α — триплет $(V_\alpha, \leqslant_\alpha, g_\alpha)$, где V_α — набор узлов, $\leqslant_\alpha$ — частичный порядок на V_α, и $g_\alpha : V_\alpha \to E$ — отображение, ассоциирующее каждый узел с типом событий. Интерпретация эпизода в том, что события в $g_\alpha(V_\alpha)$ должны происходить в порядке, описываемом посредством $\leqslant_\alpha$. Размер α, обозначаемый как $|\alpha|$, является $|V_\alpha|$. Эпизод α является последовательным, если отношение $\leqslant_\alpha$ является полным порядком (т. е. $\forall u, v \in V_\alpha$, $u \leqslant_\alpha v$ или $v \leqslant_\alpha u$).

Для заданной последовательности $\langle (E[1], t[1]), (E[2], t[2]), ..., (E[N_0], t[N_0])\rangle$ появление эпизода $\alpha = (V_\alpha, \leqslant_\alpha, g_\alpha)$ — это отображение $h : V_\alpha \to \{1, ..., N_0\}$, такое что:

1. $\forall v \in V_\alpha$, $g_\alpha(v) = E[h(v)]$;
2. $\forall u, v \in V_\alpha$, **if** $u \leqslant_\alpha v$ **then** $t[h(u)] \leqslant t[h(v)]$.

Эпизод β является подэпизодом для эпизода α, если все типы событий β имеются в α и если частичный порядок среди типов событий β тот же, что для соответствующих типов событий в α. То есть для $\alpha = (V_\alpha, \leqslant_\alpha, g_\alpha)$ и $\beta = (V_\beta, \leqslant_\beta, g_\beta)$ β является подэпизодом для эпизода α, если существует взаимно однозначное соответствие $f_{\beta\alpha} : V_\beta \to V_\alpha$, такое что

1. $\forall v \in V_\beta$, $g_\beta(v) = g_\alpha(f_{\beta\alpha}(v))$;
2. $\forall v, w \in V_\beta$, **if** $v \leqslant_\beta w$ **then** $f_{\beta\alpha}(v) \leqslant_\alpha f_{\beta\alpha}(w)$.

Частота эпизода представляет меру того, как часто эпизод появляется в последовательности событий. У повторяющегося эпизода частота превышает некоторый специфицированный порог. Цель анализа последовательности событий состоит в поиске всех повторяющихся эпизодов. Однако подсчет всех возможных появлений эпизода является неэффективным. В работе [10] частота эпизода определяется как число окон фиксированной длины (поверх последовательности символов), в которых эпизод появляется, по крайней мере, один раз. Такой подсчет не связан непосредственно с интуитивным понятием частоты эпизода, а именно, с числом его появлений. Поэтому в [11] предложена мера частоты как подсчет неперекрывающихся появлений эпизода. Появления эпизода считаются не перекрывающимися, если никакое событие, ассоциированное с одним эпизодом, не появляется между событиями, ассоциированными с другим. Пусть эпизод из N узлов $\alpha = (V_\alpha, \leqslant_\alpha, g_\alpha)$, где $V_\alpha = \{v[1], ..., v[N]\}$. Два появления h_1 и h_2 эпизода α считаются неперекрывающимися, если либо

1. $\forall v[i]$, $v[j] \in V_\alpha$, $h_2(v[i]) > h_1(v[j])$;

либо

2. $\forall v[i]$, $v[j] \in V_\alpha$, $h_1(v[i]) > h_2(v[j])$.

Коллекция появлений эпизода α считается неперекрывающейся, если в ней каждая пара появлений этого эпизода является неперекрывающейся. Соответствующая частота для эпизода α определяется как кардинальность наибольшего множества не перекрывающихся появлений α в данной последовательности событий [11].

В конкретной последовательности события, сильно отдаленные во времени, не связаны друг с другом. Появление таких отдаленных событий не должно учитываться при подсчете частоты эпизодов. Следовательно, требуются некоторые временные ограничения на продолжительность эпизодов. Одним таким временным ограничением является ограничение продолжительности эпизода (episode expiry constraint): требуется, чтобы все события эпизода появились в пределах интервала времени T_X. При этом ограничении частота определяется как число неперекрывающихся появлений эпизода в последовательности событий, таких что их продолжительность меньше T_X. Однако продолжительность эпизода — это его характеристика в целом. В пределах заданной продолжительности эпизода события могут появляться в любой момент времени. Более сильное ограничение состоит в определении верхней границы для длительности интервалов между последующими событиями в эпизоде. Однако алгоритмы поиска последовательных эпизодов с ограничением длительности интервалов между последующими событиями не способны подсчитывать эпизоды с однотипными событиями, которые, как правило, появляются после кластеризации временных сечений паттернов многомерных временных рядов, рассматриваемых в настоящей работе. Поэтому ниже рассматриваются только алгоритмы определения неперекрывающихся последовательных эпизодов с ограничением их временной продолжительности T_X.

Алгоритмы определения последовательных эпизодов

Обнаружение повторяющихся эпизодов является итеративной задачей, в которой алгоритм проходит несколько раз по последовательности событий. В каждом проходе определяется частота каждого эпизода из набора кандидатов. После каждого прохода текущий набор повторяющихся эпизодов из N узлов используется для генерирования кандидатов из $N + 1$ узлов. Алгоритмы подсчета эпизодов основаны на автоматах с конечным числом состояний. Появление последовательного эпизода отслеживается с использованием автомата с конечным числом состояний, который принимает очередное событие из эпизода и переходит в следующее состояние.

На рис. 1 приведен алгоритм 1 для подсчета неперекрывающихся появлений последовательных эпизодов [11]. Для подсчета неперекрывающихся появлений набора последовательных эпизодов требуется только один автомат на эпизод. Этот автомат отслеживает самое левое появление эпизода при проходе последовательности событий слева направо. Пусть $\alpha[j]$ обозначает тип события j-го узла эпизода α. Для каждого эпизода α в наборе кандидатов должен присутствовать соответствующий автомат. Первоначально автомат ожидает $\alpha[1]$ (т. е. первый тип событий эпизода α). Обнаружив $\alpha[1]$, автомат переходит в состояние 1 и ждет $\alpha[2]$. Таким образом, автомат, ожидающий $\alpha[j]$, после подсчета соответствующего события переходит в его j-е состояние и ждет $\alpha[j + 1]$. Появление эпизода α завершено, когда автомат достигает его N-го состояния (финального). В этот момент счетчик эпизодов инкрементируется на 1, и автомат повторно инициализируется, чтобы опять ожидать $\alpha[1]$. При каждом проходе по последовательности событий алгоритм получает частоту нескольких кандидатов в эпизоды. Для того чтобы иметь эффективный доступ к автоматам, используется структура данных *waits*(.). Например, для типа событий A все автоматы, которые ждут события типа A, хранятся в списке *waits*(A). Элементы списка *waits*(A) — пары в форме (α, j), которая означает, что автомат для эпизода α ждет события типа A, чтобы перейти в его j-е состояние.

Алгоритм делает один проход по последовательности состояний. Когда алгоритм встречает следующее событие, например $(E[i], t[i])$, этот алгоритм просматривает список *waits*($E[i]$) и делает соответствующие переходы для всех автоматов: для каждого (α, j) в *waits*($E[i]$), где $j \neq |\alpha|$, из списка удаляется (α, j) и в список *waits*($\alpha[j + 1]$) добавляется $(\alpha, j + 1)$. Если $\alpha[j + 1]$ равно $\alpha[j]$, было бы неправильным добавлять автомат непосредственно в список *waits*($\alpha[j + 1]$), поскольку алгоритм в настоящий момент перебирает тот же список. Поэтому используется временный список *bag*, который содержит все такие автоматы. Позже содержимое списка *bag* выгружается в список *waits*($\alpha[j + 1]$). В случае, когда $j = |\alpha|$, частота эпизода α.*freq* инкрементируется на 1. Автомат переустанавливается в состояние $j = 1$ и добавляется к списку *waits*($\alpha[1]$).

Алгоритм 2 (рис. 2) для генерирования кандидатов использует набор повторяющихся эпизодов

Алгоритм 1

Input: Event sequence $s = \langle (E[1],t[1]),(E[2],t[2]),...,(E[N_0],t[N_0]) \rangle$, set C of candidate N-node episodes, $N << N_0$, frequency threshold $\lambda_{\min} \in [0,1]$
Output: The set F of frequent serial episodes in C
for all event types A **do**
  Initialize $waits(A) = \phi$
end for
for all $\alpha \in C$ **do**
  Add $(\alpha,1)$ to $waits(\alpha[1])$
  Initialize $\alpha.freq = 0$
end for
Initialize $bag = \phi$
for $i = 1$ **to** N_0 **do**
  for all $(\alpha, j) \in waits(E[i])$ **do**
    Remove (α, j) from $waits(E[i])$
    Set $j' = j + 1$
    if $j' == (N + 1)$ **then**
      Set $j' = 1$
    end if
    if $\alpha[j'] = E[i]$ **then**
      Add (α, j') to bag
    else
      Add (α, j') to $waits(\alpha[j'])$
    end if
    if $j == N$ **then**
      Update $\alpha.freq = \alpha.freq + 1$
    end if
  end for
  Empty bag into $waits(E[i])$
end for
return $F = \{\alpha \in C | \alpha.freq > N_0 \lambda_{\min}\}$

Рис. 1. Алгоритм подсчета неперекрывающихся последовательных эпизодов

Алгоритм 2

Input: L_{N-1} list of $(N-1)$-node frequent episodes
Output: C_N list of k-node candidate episodes
Initialize $C_N = \phi$
Sort L_{N-1} in order of event types
Initialize $i = 0$
while $i < |L_{N-1}|$ **do**
  Initialize $j = i + 1$; $\alpha_1 = L_{N-1}(i)$; $e_{list} = \langle \alpha_1[N-1] \rangle$
  Initialize $\alpha_2 = L_{N-1}(j)$
  while $(\alpha_1[1] == \alpha_2[1],...,\alpha_1[N-2] == \alpha_2[N-2])$ **do**
    $e_{list} = e_{list} \cup \alpha_2[N-1]$
    $j = j + 1$
    $\alpha_2 = L_{N-1}(j)$
  end while
  for all event types $e_1 \in e_{list}$ **do**
    for all event types $e_2 \in e_{list}$ **do**
      $\alpha_{new} = \langle \alpha_1[1,...,(N-2)], e_1, e_2 \rangle$
      генерирование нового кандидата
      Initialize $subepisode\_chk = true$
      for $m = 1$ **to** $N - 2$ **do**
        $\alpha_{test} = \alpha_{new}[1,...,m-1,m+1,...,N]$
        if $\alpha_{test} \notin L_{N-1}$ **then**
          Set $subepisode\_chk = false$
        end if
      end for
      if $subepisode\_chk = true$ then
        $C_N = C_N \cup \alpha_{new}$
      end if
    end for
  end for
end while
return C_N

Рис. 2. Алгоритм генерирования кандидатов

Алгоритм 3

Input: Event sequence $s = \langle (E[1], t[1]), (E[2], t[2]), \ldots, (E[N_0], t[N_0]) \rangle$, set C of candidate N-node episodes, $N << N_0$, frequency threshold $\lambda_{\min} \in [0,1]$, expiry time T_X

Output: The set F of frequent serial episodes in C

```
for all event types A do
    Initialize waits(A) = ϕ
end for
for all α ∈ C do
    Add (α,1) to waits(α[1])
    Initialize α.freq = 0
end for
Initialize bag = ϕ
for i = 1 to N_0 do
    for all (α, j) ∈ waits(E[i]) do
        if j == 1 then
            Update α.init[1] = t[i]
        else
            Update α.init[j] = α.init[j − 1]
            Remove (α, j) from waits(E[i])
        end if
        if j < N then
            if α[j + 1] = E[i] then
                Add (α, j + 1) to bag
            else
                Add (α, j + 1) to waits(α[j + 1])
            end if
        end if
        if (j == N) & (t[i] − α.init[1] ≤ T_X) then
            Update α.freq = α.freq + 1
            for all 1 ≤ k < |α| do
                Remove (α, k + 1) from waits(α[k + 1])
                Remove (α, k + 1) from bag
            end for
        end if
    end for
    Empty bag into waits(E[i])
end for
return F = {α ∈ C | α.freq > N_0 λ_min}
```

Рис. 3. Алгоритм подсчета неперекрывающихся последовательных эпизодов с ограничением продолжительности

из N узлов для построения кандидатов из $(N + 1)$ узлов. Это делается следующим образом. Пусть α и β — два повторяющихся эпизода из N узлов, у которых первые $(N - 1)$ узлов одинаковые. Два потенциальных кандидата из $(N + 1)$ узлов генерируются, присоединяя к эпизоду α N-й узел эпизода β и присоединяя к эпизоду β N-й узел эпизода α. Каждый новый эпизод объявляется как кандидат из $(N + 1)$ узлов, если уже известно, что все его подэпизоды из N узлов повторяются. Это генерирование кандидатов не пропустит никаких кандидатов, потому что для неперекрывающихся появлений как меры частоты все подэпизоды повторяющегося последовательного эпизода должны сами быть повторяющимися.

Как отмечалось, для рассматриваемого ниже примера обнаружения паттернов управляющего сигнала дистанционного манипулятора на основе абстракции данных имеет смысл подсчитывать только те повторения эпизода, которые не слишком широко простираются во времени (episode expiry constraint). Алгоритм 3 (рис. 3) модифицирует алгоритм 1 (см. рис. 1), чтобы подсчитывать последовательные эпизоды с ограничением продолжительности. Он считает максимальное число неперекрывающихся появлений эпизода, где каждое появление покрывает временной интервал, меньший заданного порога T_X. Отслеживание только самых левых появлений эпизода, как это делается в алгоритме 1 (см. рис. 1), не подходит для подсчета последовательных эпизодов с ограничением продолжительности. Более эффективно отслеживать первое появление каждого типа событий. Поэтому используется термин "самое внутреннее появление" (inner most occurrence) для выделения эпизода, начинающегося с самого позднего появления события того типа, с которого стартует этот эпизод и в котором после этого впервые появляются все последующие типы событий. Поскольку все автоматы эпизода, ожидающего одного и того же типа событий, заменяются самым последним автоматом, можно иметь самое большее N автоматов ожидающих эпизода, где N — число узлов в эпизоде. Для того чтобы вычислить протяженность появления, время инициализации каждого автомата хранится в массиве *init*[] на уровне эпизода. В момент, когда автомат достигает его финального состояния, выполняется проверка продолжительности эпизода с использованием времени инициализации. Если протяженность появления эпизода меньше T_X, частота $\alpha.freq$ инкрементируется на 1 и удаляются все автоматы для эпизода, за исключением автомата в состоянии $j = 1$. В противном случае удаляется только текущий автомат, а другим разрешается продолжать переходы между состояниями. Алгоритм 2 (см. рис. 2) для генерирования кандидатов применяется также для обнаружения неперекрывающихся последовательных эпизодов с ограничением времени продолжительности (expiry time constraint) посредством алгоритма 3 (рис. 3).

Обнаружение паттернов управляющего сигнала дистанционного манипулятора на основе абстракции данных

В последнее время широкое распространение получают дистанционные манипуляторы, передающие движения руки оператора в трехмерном пространстве. В настоящей работе исследуются управляющие сигналы разрабатываемого авторами манипулятора (рис. 4), чувствительным элементом которого является прецизионный трехосевой MEMS-акце-

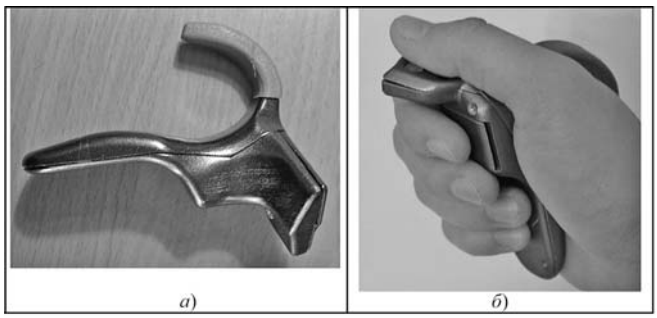


Рис. 4. Дистанционный манипулятор с прецизионным акселерометром:
а — внешний вид; *б* — манипулятор в руке оператора

лерометр (Micro-Electro-Mechanical System) [12]. Акселерометр обладает чувствительностью к земной гравитации, поэтому изменение наклона манипулятора относительно направления ускорения g свободного падения приводит к изменению значений проекций этого ускорения на чувствительные оси x, y, z этого акселерометра. Значения этих проекций являются управляющими сигналами для позиционирования курсора (стандартно управляемого компьютерной мышью) на экране дисплея компьютера.

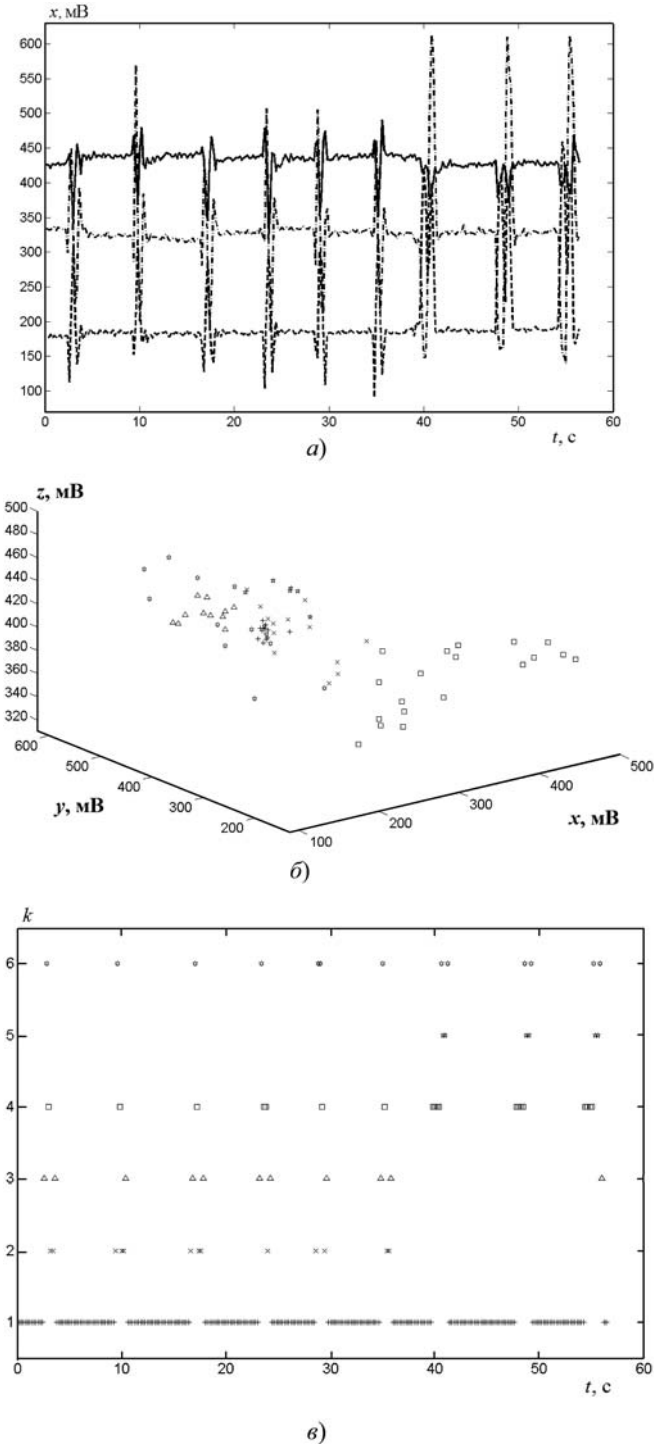


Рис. 5. Абстракция данных:
а — исходный трехмерный временной ряд; *б* — кластеризация точек, представляющих временные сечения трехмерного ряда; *в* — кодирование временных сечений трехмерного ряда; k — номер кластера

Эти записи (временные ряды) управляющих сигналов дистанционного манипулятора могут использоваться для обнаружения паттернов. Паттерны записи управляющих сигналов манипулятора представляют произвольные движения — жесты руки оператора. Конкретные паттерны временного ряда повторяются, поскольку последовательности одинаковых жестов руки оператора определяются особенностями построения этих движений [13].

Для обработки используется запись с чувствительных осей x, y, z MEMS-акселерометра для двух последовательностей однотипных жестов руки оператора (рис. 5, *а*). Сигналы $x = x(t)$, $y = y(t)$ и $z = z(t)$ на рис. 5, *а* показаны, соответственно, пунктиром "--", штрих пунктиром "·-" и сплошной линией "—". Интервал дискретизации времени 200 мс (это соответствует частоте дискретизации 5 Гц), длина записей $N_0 = 282$. Первый тип жестов — это круговые движения с паузой (общая продолжительность круговых движений от 0 до 37 с). Второй тип жестов представляет собой наклоны влево—вправо с паузой (общая продолжительность наклонов от 37 до 57 с). На рис. 5, *б* показана кластеризация точек, представляющих временные сечения трехмерного ряда, на шесть кластеров с использованием алгоритма k средних: "+" обозначает точки первого кластера, представляющие временные сечения для пауз между жестами, "×" — второго кластера, "Δ" — третьего кластера, "□" — четвертого кластера, "пятиугольник" — пятого кластера, "шестиугольник" — шестого кластера. Эта кластеризация означает классификацию временных сечений трехмерного ряда на рис. 5, *а*. На рис. 5, *в* представлена абстракция трехмерного временного ряда: временные сечения трехмерного ряда (рис. 5, *а*) кодируются как последовательность меток кластеров, т. е. цифр из набора {1, 2, 3, 4, 5, 6}.

На рис. 6, *а* и рис. 6, *б*, соответственно, показаны гистограммы распределения временных сечений трехмерного временного ряда по шести кластерам для круговых движений с паузой (общая продолжительность круговых движений от 0 до 37 с) и последующих наклонов влево—вправо с паузой (общая продолжительность наклонов от 37 до 57 с).

Паттерны исходного трехмерного временного ряда (см. рис. 5, *а*) представляются повторяющимися компактными эпизодами из последовательности событий на рис. 5, *б*, т. е. с ограничением временной продолжительности. Для трехмерного временного ряда паттерны обнаруживаются при ограничении длины эпизодов $N \leqslant 16$ и при ограничении их временной продолжительности $T_X = 1{,}8$ с.

Первый повторяющийся эпизод имеет длину 8: $1 \to 3 \to 6 \to 4 \to 2 \to 2 \to 3 \to 1$ ($\to$ показывает переходы между кластерами) и его повторяемость 3 (рис. 7, *а*). На рис. 5, *в* он представлен эпизодами ⟨(1,2.4), (3,2.6), (6,2.8), (4,3.0), (2,3.2), (2,3.4), (3,3.6), (1,3.8)⟩ и ⟨(1,34.6), (3,34.8), (6,35.0), (4,35.2), (2,35.4), (2,35.6), (3,35.8), (1,36.0)⟩, а также он подэпизод эпизода длиной 9: ⟨(1,16.4), (2,16.6), (3,16.8), (6,17.0), (4,17.2), (2,17.4), (2,17.6), (3,17.8), (1,18.0)⟩.

Кроме этого, он отличается от эпизодов ⟨(1,23.0), (3,23.2), (6,23.4), (4,23.6), (4,23.8), (2,24.0), (3,24.2), (1,24.4)⟩ и ⟨(1,28.4), (2,28.6), (6,28.8), (6,29.0), (4,29.2), (2,29.4), (3,29.6), (1,29.8)⟩ на 1 и от эпизода ⟨(1,28.4), (2,28.6), (6,28.8), (6,29.0), (4,29.2), (2,29.4), (3,29.6), (1,29.8)⟩ на 2 по мере редактирования (по числу символов в первой последовательности, которые нужно изменить для получения второй последовательности) [14].

Второй повторяющийся эпизод имеет длину 10: $1 \to 4 \to 4 \to 4 \to 4 \to 6 \to 5 \to 5 \to 6 \to 1$ (рис. 7, *б*), и его повторяемость 3. На рис. 5, *в* он представлен эпизодами ⟨(1,39.6), (4,39.8), (4,40.0), (4,40.2), (4,40.4), (6,40.6), (5,40.8), (5,41.0), (6,41.2), (1,41.4)⟩ и ⟨(1,47.6), (4,47.8), (4,48.0), (4,48.2), (4,48.4), (6,48.6), (5,48.8), (5,49.0), (6,49.2), (1,49.4)⟩, а также он подэпизод эпизода длиной 11: ⟨(1,54.2), (4,54.4), (4,54.6), (4,54.8), (4,55.0), (6,55.2), (5,55.4), (5,55.6), (6,55.8), (3,56.0), (1,56.2)⟩.

Отображение временных интервалов для повторяющихся эпизодов на трехмерный временной ряд на рис. 5, *а* показывает, что в нем присутствуют два паттерна, представляющие два типа жестов руки оператора.

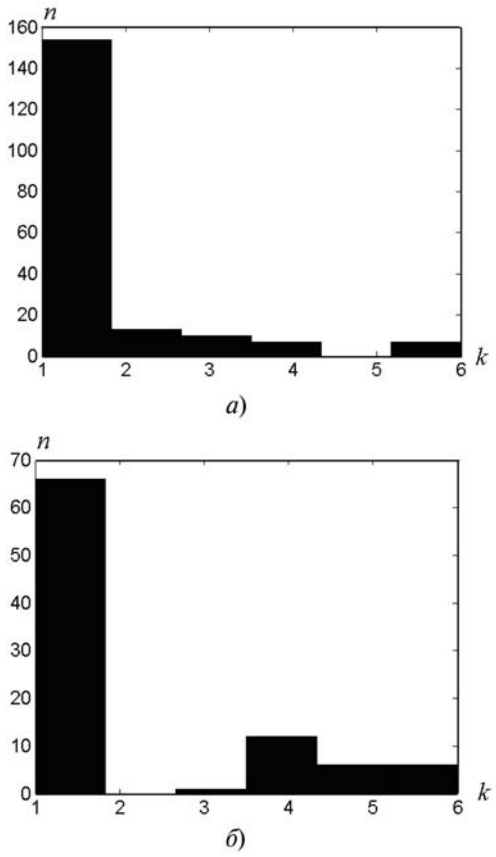


Рис. 6. Гистограммы распределения временных сечений трехмерного временного ряда по шести кластерам:
а — круговых движений с паузой; *б* — наклонов влево—вправо с паузой

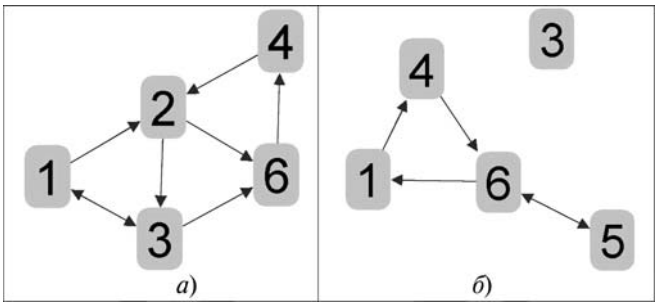


Рис. 7. Графы повторяющихся эпизодов:
а — первого; *б* — второго

Заключение

При обнаружении паттернов в многомерных временных рядах каждая компонента анализируется по отдельности, и паттерн многомерного ряда фиксируется после определения синхронных паттернов многомерного временного ряда. Временные затраты на обработку пропорциональны числу компонент M многомерного временного ряда (и растут линейно с ростом M). Используемая в настоящей статье кластеризация временных сечений является процедурой агрегирования компонент многомерного временного ряда в последовательность эпизодов, поэтому временные затраты на поиск повторяющихся эпизодов не зависят от числа компонент многомерного временного ряда. Таким образом, с точки зрения временных затрат обнаружение последовательности повторяющихся эпизодов является более эффективным, чем поиск паттернов многомерного временного ряда.

Список литературы

1. **Mueen A., Keogh E., Zhu Q., Cash S., Westover B.** Exact discovery of time series motifs // Proceedings of the SIAM International Conference on Data Mining (SDM 2009). American Statistical Association (ASA). 2009. P. 473—484.
2. **Mueen A., Keogh E. J.** Online discovery and maintenance of time series motifs // Proceedings of the 16th ASM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2010), July 25—28, 2010. Washington, DC: ACM. 2010. P. 1089—1098.
3. **Кухаренко Б. Г., Пономарев Д. И.** Использование метода Прони для оценки временного масштаба при обнаружении паттернов во временных рядах // Информационные технологии. 2012. № 1. С. 37—42.
4. **Кухаренко Б. Г., Пономарев Д. И.** Аппроксимация смесью Гауссовых распределений в модели переключающегося фильтра Калмана для идентификации режимов колебаний временных рядов // Информационные технологии. 2012. № 7. С. 2—6.
5. **Patnaik D., Marwah M., Sharma R. K., Ramakrishnan N.** Sustainable operation and management of data center chillers using temporal data mining // Proceedings of the 15th ACM SIGKDD In-

ternational Conference on Knowledge Discovery and Data Mining (KDD 2009). June 28 — July 1, 2009. Paris, France: ACM. P. 1305—1314.

6. **Hao M., Marwah M., Janetzko H., Sharma R., Keim D. A., Dayal U., Patnaik P., Ramakrishnan N.** Visual Analysis of Frequent Patterns in Large Time Series // Proceedings of the IS & T/SPIE Conference on Visualization and Data Analysis (VDA). San Francisco, CA, Jan 2011.

7. **Han J., Kamber M.** Data Mining: Concepts and Techniques. 2nd ed. New York: Morgan Kaufmann Publishers, 2006.

8. **Bouman C. A.** CLUSTER: An Unsupervised Algorithm for Modeling Gaussian Mixtures. Purdue University, West Lafayette: School of Electrical Engineering. 2005. P. 1—20.

9. **Dempster A., Laird N. M., Rubin D. B.** Maximum likelihood from incomplete data via the EM algorithm // Journal of the Royal Statistical Society B. 1977. V. 39. N 1. P. 1—38.

10. **Mannila H., Toivonen H., Verkamo A. I.** Discovery of frequent episodes in event sequences // Data Mining and Knowledge Discovery. 1997. V. 1. N 3. P. 259—289.

11. **Laxman S., Sastry P. S., Unnikrishnan K.** Discovering frequent episodes and leaning hidden Markov models: A formal connection // IEEE Transactions on Knowledge and Data Engineering. 2005. V. 17. N 11. P. 1505—1517.

12. **Кухаренко Б. Г., Пономарев Д. И.** Дистанционный манипулятор на основе MEMS-акселерометра в качестве чувствительного элемента // Нано- и микросистемная техника. 2012. № 2. С. 49—54.

13. **Бернштейн Н. А.** Очерки по физиологии движений и физиологии активности. М.: Медицина, 1966.

14. **Левенштейн В. И.** Двоичные коды с исправлением выпадений, вставок и замещений символов // Докл. АН СССР. 1965. Т. 163. № 4. С. 845—848.

УДК 004.934

А. В. Савченко, канд. техн. наук, доц.,
e-mail: avsavchenko@hse.ru,
Национальный исследовательский университет
Высшая школа экономики, г. Нижний Новгород

Адаптивный алгоритм распознавания речи на основе метода фонетического декодирования слов в задаче голосового управления[1]

Ставится и решается задача автоматического распознавания речи для системы голосового управления. Предложен адаптивный алгоритм распознавания, на первом этапе которого для всех выделенных слогов распознаются гласные фонемы, а на втором происходит уточнение произнесенных слогов. Показано, что такой подход приводит к созданию высоконадежной обучаемой системы, в которой продолжительность настройки под диктора на порядок ниже аналогичного показателя для существующих систем. Экспериментально продемонстрировано, что предложенный подход характеризуется удовлетворительной точностью и высокой вычислительной эффективностью.

Ключевые слова: *автоматическое распознавание речи, системы голосового управления, слоговая фонетика, метод фонетического декодирования, принцип минимума информационного рассогласования*

[1] Работа выполнена при финансовой поддержке Минобрнауки РФ по государственному контракту № 07.514.11.4137 ФЦП "Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007—2013 годы".

Введение

В настоящее время все более актуальной становится задача создания автоматизированных систем, которые не только обеспечивают сбор информации от некоторого объекта, но и используют полученную информацию для последующего управления объектом. При этом большая часть работы по управлению выполняется самой автоматизированной системой. Однако нередко требуется обеспечить человеку доступ к подсистеме управления для того, чтобы позволить ему вносить некие корректировки в принимаемые машиной решения. Соответственно, требуется некий механизм, позволяющий человеку получать информацию от управляемого объекта и передавать свои корректировки вычислительной технике. К сожалению, прямой доступ к управляемому объекту зачастую затруднен, а нередко и просто невозможен. Поэтому в настоящее время при обеспечении коммуникации с удаленными объектами используется мобильная связь [1]. Тогда основной проблемой становится обеспечение максимальной надежности неотъемлемой части таких систем — модуля автоматического распознавания речи (АРР) [2].

К сожалению, существующие в настоящее время коммерческие продукты используют для решения этой задачи ту же технологию, что и для систем диктовки текста [2]. Несмотря на значительные успехи этой технологии, основанной на аппарате скрытых Марковских моделей (СММ) [3], известные коммерческие системы (такие как Microsoft Voice Command, Windows Speech Recognition, Nuance Dragon NaturallySpeaking, Google Voice Search и т. п.) еще не достигли приемлемого уровня надежности. Зачастую они оказываются недостаточно гибкими: не могут за короткое время перенастроиться на нового диктора и/или рабочий словарь и полностью поддерживают только английский язык.

Поиску путей преодоления указанных проблем и посвящена настоящая работа. В ней предложен новый подход к построению надежных систем голосового управления, основанный на принципе минимума информационного рассогласования (МИР) Кульбака—Лейблера [4], методе фонетического декодирования [5] и аппарате слоговой фонетики [6].

Задача автоматического распознавания голосовых команд

Пусть задано множество из $L > 1$ эталонных команд $\{X_l\}$, где $l = \overline{1, L}$ — номер слова-эталона. Согласно общепринятому фонетическому подходу [2, 7], каждая эталонная команда разбивается (автоматически или, чаще всего, вручную) на последовательность фонем (транскрипцию) $X_l = \{c_{l,1}, c_{l,2}, ..., c_{l,L_l}\}$. Здесь L_l — длительность команды (в фонемах), а числа $c_{l,j} \in \{1, ..., R\}$ — номера фонем из некоторого фонетического алфавита $\{\mathbf{x}^*_r\}$, $r = \overline{1, R}$, где R — число фонем в алфавите. Задача состоит в том, чтобы поступившему на вход речевому сигналу X поставить в соответствие наиболее близкое к нему слово-эталон.

Для решения задачи на первом этапе сигнал X разбивается на ряд непересекающихся квазистационарных сегментов $\{\mathbf{x}(t)\}, t = \overline{1, T}$, длиной $\tau = 10...15$ мс, где T — общее число сегментов. Далее каждый парциальный сигнал $\mathbf{x}(t)$ рассматривается в пределах конечного списка фонем $\{\mathbf{x}^*_r\}$ и отождествляется с той из них, которая отвечает принципу минимума заданной исследователем меры близости $\rho(\mathbf{x}^*_r, \mathbf{x}(t))$ между сигналом $\mathbf{x}(t)$ и эталоном $\mathbf{x}^*_r$:

$$v(r) = \underset{r \in \{1, ..., R\}}{\arg\min}\ \rho(\mathbf{x}^*_r, \mathbf{x}(t)). \qquad (1)$$

Для выбора меры близости в (1) воспользуемся Гауссовой (нормальной) аппроксимацией закона распределения речевого сигнала на интервалах его квазистационарности $\tau \approx \text{const}$. Известно [5, 7], что в этом случае критерий, основанный на методе ближайшего соседа (1) и принципе МИР [4] с решающей статистикой вида

$$\rho_{KL}(\mathbf{x}^*_r, \mathbf{x}(t)) = \frac{1}{F}\sum_{f=1}^{F}\left(\frac{G_x(f)}{G_r(f)} - \ln\frac{G_x(f)}{G_r(f)} - 1\right), \qquad (2)$$

эквивалентен оптимальному методу максимального правдоподобия. Здесь $G_x(f)$ — выборочная оценка спектральной плотности мощности (СПМ) входного сигнала $\mathbf{x}(t)$ в функции дискретной частоты f; $G_r(f)$ — СПМ эталона r-й фонемы $\mathbf{x}^*_r$; F — верхняя граница частотного диапазона речевого сигнала или используемого канала связи.

Оценка СПМ чаще всего осуществляется на основе авторегрессионной (АР) модели [8] речевого сигнала, главное достоинство в задаче АРР которой [2] состоит в возможности предварительной нормировки речевых сигналов по дисперсиям их порождающих процессов. Такая нормировка обусловлена физическими особенностями голосового механизма человека: воздушный поток на входе его модели "акустической трубы" имеет приблизительно одну и ту же интенсивность на интервалах длительностью в целое слово. Тогда отношение СПМ в (2) приобретает вид [7]

$$\frac{G_x(f)}{G_r(f)} = \frac{\left|1 + \sum_{m=1}^{p} a_r(m)\exp(-j\pi mf/F)\right|^2}{\left|1 + \sum_{m=1}^{p} a_x(m)\exp(-j\pi mf/F)\right|^2}. \qquad (3)$$

Здесь p — порядок АР-модели; $j = \sqrt{-1}$; $a_r(m)$ и $a_x(r)$ — оценки АР коэффициентов эталона $\mathbf{x}^*_r$ и входного сигнала $\mathbf{x}(t)$, получаемые на основе алгоритма Левинсона—Дурбина и метода Берга [8].

Отметим несомненное сходство статистики (2) с рассогласования Итакуры—Саито [2], издавна применяемым в задачах АРР, которое, как известно [2], сильно коррелированно с субъективными оценками близости речевых сигналов (MOS, mean opinion score). Отличие состоит лишь в обратном порядке эталонных и входных признаков. В результате (1) совместно с (2) определяет критерий АРР, обладающий свойством оптимальности в байесовском смысле и одновременно связанный с экспертной оценкой MOS.

На втором этапе полученная согласно (1), (2) транскрипция $\{\mathbf{x}^*_{v(1)}, \mathbf{x}^*_{v(2)}, ..., \mathbf{x}^*_{v(T)}\}$ сигнала X обычно выравнивается по темпу речи с транскрипцией каждого слова-эталона для установления временного соответствия между звуками сопоставляемых речевых образов. Для этого можно воспользоваться, например, алгоритмом Dynamic Time Warping [9], основанным на принципах динамического программирования, или вероятностным аппаратом СММ [3]. Наиболее близкая в смысле среднего рассогласования вида (2) после временного выравнивания [10] команда и будет являться решением задачи.

Ограничения канонического подхода

Описанная технология АРР на основе СММ в настоящее время считается каноническим решением задачи и применяется во всех коммерческих системах. Это не удивительно, так как практически все эти системы уходят корнями [11] в исследовательскую группу из университета Карнеги—Меллон, получившую в начале 70-х годов XX века проект

DARPA по созданию системы распознавания речи. К сожалению, практическая реализация этого подхода к задаче голосового управления наталкивается на ряд трудностей.

Во-первых, несмотря на то, что, как говорится в [12], распознавание речи на фонетическом уровне (первый этап) в настоящее время сравнимо по качеству с надежностью распознавания отдельных звуков человеком, такая точность достигается только для очень представительной высококачественной фонетической базы данных. Ее наполнение для каждого разговорного языка не автоматизировано.

Во-вторых, для распознавания на втором этапе требуется, чтобы каждое эталонное слово было разбито на фонемы. Для нескольких тысяч слов такое разбиение выполняется вручную, для остальных слов процесс можно автоматизировать при наличии качественных речевых корпусов длительностью 100 ч (и более) [12].

В-третьих, для реализации алгоритма выравнивания в режиме реального времени и большого словаря эталонов требуются мощности, значительно превосходящие возможности современного персонального компьютера и, тем более, сотового телефона.

Неудивительно, что точная реализация классического подхода стала возможной лишь в проектах больших корпораций (Microsoft, Google, Apple, IBM, Nuance). При этом для распознавания в режиме реального времени и при малопроизводительном оборудовании используются облачные вычисления и технология клиент-сервер. Остальные коллективы вынуждены ограничиться построением систем с небольшим заранее фиксированным словарем.

Наконец, основное ограничение существующих систем заключается в недостаточной точности распознавания [2]. Для преодоления этой проблемы в системах существует возможность настройки под конкретного диктора [2, 11]. К сожалению, такая настройка требует либо произнесения диктором всех слов-эталонов, либо чтения теста в течение длительного времени (не менее 30 мин). А для клиент-серверных решений (таких, как Google Voice Search, Apple SIRI) функциональность настройки ограниченна или отсутствует.

Таким образом, классическая технология АРР не удовлетворяет всем требованиям задачи распознавания речевых команд. Продукты больших корпораций обычно разрабатываются для решения других задач АРР (голосовой поиск, диктовка текстов), которые, с одной стороны, не предъявляют повышенных требований к точности распознавания, а, с другой стороны, являются слишком общими (поддержка различных дикторов, большой словарь, не привязанный к конкретной предметной области). А возможности остальных продуктов чересчур ограниченны (сложность настройки под диктора, длительное время обновления словаря для новой предметной области). Поиск путей преодоления указанных недостатков за счет отказа от динамического выравнивания слов [9] на втором этапе распознавания и является основной целью настоящей работы.

Альтернативный подход к построению систем распознавания голосовых команд

Одним из требований к системам АРР [11] является распознавание ими *естественной* речи. Это требование приводит к существенной вариативности речи и к снижению надежности систем АРР. Можно предположить, что некоторое смягчение этого требования позволит повысить точность распознавания и, как следствие, сократить время диалога с удаленным объектом при голосовом управлении.

Например, известно [11], что наиболее хорошо распознаются ударные гласные. Поэтому в настоящей работе предлагается следующее ограничение [13]: диктор должен произносить все команды с четким выделением каждого слога. Тогда, как известно из исследований в области слоговой фонетики [6], нетрудно разбить входной сигнал на изолированную последовательность слогов и, таким образом, свести задачу распознавания голосовых команд к распознаванию слогов [14] на основе метода фонетического декодирования [5].

Далее предполагаем, что входное слово X разбито на N слогов, причем границы каждого n-го слога ($n = \overline{1, N}$) определены с точностью до номера квазистационарного сегмента ($t_n^{(1)}$, $t_n^{(2)}$) с очевидными ограничениями

$$\forall n \in \{1, ..., N\}\ 1 \leqslant t_n^{(1)} < t_n^{(2)} \leqslant T,$$

$$\forall n \in \{1, ..., N-1\} t_n^{(2)} < t_{n+1}^{(1)}.$$

Тогда второй этап АРР, на котором уже не требуется динамическое выравнивание по длительности, разбивается на две части.

Вначале для каждого n-го слога проводится распознавание только [14] среди гласных звуков, т. е. фонетический алфавит $\{\mathbf{x}_r^*\}$, $r = \overline{1, R}$, состоит из эталонов гласных фонем. Как отмечено выше, распознавание ударных гласных может быть выполнено с достаточно высокой надежностью. Для этого на основе всех $v(t)$, $t = \overline{t_n^{(1)}, t_n^{(2)}}$, требуется принять решение в пользу принадлежности распознаваемого слога к одной из R гласных. Это решение может быть принято на основе комбинирования решающих правил. Воспользуемся простым агрегирова-

нием [14]: n-му слогу ставится в соответствие последовательность частот $\mu_n(r)$, $r = \overline{1, R}$, где

$$\mu_n(r) = \frac{1}{t_n^{(2)} - t_n^{(1)} + 1} \sum_{t_n^{(1)}}^{t_n^{(2)}} \delta(v(t) - r), \qquad (4)$$

$\delta(x)$ — дискретная дельта-функция.

Далее для каждой команды-эталона X_l оценивается ее корреляция с распознаваемым речевым сигналом

$$\mu_l = \begin{cases} \prod_{n=1}^{N} \mu_n(c_{l,n}), & L_l = N; \\ 0, & L_l \neq N. \end{cases} \qquad (5)$$

Тогда решение задачи АРР принимается в пользу слова X^* по следующему критерию:

$$X^* = \underset{X_l,\ l = \overline{1, L}}{\arg\max}\, \mu_l. \qquad (6)$$

Система выражений (4)—(6) и определяет предлагаемый подход к распознаванию голосовых команд. Остановимся подробнее на преимуществах этого подхода.

Во-первых, настройка под нового диктора осуществляется путем записи его произношения всех гласных фонем. Так как их число обычно невелико (R = 6...20), то процедура такой настройки не требует длительного времени. В этой процедуре проявляется еще одно преимущество АР модели акустической трубы — возможность верификации эталонных звуков. Если на вход АР процесса с коэффициентами, оцененными по эталонному сигналу от диктора, подать дискретный дельта-импульс с частотой, равной частоте основного тона речевого сигнала (100...150 Гц), то выходной АР сигнал можно подать на вход звукового устройства ("озвучить"). Если этот сигнал не воспринимается диктором как произнесенный им гласный звук, настройку АР модели для звука следует повторить.

Во-вторых, процедура присвоения каждой эталонной команде последовательности кодов гласных фонем $\{c_{l,1}, c_{l,2}, ..., c_{l,L_l}\}$ может быть также выполнена автоматически по текстовому представлению команды на основе алгоритмов фонетического синтеза речи [15]. В результате процедура настройки на новый словарь полностью автоматизируется.

Наконец, предлагаемая система может постоянно обучаться в процессе своего функционирования. Действительно, отличительной особенностью систем голосового управления является выполнение команды только после подтверждения пользователем правильности ее распознавания. После получения такого подтверждения система сохраняет все выделенные в распознаваемой команде слоги. В дальнейшем эту информацию можно использовать для

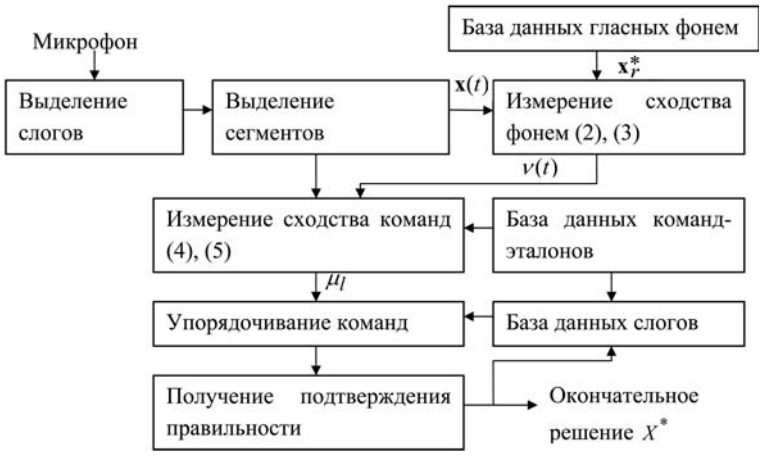


Рис. 1. Функциональная схема устройства для распознавания голосовых команд

уточнения результатов распознавания (6) за счет учета контекстного звучания согласных в слоге. Это уточнение выполняется путем распознавания слогов с использованием канонического подхода к АРР. При этом здесь уже сложное с вычислительной точки зрения динамическое выравнивание выполнять не требуется.

На рис. 1 представлена функциональная схема предлагаемой системы распознавания голосовых команд. В речевом сигнале, поступившем на вход с микрофона, автоматически выделяются слоги с использованием простого амплитудного ограничителя или более сложных алгоритмов [16], основанных на предварительной фильтрации речевого сигнала. Далее каждый слог разбивается на неперекрывающиеся сегменты длительностью τ мс. Для каждого сегмента вычисляется ближайшая в смысле выбранного рассогласования (2), (3) гласная фонема (1). После этого для слога целиком согласно (4) определяется последовательность частот $\mu_n(r)$. Далее для каждого эталонного слова определяется его близость μ_l с входным словом (5) и выбираются все наиболее близкие (6) эталонные команды (с максимальными μ_l). Наконец, все выбранные эталонные команды сортируются по степени близости суммы рассогласований между образующими их слогами и слогами, выделенными во входном слове. Упорядоченные эталоны последовательно предлагаются пользователю для гарантии правильности распознавания. Для получения ответа (вида "да/нет") может быть использована та же самая схема распознавания по образующим слова гласным звукам. После подтверждения правильности распознавания слоги, образующие входную команду, при необходимости добавляются в базу данных слогов.

Результаты экспериментальных исследований

Рассмотрим применение предложенной системы (рис. 1) в задаче распознавания голосовых команд. В качестве словаря эталонов возьмем прайс-лист

одного ресторана быстрого обслуживания, состоящий из 289 слов/словосочетаний на русском языке. Запись речевого сигнала осуществлялась через встроенный в ноутбук микрофон (Realtek High Definition Audio). Каждый записанный речевой сигнал (отдельные звуки и слова/словосочетания из словаря эталонов) сохранялся в виде отдельного звукового wav-файла (моно, частота дискретизации 8000 Гц, 16 бит на отсчет). В качестве предварительной обработки из речевого сигнала удалялись начальные и конечные паузы. Порядок АР модели $p = 20$.

Тестирование системы проводилось группой дикторов разного возраста, пятью мужчинами и пятью женщинами. На предварительном этапе осуществлялась настройка системы под конкретного диктора. В режиме настройки диктор четко проговаривал каждый из 10 гласных звуков русского языка: "а", "е", "ё", "и", "о", "у", "ы", "э", "ю", "я". Для каждой фонемы-эталона АР коэффициенты оценивались по всему сигналу целиком, без его разделения на сегменты. Запись звука повторялась до тех пор, пока синтезированный АР процесс по звучанию не становился близок к произнесенному диктором звуку. Близость оценивалась самим диктором "на слух". В среднем для каждого звука потребовалось 2...3 итерации. Среднее время настройки всей фонетической базы данных в расчете на одного диктора составило 2,75 мин (минимальное время настройки — 1,5 мин, максимальное — 5,1 мин).

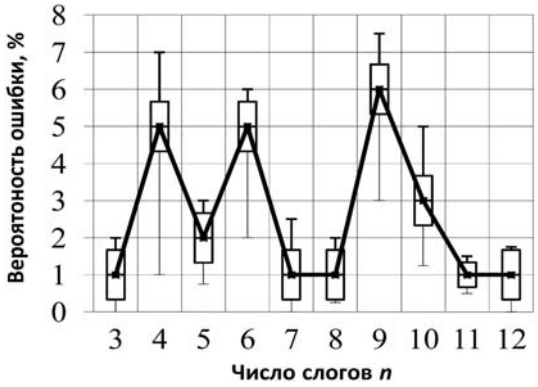


Рис. 2. Зависимость вероятности ошибки для предложенной системы от числа слогов *n* во входном словосочетании

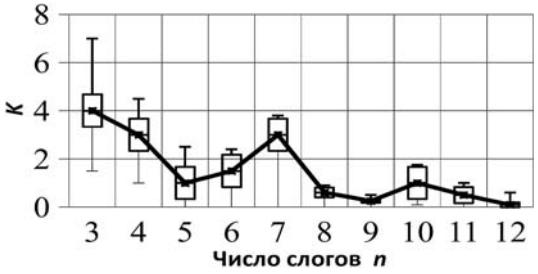


Рис. 3. Зависимость числа проверяемых эталонов *K* для предложенной системы (рис. 1) от числа слогов *n* во входном словосочетании

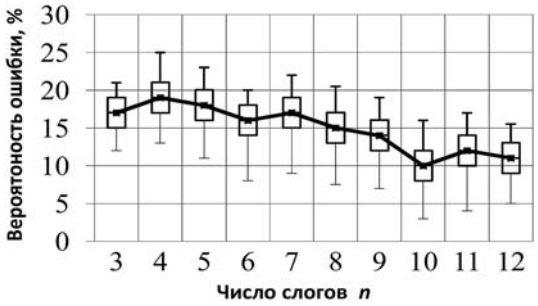


Рис. 4. Зависимость вероятности ошибки для Google Voice от числа слогов *n* во входном словосочетании

Для тестирования качества распознавания слов каждым диктором были произнесены по 10 реализаций каждого слова из словаря эталонов, а также по 10 реализаций слов "да" и "нет". Основным требованием к произнесению команд было разделение слов на открытые слоги с четкой паузой между слогами. В процессе распознавания слоги выделялись простейшим амплитудным детектором паузы, определенной как сигнал с малой амплитудой длительностью не менее 70 мс. Далее каждый выделенный слог членился на последовательность пересекающихся сегментов длительностью $\tau = 10$ мс (80 отсчетов). Сегменты сопоставлялись (2), (3) со всеми гласными звуками из фонетической базы данных этого диктора, после чего выбирался наиболее близкий к сегменту (1) эталонный звук. При распознавании выбирались наиболее близкие (в смысле (5)) команды-эталоны (с максимальным μ_l). Качество распознавания оценивалось по двум показателям:

- вероятности ошибки (отсутствии произнесенной команды в списке альтернатив);
- среднему числу *K* альтернатив, которые после упорядочивания находятся ближе к входному слову, чем произнесенная команда.

Результаты в виде зависимости усредненных по дикторам вероятности ошибки и числа проверяемых эталонов *K* от числа слогов *n* во входном словосочетании показаны на рис. 2, 3 в виде диаграмм типа "ящик с усами". Заметим, что большая часть ошибок при распознавании связана с неверным определением числа слогов во входной команде из-за присутствия постороннего шума или из-за малой паузы между слогами (менее 70 мс). При распознавании слов "да" и "нет" вероятность ошибки составила 0,75 %, причем *все* ошибки возникли при неверном выделении слогов.

Для сравнения, на рис. 4 показана вероятность ошибки для распознавания тех же словосочетаний в русскоязычной версии системы Google Voice Search [17]. Система выдает только наиболее близкое словосочетание, поэтому оценивать величину *K* здесь не нужно. Здесь качество дикторонезависимого распознавания из большого словаря очень высоко. Тем не менее, в среднем вероятность ошибки на 10 % выше, чем для предложенного подхода.

Кроме того, некоторые словосочетания отсутствуют в словаре системы, поэтому они остались нераспознанными.

По результатам проведенного экспериментального исследования можно сделать следующие выводы. Во-первых, точность классификации для предложенной системы (см. рис. 2) выше, чем для канонического подхода (рис. 4). Во-вторых, качество предложенной системы можно повысить, если использовать более совершенные алгоритмы выделения открытых слогов. И, в-третьих, можно сделать главный вывод — предлагаемая система (см. рис. 1), основанная на требовании четкого слогового произношения, позволяет построить надежную систему голосового управления, в которой преодолены описанные ранее проблемы канонического подхода.

Заключение

Задача построения надежных систем голосового управления [2] является одной из центральных во всем направлении АРР. К сожалению, использование здесь канонического подхода к распознаванию естественной речи, применяемому в системах диктовки текста, не всегда позволяет получить удовлетворительное качество распознавания, особенно, если требуется за короткое время перенастраивать систему на новое множество команд. При этом основной способ повышения надежности [11] — настройка под конкретного диктора — наталкивается на сложности, связанные с длительностью этой процедуры.

В настоящей работе показано, что если искусственно ввести требование к пользователю по четкому слоговому произношению команд [13], то можно построить достаточно надежную вычислительно эффективную систему с автоматически перенастраиваемым словарем и с увеличением качества распознавания в процессе функционирования. Отметим, что предложенный подход может применяться не только в голосовом управлении, но и в других сферах, требующих быстрой адаптации рабочего словаря, таких как заказы по телефону некоторой продукции, например лекарств. В подобных задачах требование к слоговому произношению является зачастую вполне естественным.

Список литературы

1. **Benesty J., Sondh M., Huang Y.** (eds.). Springer Handbook of Speech Recognition. New York: Springer. 2008. 1159 p.

2. **Tan B.** A Distributed Speech Remote Control System Based on Web Service and Automatic Speech Recognition // Electrical Power Systems and Computers, Lecture Notes in Electrical Engineering. 2011. Vol. 99. P. 771—778.

3. **Рабинер Л. Р.** Скрытые марковские модели и их применение в избранных приложениях при распознавании речи: обзор // ТИИЭР. 1989. № 77 (2).

4. **Kullback S.** Information Theory and statistics. New York: Dover Pub. 1997. 399 p.

5. **Савченко В. В.** Метод фонетического декодирования слов в задаче автоматического распознавания речи на основе принципа минимума информационного рассогласования // Изв. вузов России. Радиоэлектроника. 2009. Вып. 5. С. 31—41.

6. **Белявский В. М., Светозарова Н. Д.** Слоговая фонетика и три фонетики Л. В. Щербы. Теория языка, методы его исследования и преподавания. Л.: Наука, 1981. С. 36—40.

7. **Савченко В. В., Пономарев Д. А.** Оптимизация фонетической базы данных по группе дикторов на основе информационной теории восприятия речи // Информационные технологии. 2009. № 12. С. 7—12.

8. **Марпл С. Л.-мл.** Цифровой спектральный анализ и его приложения. М.: Мир, 1990. 584 с.

10. **Савченко А. В.** Метод направленного перебора словаря в задаче автоматического распознавания речи на основе принципа минимума информационного рассогласования // Системы управления и информационные технологии. 2009. № 35 (1). С. 83—91.

11. **Anusuya M. A., Katti S. K.** Speech recognition by Machine // A Review, International Journal of Computer Science and Information Security. 2009. 6 (3).

12. **Бабин Д. Н., Мазуренко И. Л., Холоденко А. Б.** О перспективах создания системы автоматического распознавания слитной устной русской речи // Интеллектуальные системы. Т. 8. 2004. Вып. (1-4). С. 45—70.

13. **Патент РФ № 2011125526/08 21.06.2011.** Савченко А. В., Савченко В. В., Акатьев Д. Ю. Устройство для фонетического анализа и распознавания речи. Патент России на полезную модель № 111944, 2011, Бюл. № 36.

14. **Sirigos J., Fakotakis N., Kokkinakis G.** A hybrid syllable recognition system based on vowel spotting // Speech Communication. 2002. Vol. 38. P. 427—440.

15. **Кипяткова И. С., Карпов А. А.** Разработка и оценивание модуля транскрибирования для распознавания и синтеза русской речи // Искусственный интеллект. 2009. 3. С. 178—185.

16. **Janakiraman R., Kumar J. C., Murthy H. A.** Robust syllable segmentation and its application to syllable-centric continuous speech recognition // In Proc. National Conference on Communications. 2010. P. 1—5.

17. **Schuster M.** Speech Recognition for Mobile Devices at Google // Lecture Notes in Computer Science. 2010. Vol. 6230. P. 8—10.

УДК 519.256

Е. А. Будников, студент,
Московский физико-технический институт,
В. В. Стрижов, канд. физ.-мат. наук, науч. сотр.,
Вычислительный центр РАН,
e-mail: strijov@ccas.ru

Оценивание вероятностей появления строк в коллекции документов[1]

Рассматривается задача оценивания вероятностей появления строк в документах. Для решения задачи использована модель n-грамм. Для решения проблемы большого числа параметров предложено использовать модель n-грамм на классах. Для решения проблемы нулевых вероятностей появления строк применяют три дисконтные модели: Гуда—Тьюринга, Катца и абсолютного дисконтирования. Описан проведенный эксперимент на синтетических данных. Предлагаемая модель проиллюстрирована вычислительным экспериментом на реальных данных.

Ключевые слова: *языковая модель, дисконтная модель, n-граммы на классах, модель Гуда—Тьюринга, модель Катца, абсолютное дисконтирование*

Введение

В задачах, связанных с анализом текста, требуется оценить априорную вероятность появления строк. Для этого используют метод n-грамм [1—4], который заключается в том, что апостеорная вероятность появления слова после некой строки зависит не от всех слов строки, а лишь от последних $n-1$ слов.

Основными недостатками этого метода являются, во-первых, сложность получения оценок большого числа параметров статистической модели и, во-вторых, проблема наличия нулевых оценок вероятности появления слов в строках, которые не встречаются в процессе обучения. Для решения этих проблем предлагается использовать метод n-грамм на классах. Этот метод заключается в том, что все слова языка разбиваются на классы, тем самым снижается число параметров, затем во время обучения настраиваются вероятности появления в языке шаблонов строк, состоящих из названий классов, а также вероятности появления слова в определенном классе [1, 5]. Число строк с нулевой вероятностью уменьшается, однако они остаются.

Для перераспределения вероятностей предлагается использовать различные дисконтные модели [1, 3, 4]. В модели Гуда—Тьюринга [6] все n-граммы разбиты на группы в зависимости от частоты появления, а затем происходит сглаживание этих частот между соседними группами. Этот метод прост в реализации, однако неустойчив. Что означает эта неустойчивость, будет пояснено ниже. Также он сглаживает и оценки вероятностей n-грамм, которые встречаются в обучении достаточно часто и могут быть признаны надежно обученными.

В модели Катца [7] выбирается соответствующий порог, и оценки вероятностей n-грамм, частота появления которых в обучении больше этого порога, не сглаживаются. Однако эта модель также неустойчива.

Модель абсолютного дисконтирования [8] использует другой подход. Из всех ненулевых частот вычитается фиксированное число, которое потом перераспределяется между n-граммами, не встретившимися в обучении. Можно подобрать это число так, чтобы суммарное уменьшение вероятности было таким же, как и в модели Гуда—Тьюринга. В данной работе предложены и реализованы алгоритмы оценивания вероятностей появления строк в коллекции документов. При решении задач оценки параметров статистических моделей с использованием байесовского вывода эти вероятности считаются априорными в контексте языка, на котором написаны документы коллекции.

1. Статистическая модель и преплексия

Пусть $W = \overline{w_1 w_2 \dots w_k}$ — строка из слов w_i из словаря Ω. Обозначим подстроку строки W как $w_i^j = \overline{w_i w_{i+1} \dots w_j}$, где i — позиция первого символа подстроки, а j — позиция последнего. Согласно этому обозначению строка $W \equiv w_1^k$. Вероятность появления строки равна произведению апостеорных вероятностей появления каждого слова этой строки при условии известной предыстории, т. е. подстроки, предшествующей данному слову:

$$Pr(w_1^k) = Pr(w_k | w_1^{k-1}) Pr(w_{k-1} | w_1^{k-2}) \dots \\ \dots Pr(w_2 | w_1) \cdot Pr(w_1).$$

Определение 1. *Моделью естественного языка назовем семейство функций*

$$f : \mathsf{R}^P \times \mathsf{R}^N \to \mathsf{R}^k,$$

где R^P — пространство параметров; R^N — пространство исходных строк; R^k — пространство зависимых переменных.

Определение 2. *Статистической моделью естественного языка называется семейство функций*

$$f : \mathsf{R}^P \times \Omega^* \to [0, 1],$$

[1] Работа выполнена при поддержке Министерства образования и науки РФ в рамках Государственного контракта 07.524.11.4002.

где R^P — пространство параметров; Ω^* — пространство строк, составленных из слов словаря Ω, а [0, 1] — интервал оценки вероятности появления строки в языке.

Качество модели оценивается по тестовым строкам текста значением перплексии.

Определение 3. *Перплексией называется величина, обратная к величине средней вероятности, приписываемой каждому слову строки*

$$\mathrm{PP} = \frac{1}{\sqrt[k]{Pr(w_1 w_2 \ldots w_k)}}.$$

Чем больше в среднем число слов, которые могут идти после заданного предыдущего слова, тем больше перплексия модели.

2. Модель *n*-грамм

При отсутствии предположений о длине предыстории и о вероятности $Pr(w_k|w_1^{k-1})$ число параметров будет равно числу всевозможных строк языка и будет бесконечно растущим с ростом длины строки. В методе n-грамм две предыстории считаются одинаковыми, если они оканчиваются на одинаковые $n-1$ слов.

Определение 4. *Модель естественного языка называется моделью на n-граммах, если для параметров модели выполнено условие*

$$Pr(w_k|w_1^{k-1}) = Pr(w_k|w_{k-n+1}^{k-1}). \qquad (1)$$

Пример 1. *Статистическая модель биграмм задает следующее семейство функций:*

$$f = Pr(w_1 w_2 \ldots w_n) = Pr(w_n|w_{n-1}) \cdot Pr(w_{n-1}|w_{n-2}) \ldots \\ \ldots Pr(w_2|w_1) \cdot Pr(w_1).$$

Число параметров модели (1) определено следующим утверждением.

Утверждение 1. *Если словарь содержит V слов, то модель n-грамм содержит* $V^n - 1$ *параметров.*

Если словарь содержит V слов, то униграммы порождают модель, имеющую $V-1$ независимых параметров: V параметров $Pr(\tilde{w}_i)$ связаны равенством

$$\sum_{i=1}^{V} Pr(\tilde{w}_i) = 1,$$

где $\tilde{w}_i$ — слова из словаря. Биграммы порождают $V^2 - 1$ независимых параметров: $V(V-1)$, имеющих форму $Pr(w_2|w_1)$, и $V-1$, имеющих форму $Pr(w)$. Далее по индукции легко показать, что модель n-грамм содержит $V^n - 1$ параметров.

Оценка параметров модели выполняется по коллекции текстов T. Пусть $C(w)$ — число раз, которые строка w встретилась в обучающем тексте. Тогда в случае униграмм максимум правдоподобия для параметра $Pr(w)$ достигается при $Pr(w) = \frac{C(w)}{T}$. Действительно, $V^{n-1}(V-1)$ параметров имеют форму $Pr(w_n|w_1^{n-1})$ и $V^{n-1} - 1$ параметров более низкого порядка (по предположению индукции). Всего

$$V^{n-1}(V-1) + V^{n-1} - 1 = V^n - 1.$$

Для случая n-грамм максимум правдоподобия равен

$$Pr(w_n|w_1^{n-1}) = \frac{C(w_1^{n-1} w_n)}{\sum_w C(w_1^{n-1} w)}.$$

3. Модель *n*-грамм на классах

Пусть существует некоторая функция $\pi:\Omega \to G$, где Ω — множество слов, словарь, а G — множество классов слов. Тогда обозначим $Pr(w|g)$ вероятность появления в языке слова w, если известен его класс g, а $Pr(g_n|g_1^{n-1})$ — вероятность встретить слово из класса g_n после последовательности слов, имеющих форму $g_1 g_2 \ldots g_{n-1}$. Оценим только параметры вида $Pr(g_n|g_1^{n-1})$ и $Pr(w|g)$.

Определение 5. *Модель n-грамм назовем моделью n-грамм на классах, если выполняется гипотеза*

$$Pr(w_k|w_1^{k-1}) = Pr(w_k|g) Pr(g_k|g_1^{k-1}), \text{ где } k = 1, \ldots, n. \qquad (2)$$

Пример 2. *Статистическая модель биграмм на классах задает следующее семейство функций:*

$$f = Pr(w_1 w_2 \ldots w_n) = Pr(w_n|g_n) \cdot Pr(g_n|g_{n-1}) \ldots \\ \ldots Pr(w_2|g_2) \cdot Pr(g_2|g_1) \cdot Pr(w_1|g_1) \cdot Pr(g_1).$$

Число параметров модели (2) определено следующим утверждением.

Утверждение 2. *Если словарь содержит V слов и имеется C классов, то модель n-грамм на классах содержит* $C^n + V - C - 1$ *параметров.*

Действительно, имеется $C^n - 1$ параметров вида $Pr(g_n|g_1^{n-1})$, что доказывается аналогично Утверждению 1, и $V-C$ параметров вида $Pr(w|g)$, так как всего таких вероятностей V ($Pr(w_i|g_i)$, $i \in \{1, \ldots, V\}$), но для каждого класса $g \in G$ выполняется равенство:

$$\sum_{w:\pi(w)=g} Pr(w|g) = 1.$$

Опишем алгоритм построения функции π на примере биграмм. Пусть $T = (t_1, t_2, \ldots, t_T)$ — текст,

причем все слова содержатся в словаре Ω. Функция правдоподобия данного текста тогда равна

$$L(T) = Pr(T) = \prod_{x,\, y \in \Omega} Pr(y|x)^{C(xy)},$$

где x, y — слова из словаря, а $C(xy)$ показывает, сколько раз последовательность слов "xy" встретилась в обучающей выборке T. Решается задача максимизации

$$L(T) \to \max. \qquad (3)$$

Утверждение 3. *Задача максимизации (3) равносильна максимизации функции*

$$F_\pi = \sum_{g,\, h \in G} C(gh) \cdot \log C(gh) - 2 \sum_{h \in G} C(h) \cdot \log C(h),$$

где $C(gh)$ — функция, которая показывает, сколько раз в обучающем тексте встретились строки вида "xy", где $\pi(x) = g$, а $\pi(y) = h$.

Для удобства будем использовать логарифм функции правдоподобия вместо самой функции:

$$\log L(T) = \sum_{x,\, y \in \Omega} C(xy) \cdot \log Pr(y|x).$$

Из данного выше определения модели n-грамм на классах заключаем, что максимум правдоподобия для биграмм достигается при

$$Pr(w_i|w_{i-1}) = \frac{C(w_i)}{C(\pi(w_i))} \cdot \frac{C(\pi(w_{i-1})\pi(w_i))}{C(\pi(w_{i-1}))},$$

где $C(w_i)$ — число раз, когда слово w_i встретилось в обучающей выборке, а $C(\pi(w))$ — число раз, когда слова из класса $\pi(w)$ встретились в выборке, аналогично $C(\pi(w_x)\pi(w_y))$ — число пар вида "$\pi(w_x)\pi(w_y)$", встретившихся в выборке. Подставим теперь это выражение в функцию правдоподобия и преобразуем:

$$\log L(T) =$$

$$= \sum_{x,\, y \in \Omega} C(xy) \cdot \log\left(\frac{C(y)}{C(\pi(y))} \cdot \frac{C(\pi(x)\pi(y))}{C(\pi(x))}\right) =$$

$$= \sum_{x,\, y \in \Omega} C(xy) \cdot \log\left(\frac{C(y)}{C(\pi(y))}\right) +$$

$$+ \sum_{x,\, y \in \Omega} C(xy) \cdot \log\left(\frac{C(\pi(x)\pi(y))}{C(\pi(x))}\right) =$$

$$= \sum_{y \in \Omega} C(y) \cdot \log\left(\frac{C(y)}{C(\pi(y))}\right) + \sum_{g,\, h \in G} C(gh) \cdot \log\left(\frac{C(gh)}{C(g)}\right) =$$

$$= \sum_{y \in \Omega} C(y) \cdot \log C(y) - \sum_{y \in \Omega} C(y) \cdot \log C(\pi(y)) +$$

$$+ \sum_{g,\, h \in G} C(gh) \cdot \log C(gh) - \sum_{g,\, h \in G} C(gh) \cdot \log C(g) =$$

$$= \sum_{y \in \Omega} C(y) \cdot \log C(y) + \sum_{g,\, h \in G} C(gh) \cdot \log C(gh) -$$

$$- 2 \sum_{h \in G} C(h) \cdot \log C(h).$$

Заметим, что первое слагаемое не зависит от выбора функции π и значит, его рассматривать необязательно, когда мы будем оптимизировать π. Поэтому будем максимизировать функцию

$$F_\pi = \sum_{g,\, h \in G} C(gh) \cdot \log C(gh) - 2 \sum_{h \in G} C(h) \cdot \log C(h).$$

Приведем теперь алгоритм построения функции π. Перед запуском алгоритма определяется число классов.

1. Для всех слов $w \in \Omega$ из словаря $G(w) = 1$ инициализировать набор классов.

2. Для всех начал слов $i = 1...n$ и всех классов $c \in G$ повторять следующие шаги, пока F_π не перестанет увеличиваться.

3. Переместить слово w в класс c, запомнив его предыдущий класс.

4. Вычислить изменения F_π для этого перемещения в c. Переместить слово w назад в его предыдущий класс.

5. Переместить слово w в класс, который больше всего увеличивает F_π, или никуда не перемещать, если увеличения ни на каком перемещении не происходит.

Вышеописанный алгоритм сходится к локальному максимуму F_π. Это утверждение следует из того, что на каждом шаге значение F_π увеличивается.

4. Дисконтная модель

Рассмотрим событие S, которое встретилось s раз при общем числе наблюдений A. Тогда оценка вероятности S по принципу наибольшего правдоподобия будет

$$Pr(S) = \frac{s}{A}.$$

Но в соответствии с этим принципом событиям, которые не были встречены среди обучающего текста T, будут приписаны нулевые вероятности, а значит, будучи встреченными на тесте, они никогда не будут распознаны. Чтобы решить проблему предлагается в оценке вероятности события вместо числа s брать

$$s' = d_s s,$$

где d_s — множитель, зависящий от числа раз, которое событие встретилось в обучающем тексте. Тогда получим дисконтную оценку вероятности события S:

$$Pr_{\text{discount}}(S) = \frac{s'}{A} = \frac{d_s s}{A}.$$

Рассмотрим различные дисконтные методы, которые различаются стратегией выбора d_s. Обозначим c_s, число всех событий, которые встретились в процессе обучения ровно s раз. Тогда общее число

наблюдений $A = \sum_{s \geqslant 1} c_s s$. Таким образом, мы перераспределили оценки вероятности между событиями, вероятность всех не встретившихся в обучении слов равна $1 - \frac{1}{A} \sum_{s \geqslant 1} d_s c_s s$. Если c_0 — число таких событий, то оценка вероятности каждого из них равна

$$\frac{1}{c_0}\left(1 - \frac{1}{A} \sum_{s \geqslant 1} d_s c_s s\right).$$

Сравним описанную модель с моделями, предложенными в работах [6, 7, 8].

5. Дисконтная модель Гуда—Тьюринга

В работе [6] предлагается следующая стратегия выбора множителя:

$$d_s = (s + 1)\frac{c_{s+1}}{s \cdot c_s}.$$

Эта стратегия называется оценкой Гуда—Тьюринга. Несмотря на очевидную простоту стратегии у нее есть существенный недостаток: она проваливается в случае, если $c_a = 0$ для некоторого a и существует $b > a$, такой, что $c_b \neq 0$. Также существенно, что дисконтирование необходимо для параметров, оценка которых является ненадежной, т. е. для тех событий, которые встречаются в обучении менее некоторого числа раз k, выбранного априори.

6. Дисконтная модель Катца

Решение проблемы вычисления значения множителя было предложено в работе [7]. Пусть есть некое, достаточно большое число k, такое, что все оценки вероятностей событий, встретившихся в процессе обучения более k раз, признаем надежными. При этом d_s будет выглядеть так:

$$d_s = \begin{cases} \dfrac{(s+1)\dfrac{c_{s+1}}{s \cdot c_s} - (k+1)\dfrac{c_{k+1}}{c_1}}{1 - (k+1)\dfrac{c_{k+1}}{c_1}}, & 1 \leqslant s \leqslant k, \\ 1, & s > k. \end{cases}$$

Этот метод тоже нестабильный, так как возможны ситуации, когда $d_s < 0$.

7. Модель абсолютного дисконтирования

Одной из альтернатив модели Гуда—Тьюринга является модель абсолютного дисконтирования [8]. В этой модели происходит уменьшение числа a для каждого события на фиксированное число m согласно формуле

$$d_s = \frac{s - m}{s}.$$

Для того чтобы уменьшение суммарной вероятности было таким же, как в модели Гуда—Тьюринга, необходимо, чтобы выполнялось равенство

$$m = \frac{c_1}{\sum_{s \geqslant 1} c_s}.$$

8. Вычислительный эксперимент

Для проведения вычислительного эксперимента был использован корпус данных ISABASE для обучения статистических моделей. Число n-грамм в исходных данных представлено в табл. 1. Корпус разбит на обучающую и тестовую выборку.

Для теста использовались данные из того же корпуса. Можно обратить внимание, что несмотря на большой размер обучающей выборки, число новых слов довольно велико, а почти все биаграммы и триграммы новые.

Число невидимых слов, которые мы ожидаем увидеть — внешний параметр модели. Настраивался он методом скользящего контроля. Получили четко выраженный минимум при 11 500 словах. Это значение и было использовано для подсчета перплексии тестовой выборки.

В табл. 2 показаны результаты вычисления перплекции для вышеописанных моделей. Вычислительный эксперимент показал, что модель без дисконтирования не дала оценку перплексии по очевидной причине присутствия в тестовой выборке новых

Таблица 1

Число n-грамм в обучающей и тестовой выборках

Выборка	Число		
	униграмм ($n = 1$)	биграмм ($n = 2$)	триграмм ($n = 3$)
Обучающая	199 260	1 576 969	2 462 694
Тестовая (все элементы)	4901	6942	6215
Тестовая (новые элементы)	489	4063	5549

Таблица 2

Перплексия текстовой выборки

Модель дисконтирования	Перплексия	
	n-граммы	n-граммы на классах
Без дисконтирования	∞	∞
Модель Гуда—Тьюринга	∞	∞
Модель Катца	9560	7420
Модель абсолютного дисконтирования	10 070	8210

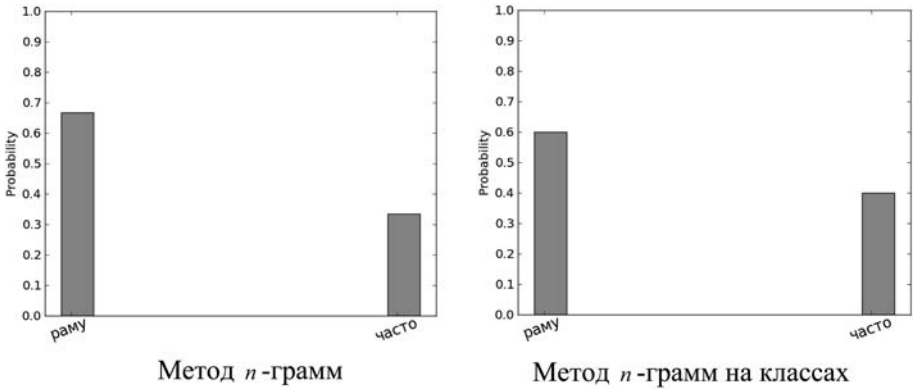


Рис. 1. Методы без дисконтирования

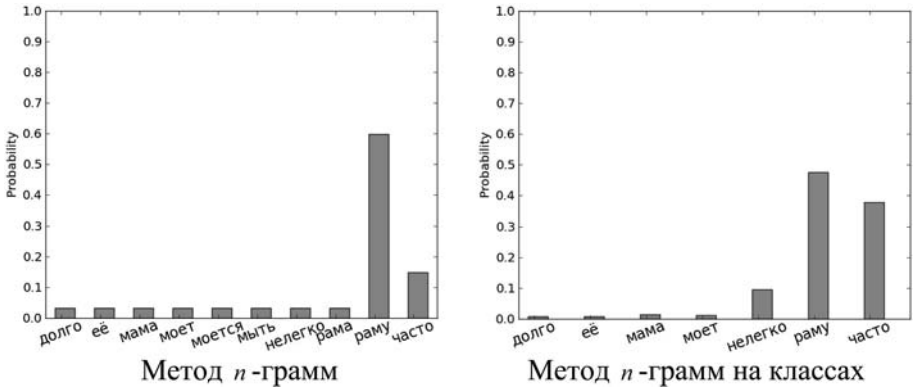


Рис. 2. Модель дисконтирования Гуда—Тьюринга

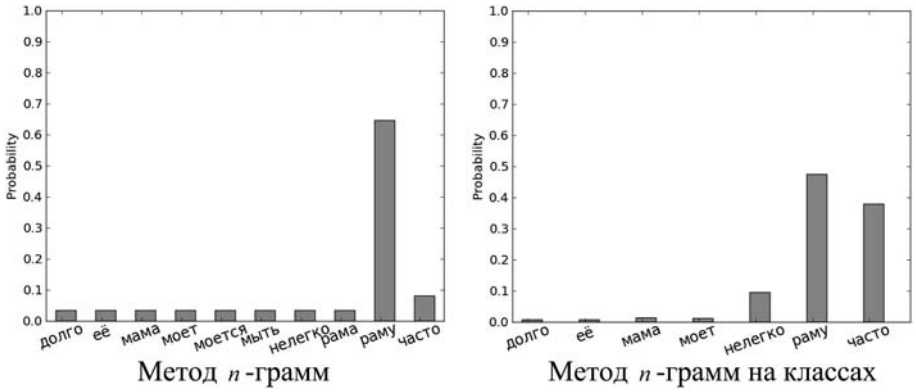


Рис. 3. Модель дисконтирования Катца

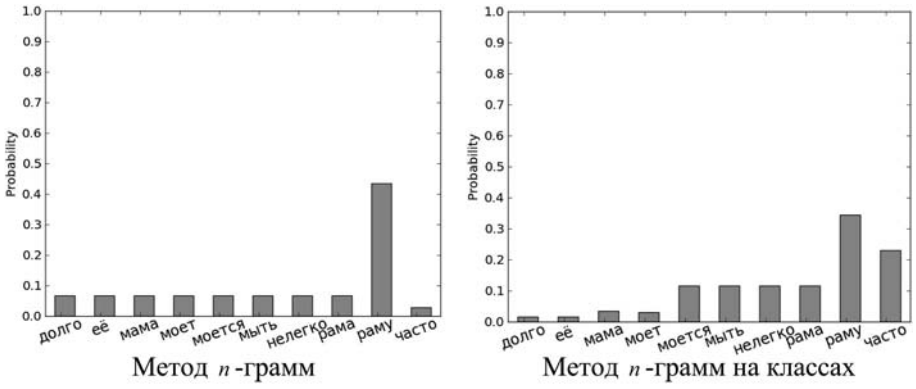


Рис. 4. Модель абсолютного дисконтирования

слов по сравнению с обучающей, а модель с дисконтированием Гуда—Тьюринга не дала оценку перплексии, потому что среди n-грамм был разрыв в частотах, а в процессе теста встретились слова, которые были оценены нулевой вероятностью. Модель Матца и модель абсолютного дисконтирования дали сравнимые по значению оценки перплексии.

Также в вычислительном эксперименте демонстрировалась работа комбинаций алгоритмов на небольшом массиве синтетических данных, состоящих из текста на тему "Мама моет раму". В первой серии экспериментов оценивалось распределение вероятностей появления слова после заданной строки текста. Во второй серии экспериментов оценивалась перплексия различных тестовых строк: встречающейся в обучающем тексте и двух, не встречающихся в тексте. В обеих сериях для методов n-грамм и n-грамм на классах проводили по четыре типа экспериментов: без дисконтирования и по каждому из трех типов дисконтирования.

В первой серии экспериментов оценивали распределение вероятностей появления слова после фразы "Мама моет...". На рис. 1—4 столбики гистограммы показывают оценку вероятности появления слова из корпуса после этой фразы.

Оба метода без дисконтирования (см. рис. 1) дали примерно одинаковые результаты, распределив лишь немного иначе вероятности между двумя вариантами продолжения. Метод n-грамм на классах немного сгладил разницу между вероятностями. Это связано с тем, что описанный в работе алгоритм определил слова "раму" и "часто" в один класс, а вероятности между этими словами распределяются в зависимости от суммарной частоты появления в обучающем тексте, а не только после строки "Мама моет...", а точнее, шаблона строки

"g_1g_2...", где g_1 — класс слова "мама", а g_2 — класс слова "моет".

На рис. 2 продемонстрирован метод дисконтирования Гуда—Тьюринга. В графики были включены лишь варианты с вероятностями > 0,004. Оценка вероятности слова "часто" в методе *n*-грамм снизилась. Это связано с тем, что метод дисконтирования предполагает, что оценка вероятности появления события, встретившегося однажды или дважды в процессе обучения, не должна существенно отличаться от оценки вероятности появления события, в процессе обучения не встретившегося.

На рис. 3 показано, что в модели дисконтирования Катца оценки параметров не сглаживаются: первые столбики гистограмм, представленные методом *n*-грамм имеют равные вероятности.

Рис. 4. представляет результаты модели абсолютного дисконтирования. В этой модели при использовании метода *n*-грамм оценка вероятности события, встречавшегося в обучении, в итоге оказывается ниже оценки вероятности события, которое в обучении не встретилось. Это объясняется высокой долей биграмм, которые встретились в обучении только один раз, среди всех встретившихся в обучении биграмм.

Во второй серии экспериментов оценивалась перплексия различных тестовых строк.

В табл. 3 и 4 показано, что метод *n*-грамм на классах без дисконтирования и метод *n*-грамм на классах с дисконтированием Катца являются предпочтительными. Они надежно оценивают вероятности строк и имеют минимальные перплексии.

Таблица 3

Перплексия подстроки из обучающего текста "Мама моет часто"

Модель дисконтирования	Перплексия	
	n-граммы	*n*-граммы на классах
Без дисконтирования	2,5	2,06186
Модель Гуда—Тьюринга	5,62562	3,82051
Модель Катца	9,66017	1,9987
Модель абсолютного дисконтирования	3,00793	2,71695

Таблица 4

Перплексия подстроки "Мама моет долго", которая не встречается в обучающем тексте

Модель дисконтирования	Перплексия	
	n-граммы	*n*-граммы на классах
Без дисконтирования	∞	6,12372
Модель Гуда—Тьюринга	12,2359	25,5717
Модель Катца	14,8572	13,3778
Модель абсолютного дисконтирования	8,53946	18,0222

Заключение

В работе были рассмотрены методы оценивания вероятностей появления строк в языке, основанные на *n*-граммах. Каждый из рассмотренных методов имеет, как показал вычислительный эксперимент, свои достоинства и недостатки. К достоинствам метода *n*-грамм без дисконтирования можно отнести линейную по размеру обучающего текста сложность алгоритма настройки параметров, к недостаткам — большое число параметров и, как следствие, плохую их обучаемость, а также нулевую оценку вероятности появления в языке *n*-грамм, которые не встретились в процессе обучения.

К достоинствам метода *n*-грамм на классах можно отнести, что число параметров линейно по размеру словаря и квадратично по числу классов, а также локальную оптимальность решения задачи разбиения слов на классы. Недостатками являются высокая вычислительная сложность алгоритма, а также наличие нулевых оценок вероятностей, хоть и на меньшем числе строк в сравнении с методом *n*-грамм.

Дисконтные модели решают проблему нулевых оценок вероятностей появления строки в документе, однако они могут работать неадекватно, если велика доля ненадежно обученных параметров. Также недостатком моделей Гуда—Тьюринга и Катца является их неустойчивость.

Список литературы

1. **Huang X., Acero A., Hon H.** Spoken Language Processing, A Guide to Theory, Algorithm, and System Development. Prentice Hall PTR, 2001.
2. **Jelinek F.** Statistical Methods for Speech Recognition. Cambridge: MIT Press, 1997.
3. **Gotoh Y., Renals S.** Statistical language modelling // Lecture Notes in Computer Science. Berlin: Springer, 2000. Vol. 2705. P. 78—105.
4. **Young S., Bloothooft G.** Corpus-Based Methods in Language and Speech Processing. Dordrecht: Kluwer Academic Publishers, 1997.
5. **Brown P. F., Delia Pietra V. J., deSouza P. V., Mercer R. L.** Class-based *n*-gram models of natural language // Proc. to the IBM Natural Language. Paris, 1990. P. 283—298.
6. **Good I. J.** The population frequencies of species and the estimation of population parameters // Biometrika. 1953. Vol. 40 (3 and 4). P. 237—264.
7. **Katz S. M.** Estimation of probabilities from sparse data for the language model component of a speech recognizer // IEEE Transactions on Acoustics, Speech and Signal Processing, 1987. Vol. 35 (3). P. 400—401.
8. **Ney H., Essen U., Kneser R.** On structuring probabilistic dependencies in stochastic language modelling // Computer Speech and Language, 1994. Vol. 8(1). P. 38.

УДК 004.7

В. В. Наумова, д-р геол.-мин. наук, зав. лаб.,
e-mail: naumova@fegi.ru,
Федеральное государственное
бюджетное учреждение науки
Дальневосточный геологический институт
Дальневосточного отделения
Российской академии наук, г. Владивосток

Виртуальные научные среды для обеспечения совместной работы территориально распределенных научных сотрудников

Обсуждаются вопросы построения виртуальных научных лабораторий и виртуальных научных сред. Приводится авторская классификация различных подходов и технологических решений. Описывается постановка задачи, проектирование и реализация тестовой версии виртуальной научной среды для обеспечения совместной работы территориально распределенных научных сотрудников.

Ключевые слова: *современные информационные технологии для научных исследований, виртуальные научные среды, виртуальные лаборатории*

В структуру Российской академии наук входят научные организации, расположенные в центральной части Российской Федерации, на Урале, в Сибири и на Дальнем Востоке. Территориальная разобщенность институтов РАН ставит задачу информационной интеграции территориально разрозненных институтов, научных групп, сотрудников между собой для совместной работы над научными проектами, обсуждений полученных результатов, удаленного участия в научных конференциях и др.

Не менее важной является задача информационной интеграции университетов России и институтов Российской академии наук.

Используя современные информационные технологии, можно решать задачи информационной интеграции территориально распределенных научных сотрудников.

Разрабатываемые в настоящее время решения можно классифицировать следующим образом.

1. Виртуальные научные лаборатории:
- моделирование научно-прикладных исследований и экспериментов с использованием средств виртуальной реальности;
- тематические виртуальные лаборатории;
- удаленный доступ к научному оборудованию.

2. Виртуальные научные среды:
- научные информационные среды Интернет (электронные библиотеки, инфраструктуры данных, центры данных и т. д.);
- информационно-вычислительные научные инфраструктуры;
- системы видеоконференц-связи;
- единые среды сбора научных данных и сервисов их обработки;
- корпоративные научные облака.

Виртуальные научные лаборатории

Назовем виртуальной научной лабораторией аппаратно-программный комплекс, обеспечивающий совместную работу территориально распределенных научных сотрудников, выполняющих традиционные для академической среды исследования в виртуальной среде.

Моделирование научно-прикладных исследований и экспериментов с использованием средств виртуальной реальности. Успешная визуализация и имитирование реальной среды взаимодействия человека и техники посредством компьютера разработана Национальным аэрокосмическим агентством США (NASA) еще 20 лет назад. Целью этой технологии являлась проверка работы техники и поведения человека при работе в сложных и опасных условиях космоса и, таким образом, оценка и улучшение космических проектов. Долгое время весьма высокая стоимость аппаратно-программных комплексов, позволяющих осуществить подобную визуализацию, ограничивала их применение только военными проектами и космической промышленностью. Однако прогресс и удешевление этих технологий за последние годы позволили внести концепцию виртуальной реальности и виртуального прототипирования во многие отрасли науки промышленности и бизнеса.

В настоящее время мы наблюдаем все более массированное применение технологий виртуального прототипирования, предназначенного для последующего производства, ее всесторонней оценки на этапе наличия виртуального прототипа (напри-

мер, безопасности, функциональности, технологичности и т. д.), оптимизации технологических процессов его изготовления. Только после получения удовлетворительных результатов принимается решение об изготовлении физического объекта.

Тематические виртуальные лаборатории. Примеров, подобных лабораторий по физике, химии, механике, информатике, биологии, экологии, архитектуре, достаточно много. Например, в Удмуртском государственном университете разработана виртуальная лаборатория конечно-элементного моделирования [3]. Система организует эффективное взаимодействие разработчиков и заказчиков, обучение пользователей и студентов и обеспечивает их основными научными сервисами для конечно-элементного моделирования на многопроцессорных вычислительных системах с широким применением web-технологий. Виртуальная лаборатория — это не просто интегрированная система для математического моделирования, объединяющая открытые научные пакеты прикладных программ, компоненты CAD-систем, удаленные вычислительные кластеры, но и основа совместной разработки, сопровождения и использования прикладного программного обеспечения.

Удаленный доступ к научному оборудованию. Комплексные системы удаленного доступа к оборудованию наноиндустрии созданы в рамках федеральной целевой программы "Развитие инфраструктуры наноиндустрии в Российской Федерации на 2008—2011 годы" [11].

Одним из примеров является интерактивный учебно-научный комплекс удаленного доступа к сверхвысоковакуумной системе анализа поверхности Multiprobe MXPS VT AFM с источником осаждения нанокластеров Nanogen-50 с квадрупольным масс-фильтром MesoQ, разработанный сотрудниками Национального исследовательского ядерного университета "МИФИ". Комплекс предназначен для дистанционного обучения и выполнения удаленных научных экспериментов по формированию нанокластеров металлов и исследованию образцов методами сканирующей зондовой микроскопии и рентгеновской фотоэлектронной спектроскопии [10].

Виртуальные научные среды

Научная информационная среда Интернет (электронные библиотеки, инфраструктуры данных, центры данных и т. д.). Многолетние исследования ученых из институтов РАН позволили собрать огромную научную информацию. В институтах полученные данные систематизируются. Создаются архивы и базы данных, ГИС, информационно-поисковые системы, электронные библиотеки. Новая цифровая и электронная среда существования данных создает условия для использования современных информационных технологий.

Результатом этой деятельности является появление разнородных документов и информационных систем.

При этом ресурсы могут храниться в различных базах данных и цифровых репозиториях и обрабатываться различными СУБД. Некоторые данные в разных подсистемах могут быть связаны между собой как на уровне взаимных ссылок, так и на уровне наличия общего контента. Немаловажной характеристикой всей системы, влияющей на доступность ресурсов для мирового сообщества, является возможность доступа к данным внешних поисковых систем (Google, Yandex и т. д.).

В этой области нельзя не отметить мировую тенденцию интеграции гетерогенных информационных систем и ресурсов, направленной на формирование единого виртуального информационного пространства, которое в конечном счете и выступает как распределенная система с характерными особенностями (иерархичность подсистем, разнородность ресурсов и программно-аппаратных сред, распределенность элементов инфраструктуры в среде единого сетевого пространства).

В Сибирском отделении РАН работы по интеграции библиографической информации начались в конце 90-х годов прошлого века. В результате проведенных работ были созданы основы для интеграции разнородных данных в виде предварительных моделей и прототипов информационных систем, обеспечивающих работу с географическим аспектом информации в "негеографических" массивах данных [2].

Информационно-вычислительные научные инфраструктуры. В Российской академии наук создана базовая информационно-вычислительная телекоммуникационная структура, позволяющая перейти на новый уровень разработки и внедрения информационных технологий в фундаментальные научные исследования.

Самым большим фрагментом этой структуры является информационно-вычислительная инфраструктура Новосибирский государственный университет (НГУ) — Сибирское отделение РАН (СО РАН), созданная в рамках научно-образовательных центров, которая объединяет вычислительные кластеры НГУ производительностью более 13 трлн операций в секунду, Института вычислительной математики и геофизики СО РАН (ИВМиМГ) производительностью более 4,8 трлн операций в секунду, блейд-системы НГУ, Института вычислительных технологий СО РАН (ИВТ), Института цитологии и генетики СО РАН (ИЦиГ) (базовых институтов для НГУ) для организации облачных вычислений и сетевые системы хранения данных (суммарной емкостью около 300 Тбайт) (устное сообщение А. М. Федотова).

Системы видеоконференц-связи. Существующие в настоящее время системы видеоконференц-связи

Российской академии наук описаны в работах В. В. Наумовой [5—7].

Все созданные и создаваемые сегодня в Российской академии наук системы видеоконференц-связи в основном предназначены для использования в целях оптимизации управления. Поэтому в этих системах терминальные устройства видеоконференц-связи располагаются в конференц-залах Президиумов РАН, Президиумов региональных научных центров, Президиумов научных центров, в кабинетах руководителей различных уровней. Однако современные технологии видеоконференц-связи позволяют использовать видеоконференц-связь не только для целей оптимизации управления, но и для решения научных и научно-организационных задач.

Единые среды сбора научных данных и сервисов их обработки. Одной из самых современных систем этого направления является океанологическая система Дальневосточного отделения РАН (ДВО РАН). Специалистами Тихоокеанского океанологического института ДВО РАН и Института автоматики и процессов управления ДВО РАН в 2008 г. начато развертывание системы оперативного наблюдения за состоянием побережья и акваторий залива Петра Великого [9]. Данные с различных экспериментальных установок, расположенных на побережье и акваториях, доставляются по радиоканалам телекоммуникационной сети залива во Владивосток, обрабатываются (в том числе и на высокопроизводительных вычислительных комплексах ДВО РАН) и предоставляются заинтересованным научным специалистам. К настоящему времени таким образом уже собирается информация с ряда экспериментальных установок, ежедневно в хранилища данных во Владивостоке пересылается 2—4 Гбайт данных удаленных научных экспериментов. Одной из важных компонент системы научного мониторинга залива Петра Великого является система видеонаблюдения. Она в настоящее время включает четыре IP-видеокамеры (1 на о. Попова, 2 на м. Шульца, 1 на о. Большой Пелис) и специализированный оптико-поляризационный комплекс. Фиксируется информация трех видов: моментальные снимки, обзорные панорамы, короткие видеозаписи. Вся видеоинформация сохраняется в базе данных Океанологической информационно-аналитической системы (ОИАС) ДВО РАН во Владивостоке. Зарегистрированным пользователям ОИАС предоставляются удобные средства для поиска и отображения нужной видеоинформации, а также программы, позволяющие извлекать полезную океанологическую информацию из изображений и видеоакваторий.

В качестве примера также можно привести виртуальную обсерваторию [4]. Виртуальная обсерватория интегрирует в единую среду гигантские астрономические архивы и базы данных, распределенные по всему миру, а также инструменты анализа данных и вычислительный сервис, используя при этом набор однородных стандартов и технологий.

Корпоративные научные облака. В Сибирском отделении РАН с 2011 г. начался проект по созданию корпоративного облака [1]. С точки зрения авторов проекта концепция частного облака — предоставление программного обеспечения и инфраструктурных решений как сервиса — соответствует потребностям и особенностям академической среды. Основа технологического решения данного проекта — интегрированные коммуникации Microsoft. Благодаря облаку институты СО РАН получают возможность без установки решений на своей территории, без настройки и последующего администрирования получить средства электронной почты и объединенных коммуникаций. Система Lync поддерживает организацию виртуальных встреч, мгновенные сообщения, совместную работу с документами, проведение аудио- и видеоконференций, интеграцию с традиционной телефонией. Предоставление сервисов Exchange и Lync из облака позволяет использовать эти возможности везде, где есть выход в сеть Интернет.

В данной работе обсуждается постановка, проектирование и реализация тестовой версии виртуальной научной среды для совместной работы территориально распределенных научных сотрудников, выполняющих традиционные для академической среды исследования в рамках совместных исследований.

Разработка Системы нацелена на то, чтобы упростить для конечного пользователя использование в едином комплексе множества различных информационных методов, интегрируя телекоммуникационную среду, современные средства коммуникации, информационно-вычислительные сети, медиасреду, информационные системы, центры данных, тематические приложения, научные сервисы, а также аналитическое оборудование институтов.

Основные задачи создаваемой Системы:

1. Обеспечивать мгновенную передачу сообщений между научными сотрудниками и уведомление о присутствии.

2. Обеспечивать проведение групповых видеоконференций с рабочих мест научных сотрудников.

3. При проведении групповых видеоконференций с рабочих мест научным сотрудникам должны быть доступны следующие сервисы:

- обеспечение трансляции заседаний в Интернет (по запросу);
- обеспечение видеозаписи заседаний (по запросу);
- обеспечение показа презентаций;
- совместная работа с документами:
- удаленный доступ к рабочим столам членов рабочих групп во время работы конференций;
- удаленный доступ к функционалам программ членов рабочих групп во время работы конференций:
- удаленный доступ к сервисам, предоставляемым в сетях РАН: высокопроизводительным вычис-

лениям, информационным системам, центрам хранения данных и др.:

- удаленный доступ к уникальному аналитическому оборудованию РАН.

4. Обеспечивать участие научных сотрудников со своих рабочих мест в многоточечных видеоконференциях, проводимых в конференц-залах, комнатах переговоров и др.

5. Обеспечивать организацию личных кабинетов научных сотрудников и групп.

6. Обеспечивать организацию хранения в личных кабинетах промежуточных и окончательных результатов проектов.

Технологическим фундаментом Системы являются:

- унифицированные коммуникации Microsoft;
- системы видеоконференц-связи Polycom;
- системы научных web-сервисов.

Таким образом, создаваемая нами Система интегрирует современные унифицированные коммуникации с видеоконференц-связью, а также с удаленным доступом к получению данных, системам хранения данных и сервисам их обработки, создавая тем самым интегрированную виртуальную среду для совместной работы территориально распределенных научных сотрудников (рис. 1).

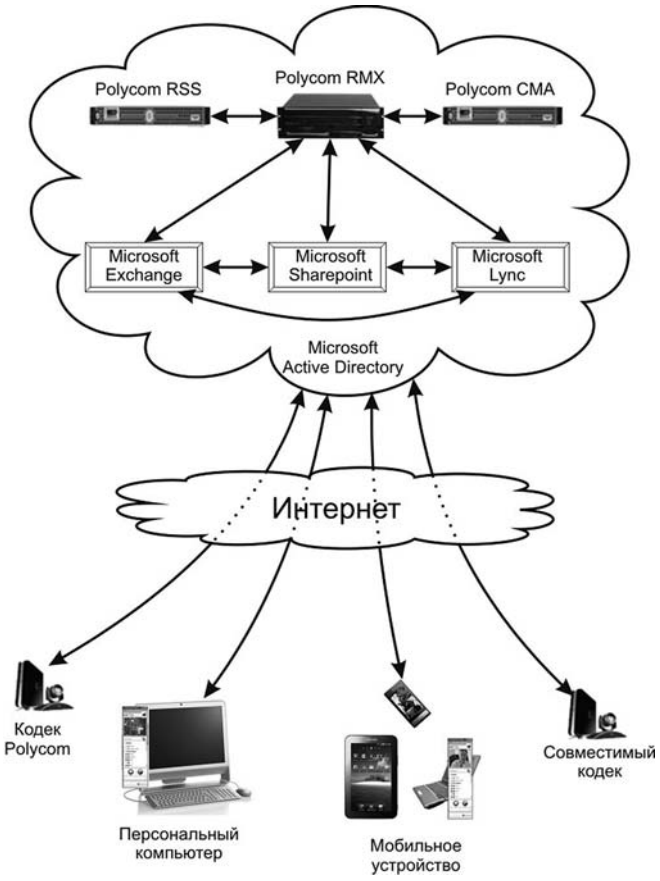


Рис. 1. Общая технологическая схема виртуальной научной среды для обеспечения совместной работы территориально распределенных научных сотрудников

Аппаратно-программные решения

Для создания Системы решено воспользоваться полностью интегрированными решениями Polycom и Microsoft различного уровня: от настольных систем до систем в конференц-залах и за их пределами.

Данное совместное решение объединяет различные платформы объединенных коммуникаций Microsoft: Microsoft Lync, Microsoft Exchange Server, Microsoft Share Point Server и решения Polycom для организации совместной работы с использованием голосовой и видеосвязи HD-качества, с гибкими возможностями масштабирования, обеспечивая качество общения, приближенное к тому, которое возможно при личной встрече, как при взаимодействии сотрудников внутри компаний, так и в среде B2B (рис. 1).

Microsoft и Polycom поставляют стандартные совместимые и интегрированные решения объединенных коммуникаций (UC) в различных конфигурациях и с применением инновационных систем видеоконференц-связи и устройств связи конечных пользователей [8]. Совместными усилиями двух компаний, благодаря использованию систем передачи голоса и видео и масштабируемой базовой инфраструктуры Polycom вместе с решением Microsoft Lync, обеспечивается высокое качество совместной работы. Решение Lync предоставляет функции получения информации о присутствии, мгновенного обмена сообщениями, конференц-связи и голосовой связи внутри организации, доступные через единый интерфейс, являющийся унифицированным как на ПК, так и на мобильном устройстве и в браузере. Только Polycom предлагает совместимые решения, реализующие интеграцию голоса, видео, телеприсутствия, сервисов и приложений во всех продуктах платформы UC Lync. Решения Polycom для голосовой и видеосвязи, способные полностью использовать информацию о присутствии, могут работать на уровне отдельных пользователей или же развертываться как автономные устройства, доступные всем сотрудникам организации.

Для оборудования базового узла Системы мы используем следующие технические решения Polycom:

- сервер видеоконференций — Polycom RMX-2000;
- устройство записи видеоконференций — Polycom RSS 2000;
- решение по организации и управлению видеоконференциями — Polycom CMA 4000.

Сервер видеоконференций — Polycom RMX-2000 способен поддерживать от 20 до 80 видеосоединений (портов) по протоколам SIP или H.323. Платформа для проведения мультимедийных конференций в режиме реального времени, построенная на основе архитектуры Advanced Telecommunications Computing Architecture (AdvancedTCA) и работает под управлением ОС Linux, обеспечивает высокоскоростные соединения, исключительно малое время задержки, высочайшую надежность и удобство об-

служивания. RMX-2000 сконструирована по модульному принципу мультимедийных IP-подсистем (IP Multimedia Subsystem, IMS), отличается высокой степенью масштабируемости и рассчитана на работу с новейшими приложениями для проведения конференций.

Ядром решения видеоконференц-связи является серверная часть Polycom CMA Server, основанное на стандартах приложение управления, обеспечивающее использование крупномасштабного телефонного справочника, централизованное обеспечение и управление для тысяч единиц оконечного оборудования, включая системы класса Telepresence с разрешением высокой четкости и традиционные системы видеоконференц-связи.

Программное обеспечение Polycom CMA Desktop — клиентское приложение для персональных компьютеров, обеспечивающее высококачественную видео и голосовую связь, а также основанный на стандартах совместный доступ к информационным ресурсам (контенту). Простой и дружественный интерфейс CMA Desktop дает возможность корпоративному пользователю начать сеанс видеосвязи с коллегами в любом месте и в любое время, просто выбрав курсором нужный контакт и нажав кнопку мыши. Встроенные средства отслеживания состояния позволяет пользователю удостовериться в присутствии пользователя и возможности соединения, а бесшовная интеграция со службами каталогов по протоколу LDAP упрощает управление и гарантирует актуальность списка контактов.

При создании Системы нами дополнительно развернуты серверные компоненты Microsoft: система электронной почты — Microsoft Exchange, объединенных коммуникаций — Lync Server и портала — Sharepoint. Эти системы используют единый для всех пользователей сервис каталогов Active Directory.

Microsoft Lync Server 2010 превращает средства связи в эффективный инструмент взаимодействия, который значительно упрощает общение и повышает качество совместной работы. Lync 2010 объединяет стандартные средства общения, работающие привычным для пользователей образом, в единую систему коммуникаций.

Основные возможности Microsoft Lync Server 2010, необходимые для построения проектируемой Системы:

1. Обмен мгновенными сообщениями и сведениями о присутствии:

- информация о присутствии (пользовательские состояния присутствия, сведения о присутствии в Microsoft Office и Share Point);
- гибкое управление контактами (группировка контактов, добавление пометок и комментариев);
- система обмена мгновенными сообщениями;
- групповые разговоры;
- веб-канал активности (статусы коллег, информация о том, чем они сейчас занимаются);
- улучшенный поиск: по нескольким словам, контактной информации.

2. Единая система конференц-связи:

- аудиоконференции (безопасная конференц-связь, гибкие возможности управления, встроенный помощник, интеграция с AD);
- видеоконференции (распознавание активно говорящего пользователя, поддержка видеоформатов высокой четкости, взаимодействие с видеоустройствами лидирующих производителей);
- совместное использование приложений и документов;
- совместное использование рабочего стола;
- виртуальная доска;
- запись собраний;
- повышенная производительность и надежность конференц-связи.

Рис. 2. Сервисы удаленного доступа к вычислительным, приборно-аналитическим и информационным ресурсам ДВО РАН

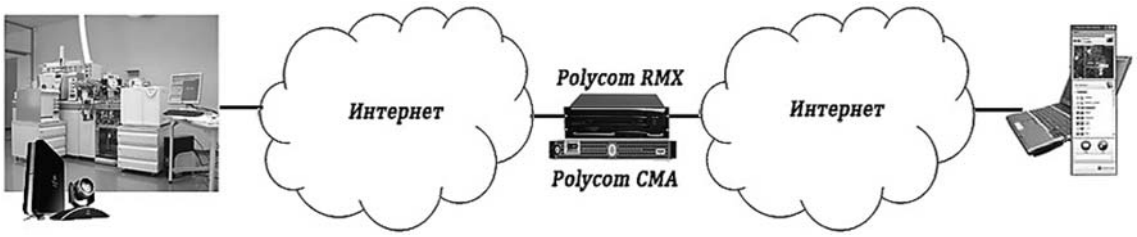


Рис. 3. Схема одного из технологических решений для организации удаленного доступа к аналитическому оборудованию, режим удаленного наблюдения за работой прибора

Основное назначение Microsoft Sharepoint 2010 — это быстрое создание внутренних сайтов для виртуальных лабораторий, проектов, совместных программ, мероприятий, личных кабинетов научных сотрудников и др. Основные задачи, которые решает подобный сайт в нашей Системе:

- публикация новостей, объявлений, календарей;
- совместный доступ к файлам;
- обсуждение различных вопросов и материалов;

и др.

Инфраструктура тестовой версии этого блока Системы базируется на сервере с системой виртуализации VMWARE ESXI, на котором развернуто пять виртуальных машин.

Сервисы удаленного доступа к научным ресурсам. Предполагается, что в виртуальную научную среду будут встроены сервисы удаленного доступа к вычислительным, аналитическим и информационным ресурсам (в нашем случае — к ресурсам Дальневосточного отделения РАН), что позволит территориально распределенным сотрудникам в виртуальной среде иметь единую точку входа к научным ресурсам для обеспечения исследований (рис. 2).

На этапе проектирования Системы нами предусматривается построение *подсистемы удаленного доступа к уникальному аналитическому оборудованию*, распложенному в Центрах коллективного пользования Дальневосточного отделения РАН (рис. 3, рис. 4, см. третью сторону обложки). При этом предлагаемые технические решения предполагают переход от использования этого оборудования в режиме удаленного наблюдения за работой прибора к проведению удаленных исследований.

Доступ будет предоставляться через единую точку входа к аналитическому оборудованию ДВО РАН.

Зарегистрированным пользователям предполагается предоставлять следующие возможности:

- обучение работе с аналитическим оборудованием;
- проведение экспериментальных исследований собственных образцов в режимах удаленного доступа к оборудованию, прямого управления из лаборатории или аудитории или по заданию заказчика без его непосредственного участия;
- участие в экспериментах на реальном оборудовании дистанционно в качестве наблюдателей;
- использование оборудования в режиме удаленного доступа для проведения удаленных исследований;
- удаленная настройка оборудования (для настройщиков оборудования).

Работа выполняется при финансовой поддержке Проекта ДВО РАН № 12-Ш-О-ОНЗ-05 "Интеграция пространственных геолого-геофизических данных и сервисов на примере разработки ГИС-портала "Геология и геофизика Дальнего Востока России" в рамках Программы фундаментальных исследований ОНЗ РАН № 7 "Геофизические данные: анализ и интерпретация".

Список литературы

1. **Дубова Н.** Академия выбирает облако // "Открытые системы". 2012. № 01. URL: http://www.osp.ru/os/2012/01/13012920/?from_mail = 1

2. **Жижимов О. Л., Молородов Ю. И., Пестунов И. А., Смирнов В. В., Федотов А. М.** Интеграция разнородных данных в задачах исследования природных экосистем // Вестник НГУ. Сер. "Информационные технологии". 2011. Т. 9. Вып. 1. С. 61—74.

3. **Копысов С. П., Новиков А. К., Рычков В. Н., Сагдеева Ю. А., Тонков Л. Е.** Виртуальная лаборатория конечно-элементного моделирования // Вестник Удмуртского университета. Компьютерные науки. 2010. Вып. 4.

4. **Малков О. Ю., Длужневская О. Б., Бартунов О. С., Золотухин И. Ю.** Международная виртуальная обсерватория: десять лет спустя // Российский научный электронный журнал "Электронные библиотеки". 2010. Вып. № 3.

5. **Наумова В. В., Сорокин А. А., Горячев И. Н.** Видеоконференцсвязь — мультимедийный сервис Корпоративной сети Дальневосточного отделения РАН // Информационные технологии. 2009. № 4. С. 66—70.

6. **Наумова В. В., Ханчук А. И., Гвишиани А. Д., Мерзлый А. М., Горячев И. Н.** Видеоконференцсвязь Отделения наук о Земле РАН: текущее состояние и перспективы развития // Открытое образование. 2010. № 5. С. 83—97.

7. **Наумова В. В., Горячев И. Н.** Разработка Системы видеоконференцсвязи Отделения наук о Земле Российской академии наук // Информационные технологии. 2011. № 3. С. 13—20.

8. **Решения** Polycom® для сред Microsoft® UC. Обзор совместного решения Polycom. URL: http://www.polycom-ua.com/assets/files/Audio/CX/uc-solution-for-microsoft-environments.pdf

9. **Фищенко В. К., Голик А. В., Антушев С. Г.** О проекте корпоративной океанологической информационно-аналитической системы ДВО РАН и задаче развертывания глобальной GRID-инфраструктуры Отделения // Открытое образование. 2008. № 4. С. 47—65.

10. **Функционирующий** в режиме удаленного доступа интерактивный учебно-научный комплекс для выполнения работ по формированию наноструктурированных материалов методом кластерного осаждения и их комплексному фазово-структурному анализу // Национальный исследовательский ядерный университет "МИФИ". URL: http://micro.maic.ru

11. **Удаленный** доступ к уникальному оборудованию наноиндустрии // Нанотехнологии и наноматеариалы. Федеральный интернет-портал http://portalnano.ru/read/databases/udalennyjdostup

УДК 004.272

Ю. Ф. Опадчий, д-р техн. наук, проф.,
Е. В. Чумакова, канд. физ.-мат. наук, доц.,
"МАТИ" — Российский государственный технологический университет имени К. Э. Циолковского,
ekat.v.ch@rambler.ru

Исследование методов вычислений элементарных математических функций и их реализация на ПЛИС

Рассматриваются алгоритмы реализации элементарных математических функций с анализом точности и быстродействия. Критерием оценки является компромисс между получением максимальных значений быстродействия, точностью вычисления и простотой их технической реализации с применением ПЛИС.

Ключевые слова: *программируемые логические интегральные схемы, алгоритм, быстродействие*

Современное развитие электронной техники характеризуется расширением области применения электронных средств, усложнением алгоритмов их работы, необходимостью создания аппаратуры управления разнообразными процессами и объектами, в том числе и быстро протекающими, в реальном масштабе времени. Одной из тенденций построения вычислительных систем реального времени является создание интегрированной вычислительной среды на базе ПЛИС. Структура ПЛИС позволяет гибко адаптироваться к заданному алгоритму обработки информации, причем сама адаптация может выполняться непосредственно во время работы устройства.

Однако при разработке на ПЛИС различных систем обработки информации часто встает вопрос о вычислении значений тригонометрических функций. К сожалению, с помощью стандартных библиотек таких популярных САПР электронной аппаратуры как MAX PLUS II и QUARTUS II нельзя решить эту задачу. Этому есть свои объективные причины. Во-первых, существует достаточно большое число алгоритмов, отличающихся как по точности и скорости вычисления, так и по аппаратным затратам на их практическую реализацию. Во-вторых, для каждого алгоритма существует некоторый оптимальный, с точки зрения точности и сложности реализации, диапазон изменения аргумента. Можно перечислить и другие причины. Однако из сказанного следует, что выбор того или иного алгоритма всегда индивидуален и в конечном счете определяется конкретными требованиями к разрабатываемой системе.

Цель настоящей работы — исследование известных методов вычисления тригонометрических функций и оценка целесообразности их реализации на ПЛИС с точки зрения быстродействия, точности вычисления и требуемых ресурсов.

Ниже приведены результаты исследований возможности применения известных методов вычисления тригонометрических функций для их реализации с использованием ПЛИС.

Исследование методов вычислений элементарных математических функций

В качестве исходных рассматривались следующие известные методы [1, 2]:

- метод выборки из таблиц с линейной интерполяцией;
- метод степенного разложения;
- метод бесконечного произведения;
- метод наилучшего многочленного приближения;
- метод рациональных приближений и цепных дробей;
- метод кубических сплайнов.

Первичный анализ позволил исключить из указанного списка методы бесконечных произведений, рациональных приближений и цепных дробей. Первый, даже при использовании произведения 20 членов, обеспечивает точность вычисления во всем диапазоне аргумента не выше 10^{-2}. Применение второго метода требует выполнения большого числа операций деления [3], практическая реализация которого является наиболее медленной.

Для метода степенных разложений были проведены исследования зависимости точности от числа членов для функции $\sin(x)$. На рис. 1 приведен по-

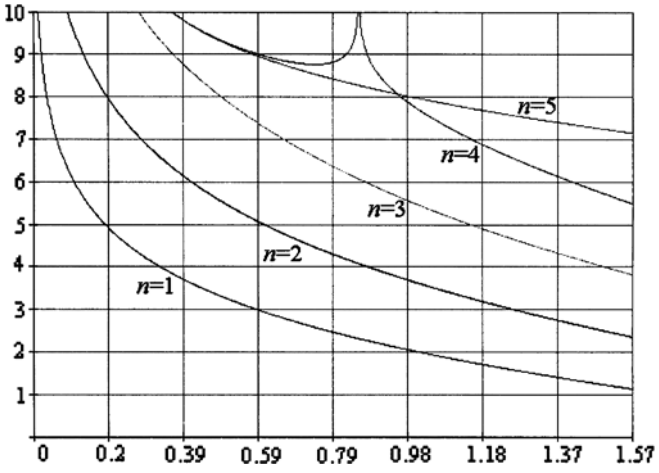


Рис. 1. Зависимость относительной погрешности от аргумента функции (*n* — параметр, характеризующий число членов в общей формуле суммы ряда)

Таблица 1

Методы вычисления функции sin(*x*)

Метод	+/−	*	/	Примечание
Выборка из таблиц с линейной интерполяцией	1/2	3	—	Точность вычисления для таблицы с $n = 1024$ — $4{,}71 \cdot 10^{-6}$
Степенное разложение	2/3	11	—	Число членов ряда $n = 5$ Точность вычисления — 10^{-7}
Наилучшее многочленное приближение	4/0	9	—	Число членов ряда $n = 4$ Точность вычисления — $1{,}1 \cdot 10^{-6}$
Кубические сплайны	5/4	12	9	Точность вычисления для $n = 17$ узлов —$1 \cdot 10^{-6}$

строенный ряд зависимостей относительной погрешности от аргумента функции для различного числа членов ряда *n*. Ось ординат — логарифмическая, а на оси абсцисс значения указаны в радианах (максимальное значение соответствует значению 90°).

Здесь можно заметить резкое увеличение точности при использовании пяти членов ряда для диапазона аргументов ~40...55°. Для обеспечения приемлемой точности необходимо вычислять не менее пяти членов ряда, что требует выполнения большого числа операций деления [1], а практическая реализация метода будет более медленной.

В табл. 1 перечислены элементарные операции, которые необходимо выполнить при вычислении функции sin(*x*) перечисленными методами.

Анализ таблицы показывает, что при примерно одинаковой точности вычисления, наименьших вычислений требуют методы выборки из таблиц с линейной интерполяцией и наилучшего многочленного приближения. Примерно аналогичные результаты получаются и при рассмотрении алгоритмов вычисления остальных из рассмотренных функций. Поэтому далее остановимся только на этих двух методах.

Метод выборки из таблицы с линейной интерполяцией базируется на ее кусочно-линейной аппроксимации, причем точность вычисления определяется количеством узловых значений функции, хранящихся в памяти ПЛИС. В общем случае максимальное значение абсолютной ошибки вычисления функции $f(x)$ на *i*-ом интервале при равностоящих узлах, т. е. при выполнении условия $x_{i+1} - x_i = \text{const} = H_x$ определяется выражением

$$\delta_M(i) = f(x_{Mi}) - A_i - \frac{A_{i+1} - A_i}{H_x}(x_{Mi} - x_i), \qquad (1)$$

где $f(x)$ — точное значение функции для заданного аргумента x; A_i и A_{i+1} — значения $f(x)$ для $x = x_i$ и $x = x_{i+1}$ соответственно, $x_i \leqslant x_{Mi} < x_{i+1}$.

Величина x_{Mi} является корнем уравнения

$$\frac{df(x)}{dx} - \frac{A_{i+1} - A_i}{H_x} = 0. \qquad (2)$$

Используя выражения (1) и (2), для заданной точности вычисления можно найти число узловых точек функции, а следовательно, необходимый объем памяти ПЛИС, требуемый при практической реализации метода. На рис. 2 для функции sin(*x*) приведена расчетная зависимость максимальной относительной погрешности $\overline{\delta_M}$ от числа узловых точек таблицы *n*.

Аппроксимирующая прямая, проходящая через две узловые точки исходной функции, описывается выражением

$$f(x) \cong A_i \frac{x_{i+1} - x}{H_x} + A_{i+1} \frac{x - x_i}{H_x}. \qquad (3)$$

На рис. 3 приведен последовательно-параллельный алгоритм, использующий рассматриваемый метод вычисления.

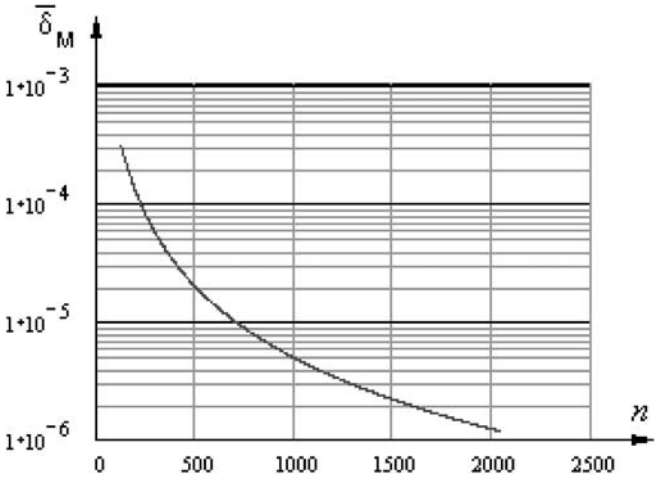


Рис. 2. Зависимость максимальной относительной ошибки вычисления функции sin(*x*) от числа узловых точек

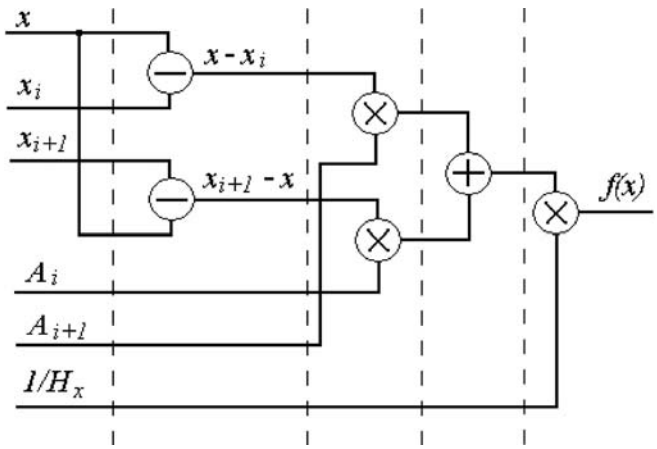


Рис. 3. Структура алгоритма последовательно-параллельных вычислений табличного метода

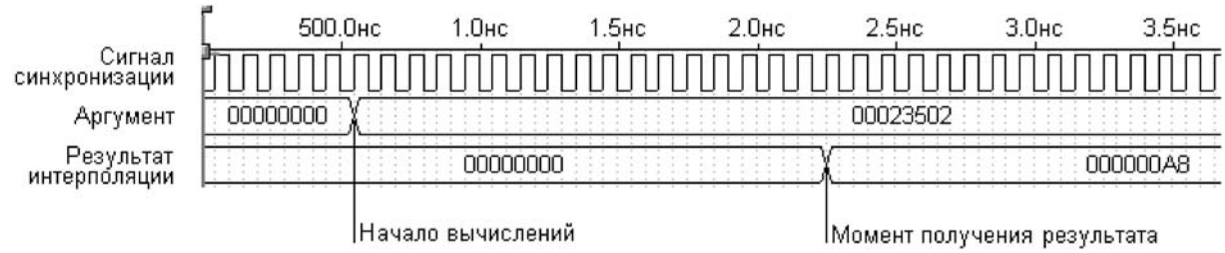


Рис. 4. Временные диаграммы, иллюстрирующие работу табличного метода

Алгоритм сводится к нахождению для заданного значения переменной x координат ближайших большей и меньшей узловых точек и выборке по ним из памяти двух значений искомой функции. Память, встроенная в ПЛИС, позволяет выбрать последовательно каждое значение за 2 такта работы. Последовательный алгоритм вычисления $\sin(x)$ выполняется за 26 тактов синхронизации, а предложенный — только за 18, т. е. примерно на треть быстрее [4]. Практическая реализация алгоритма выполнялась в среде QUARTUS II с использованием ПЛИС семейства APEX20KE. По числу занимаемых ячеек обе структуры практически одинаковы (при последовательном вычислении — 4565, в предложенной структуре — 4596). Временные диаграммы работы модуля, реализующего последовательно-параллельный алгоритм вычисления функции $\sin(x)$, приведены на рис. 4.

Следует отметить, что алгоритм рис. 3 принципиально не связан с видом исходной функции. Поэтому структура рис. 3 может быть использована при вычислении произвольных зависимостей. В каждом конкретном случае, для заданной точности вычисления, используя выражения (1), (2), необходимо оценить требуемый объем памяти ПЛИС, отводимый для хранения значений функции в узловых точках.

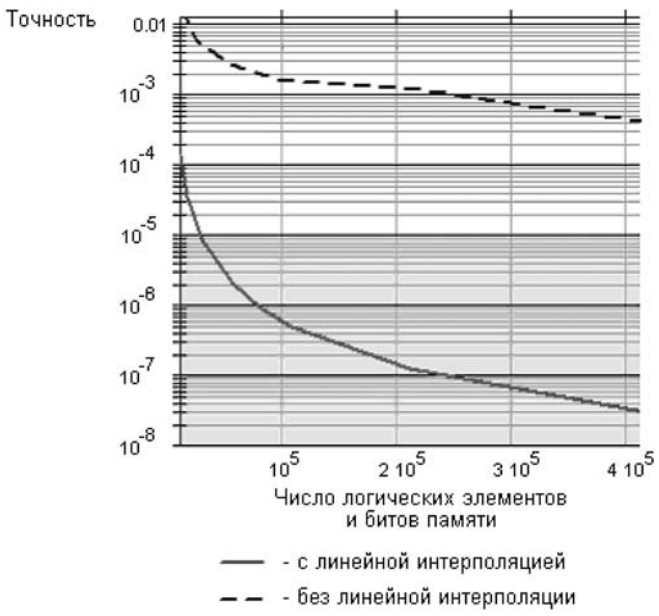


Рис. 5. Зависимость точности вычислений, реализуемых методом табличной выборки, от ресурсов ПЛИС для функции синуса

Использование метода табличной выборки без интерполяции при реализации вычислительной системы на ПЛИС дает сокращение временных затрат (3 такта — без интерполяции против 18 тактов — с линейной интерполяцией) и одновременно требует увеличения общего объема необходимой памяти с 8,3 Кбайт до 177 Мбайт. Современные ПЛИС не позволяют хранить такие большие массивы информации. Встроенная память ПЛИС составляет всего несколько десятков Кбайт. Определение наиболее оптимального варианта удобнее выполнить графическим способом. С этой целью для функции синуса были построены графики зависимостей точности вычислений табличным методом от требуемых ресурсов ПЛИС, которые приведены на рис. 5.

Из графиков видно, что для принятой точности (не менее $1 \cdot 10^{-5}$) предпочтительной является табличная выборка с линейной интерполяцией с точки зрения точности вычислений и ресурсов ПЛИС.

Метод наилучших многочленных приближений, применительно к функции $\sin(x)$, предполагает нахождение суммы ряда вида:

$$\sin\frac{\pi}{2}t = \sum_{k=0}^{3} a_{2k+1}t^{2k+1} = a_1 t + a_3 t^3 + a_5 t^5 + a_7 t^7, \quad (4)$$

где $|t| \leqslant 1$ и вычисляется из следующих условий:

$$t = \begin{cases} -2x/\pi - 2 & \text{при } -\pi \leqslant x < -\pi/2, \\ 2x/\pi & \text{при } -\pi/2 \leqslant x \leqslant \pi/2, \\ -2x/\pi + 2 & \text{при } \pi/2 < x \leqslant \pi. \end{cases}$$

Значения констант a_i задаются следующими:

$$a_1 = 1{,}57079;\ a_3 = -0{,}64592;$$
$$a_5 = 0{,}07948;\ a_7 = -0{,}00436.$$

Сравнивая результаты реализаций обоих методов, можно сделать вывод, что для функции $\sin(x)$ при почти одинаковых аппаратных ресурсах по быстродействию явное преимущество имеет метод табличной выборки с линейной интерполяцией. Похожие результаты дает исследование методов вычисления функции $\cos(x)$.

Расчеты, выполненные с применением выражений (2), (3) для функции $\arcsin(x)$, показывают, что для области определения $[-1...1]$ и точности вычисления не хуже 10^{-3} требуется хранение 16 384 узловых

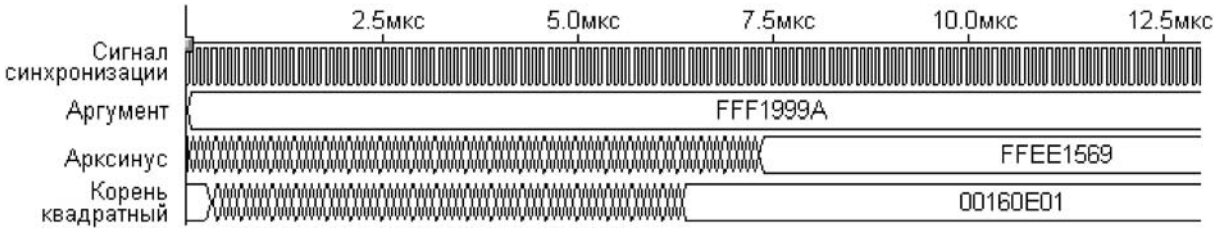


Рис. 6. Временные диаграммы функции арксинуса

значений функции. Наибольшие ошибки вычисления функции arcsin(x) соответствуют значениям аргумента по модулю, близким к единице. Используя известное соотношение

$$\arcsin x \approx \operatorname{sign}(x)\left(\frac{\pi}{2} - \arcsin\sqrt{1-x^2}\right), \qquad (5)$$

исходный интервал [−1..1] можно преобразовать в $[-1/\sqrt{2}\,...1/\sqrt{2}\,]$, что значительно снижает требуемое число узловых значений функции. Так, при $n = 512$, получаем точность вычисления arcsin(x) не хуже $1{,}5 \cdot 10^{-6}$. Однако применение такого решения требует введения операции вычисления квадратного корня.

Метод наилучшего многочленного приближения применительно к функции arcsin(x) приводит к следующему выражению:

$$\arcsin(x) = \operatorname{sign}(x)\left(\frac{\pi}{2} - \sqrt{1-|x|}\,(a_0 + a_1|x| + \right.$$

$$\left. + a_2x^2 + a_3|x|^3 + a_4x^4)\right). \qquad (6)$$

Значения констант a_i, входящих в выражение (6), задаются следующими $a_0 = 1{,}57078$; $a_1 = -0{,}21412$; $a_2 = 0{,}07948$; $a_3 = -0{,}03576$; $a_4 = 0{,}00864$.

При указанном в выражении (6) числе членов точность вычислений не хуже $8 \cdot 10^{-6}$.

Суммарное время вычисления функции arcsin(x) фактически определяется временем отыскания значения квадратного корня. Значение квадратного корня можно найти, используя итерационную формулу Герона, которая позволяет найти его на всем заданном интервале значений аргумента [1]:

$$y_{i+1} = \frac{1}{2}\left(y_i + \frac{x}{y_i}\right), \qquad (7)$$

причем

$$y_0 = \begin{cases} 0{,}57422x + 0{,}42578 & \text{при } 1 > |x| \geqslant 0{,}25, \\ 1{,}4x + 0{,}174 & \text{при } 0{,}25 > |x| \geqslant 0{,}04, \\ 4{,}1x + 0{,}06 & \text{при } 0{,}04 > |x| > 0. \end{cases}$$

Алгоритм вычислений квадратного корня, во-первых, не поддается распараллеливанию и, во-вторых, включает выполнение двух операций деления. Расчеты показывают, что, так как операция деления, связанная с преобразованием аргумента, должна выполняться и в методе табличной выборки, то с точки зрения быстродействия при вычислении функции arcsin(x) более предпочтителен метод наилучших многочленных приближений (74 против 84 тактов) [4].

На рис. 6 приведены полученные в среде Quartus II временные диаграммы моделирования алгоритма вычисления функции arcsin(x).

Сравнение методов нахождения функции arctg(x) показывает, что применение табличного метода с линейной аппроксимацией, по сравнению с методом наилучших многочленных приближений, позволяет более чем в два раза сократить суммарное время получения результата [4].

Обсуждение результатов

Окончательные результаты сравнения методов сведены в табл. 2.

Следует отметить, что в большинстве рассмотренных случаев наилучшие результаты достигаются при использовании одного и того же алгоритма, основанного на методе табличной выборки с линейной интерполяцией, т. е. на лицо унификация используемых алгоритмов вычисления. Это при практической разработке системы позволит компенсировать вынужденное увеличение необходимых аппаратных ресурсов за счет уменьшения объема ПЗУ, предназначенного для хранения самих алгоритмов программирования ПЛИС, т. е. позволит дополнительно уменьшить массу, объем и энергопотребление реальной аппаратуры, что является несомненным достоинством.

Таблица 2

Время вычисления элементарных математических функций

Функция	Название метода	Число ячеек	Время вычисления, такт
sin(x)	Табличной выборки	4596	18
cos(x)	Табличной выборки	4596	18
arcsin(x)	Наилучших многочленных приближений	15 086	74
arctg(x)	Табличной выборки	4596	18

Выводы

В результате исследования даны количественные оценки быстродействия, точности вычисления и требуемых ресурсов ПЛИС для различных методов вычисления ряда элементарных математических функций, а также практические рекомендации по их использованию. Как правило, при реализации на ПЛИС сложных алгоритмов обработки информации элементарные математические функции многократно используются в процессе вычисления. Поэтому даже незначительное снижение времени их выполнения ведет к заметному повышению общего быстродействия вычислительной структуры. Описанные выше алгоритмы позволяют исключить из цикла вычисления ряд дополнительных операций, что, естественно, снижает время их выполнения.

Список литературы

1. **Люстерник Л. А., Янпольский А. П.** Справочная математическая библиотека / Под общ. редакцией Л. А. Люстерника и А. П. Янпольского. Математический анализ. Вычисление элементарных функций. М.: Физматгиз, 1963. 248 с.
2. **Амосов А. А., Дубинский Ю. А., Копченова Н. В.** Вычислительные методы для инженеров: учеб. пособие. М.: Высш. шк., 1994. 544 с.
3. **Чумакова Е. В.** Алгоритмы операций умножения и деления для реализации на ПЛИС // Проектирование и технология электронных средств. 2005. № 2. С. 54—57.
4. **Опадчий Ю. Ф., Чумакова Е. В.** Реализация на ПЛИС вычисления элементарных математических функций // Проектирование и технология электронных средств. 2005. № 4. С. 7—12.

Информация

ИТиС 2013

С 1 по 7 сентября 2013 г.
в Калининграде состоится

РОССИЙСКАЯ АКАДЕМИЯ НАУК
ИНСТИТУТ ПРОБЛЕМ ПЕРЕДАЧИ ИНФОРМАЦИИ им. А.А.Харкевича

37-я *конференция-школа молодых ученых и специалистов*

"ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ — 2013" (ИТиС'13)

Конференция-школа ИТиС'13 будет своеобразной Universitas, состоящей из тематических семинаров по следующим основным *научным направлениям*:

- Seminarium mathematicum, или Математика и физика сложных систем
- Seminarium explorationis datorum, или Структурные методы анализа данных и оптимизации
- Seminarium protectionis informatiae, или Теория кодирования и ее приложения;
- Seminarium technologiarum retium, или Сетевые технологии и протоколы
- Seminarium operis informationis in vivo, или Биология и биоинформатика
- Seminarium linguisticum, или Компьютерная лингвистика

Кроме seminaria, впервые на конференции-школе состоятся и учебные курсы (doctrinae) по "горячим" областям исследований.

Электронный адрес организационного комитета — itas@iitp.ru,
Сайт конференции — http://itas2013.iitp.ru/rss.xml

КОМПЬЮТЕРНАЯ ГРАФИКА И ОБРАБОТКА ИЗОБРАЖЕНИЙ

УДК 004.514, 004.514.6

А. С. Зуев, канд. техн. наук,
e-mail: zuev_andrey@mail.ru
Московский государственный университет приборостроения и информатики

О возможностях реализации четырехмерных графических интерфейсов

Рассмотрены примеры реализации специальных визуальных эффектов для симуляции четырехмерных информационных пространств в графических пользовательских интерфейсах и виртуальной среде рабочего стола.

Ключевые слова: *графический интерфейс, эргономика программного обеспечения, человеко-компьютерное взаимодействие, рабочий стол*

Введение

В настоящее время в подавляющем большинстве программных продуктов (ПП) для организации человеко-компьютерного взаимодействия (HCI — *human-computer interaction*) реализованы принципы непосредственного манипулирования (DM — *direct manipulation*) с графическими пользовательскими интерфейсами (ГПИ, GUI — *graphical user interface*) класса WIMP (*Windows-Icons-Menus-Pointing device* — окна, пиктограммы, меню, позиционирующее устройство).

В. Д. Магазанчик так характеризует дисциплину HCI [1]: "Человеко-компьютерное взаимодействие — это широкая научная и прикладная дисциплина, предметом изучения которой является то, как люди используют компьютеры и как следует разрабатывать компьютерные системы, чтобы обеспечить более эффективное их использование. Дисциплина включает элементы информатики, графического дизайна, социологии и антропологии, психологии и эргономики".

Графический пользовательский интерфейс основан на визуализации объектов и процесса взаимодействия пользователя с программными и техническими средствами [2], предоставляет оператору виртуальную интерактивную среду управления их работой. Интерактивность ГПИ проявляется в учете позиции курсора на дисплее и состава выбранных объектов (например, текст, таблица, рисунок и т. п.), для которых предусмотрены опции (функции), реализуемые с помощью определенного набора элементов интерфейса.

Концепцию WIMP-интерфейсов можно пояснить следующим образом [3]:

- W — информация представляется на дисплее в виде окон;
- I — объекты представляются пиктографически в виде иконок;
- M — для группировки и представления опций используются различные меню, например, контекстные и главное;
- P — выбор объекта выполняется с помощью указательного (координатного) устройства (например, мышь, тачпад и т. п.).

Непосредственное манипулирование подразумевает возможность управления объектами ГПИ посредством обратимых действий и обратной связи. Обычной синтаксической композицией при этом является выбор объекта и активация какой-либо функции [4]. Типичными элементами ГПИ, используемыми для манипуляций с объектами и функциями, являются:

- обработчики (*handlerss*) — средства управления, непосредственно связанные с объектами, обычно проявляются после выбора объекта и могут быть "захвачены" с помощью курсора для выполнения манипуляций типа перемещения, изменения размеров, вращения и т. п.;
- управления (*controls*) — средства инициации функций или определения параметров — кнопки различного вида и поля вывода, ввода, форматного ввода;
- меню (*menus*) — совокупности управлений с типовой организацией.

На верхнем уровне систем непосредственного манипулирования обычно находится одна из метафор графического представления [5]. В настоящее время широко используется метафора "рабочий стол", определяющая первичную рабочую область — основное окно графической среды пользователя вместе с добавляемыми в него объектами и фоновым изображением [2]. Данная метафора позволяет сформировать среду виртуального рабочего стола, реализующую какие-либо решения. Эти решения расширяют соответствующую рабочую область вне физических пределов дисплея с помощью специальных функциональных возможностей ПП.

Концепция WIMP-интерфейса была предложена в 1980 г., а ее первая реализация была выполнена в 1984 г. в компьютере Apple Macintosh. В дальнейшем, с развитием вычислительной техники и мультимедиа, посредством реализации специальных визуальных и анимационных эффектов стало возможным построение пространственных интерфейсов. В. Д. Магазанчик [1]: "В пространственных интерфейсах объекты обычно плоские, но путем изменения их размера создают ощущение удаленности—приближенности. Для создания ощущения трехмерности пользуются углом освещения, размерами и наложениями одних изображений на другие". Более того, имитация трехмерного (объемного) информационного пространства размещения объектов позволяет создать более привычную для пользователя среду человеко-компьютерного взаимодействия.

В настоящее время конкурентоспособность ПП и многих видов перспективной компьютерной техники (например, мобильных телефонов, смартфонов, планшетных компьютеров и т. п.) существенно зависит от качества организации человеко-компьютерного взаимодействия посредством ГПИ. Поэтому одним из перспективных направлений исследований в сфере HCI является расширение возможностей реализации пространственных интерфейсов как в целом в ГПИ, так и в среде виртуального рабочего стола.

Примеры актуальных решений в организации среды виртуального рабочего стола посредством пространственных интерфейсов

В настоящее время разработано множество вариантов сред виртуального рабочего стола, их примерами являются BumpTop, GNOME, KDE, Xfce, LXDE, EDE, IRIX Interactive Desktop, OpenWindows, Ambient desktop, Mezzo, ROX Desktop, Unity и т. п. На рис. 1—3 (см. четвертую сторону обложки) представлены некоторые актуальные решения в организации среды виртуального рабочего стола. В приложении BumpTop объекты располагаются на внутренних поверхностях куба, а в Cube Desktop Switcher — на внешних. В iPhone 4 и iPad 2 с помощью фронтальной камеры возможно распознавание направления взгляда пользователя и симуляция трехмерного интерфейса с помощью соответствующей проекции на дисплей.

Примеры реализации четырехмерных графических интерфейсов

Интерфейс является ориентированным на человека, если он отвечает его нуждам и учитывает его слабости [6]. В существующих в настоящее время средах рабочего стола реализованы виртуальные двухмерные и трехмерные информационные пространства, наилучшим образом соответствующие привычной среде деятельности пользователя. Вместе с тем виртуальное информационное пространство потенциально может обладать большей размерностью, ограничениями для которой являются возможности вычислительной техники и мультимедиа в контексте его представления, а также способности пользователя к его восприятию. В настоящее время автором данной статьи выполняются исследования возможностей реализации специальных визуальных и анимационных эффектов для симуляции четырехмерных информационных пространств организации человеко-компьютерного взаимодействия, в том числе в среде виртуального рабочего стола.

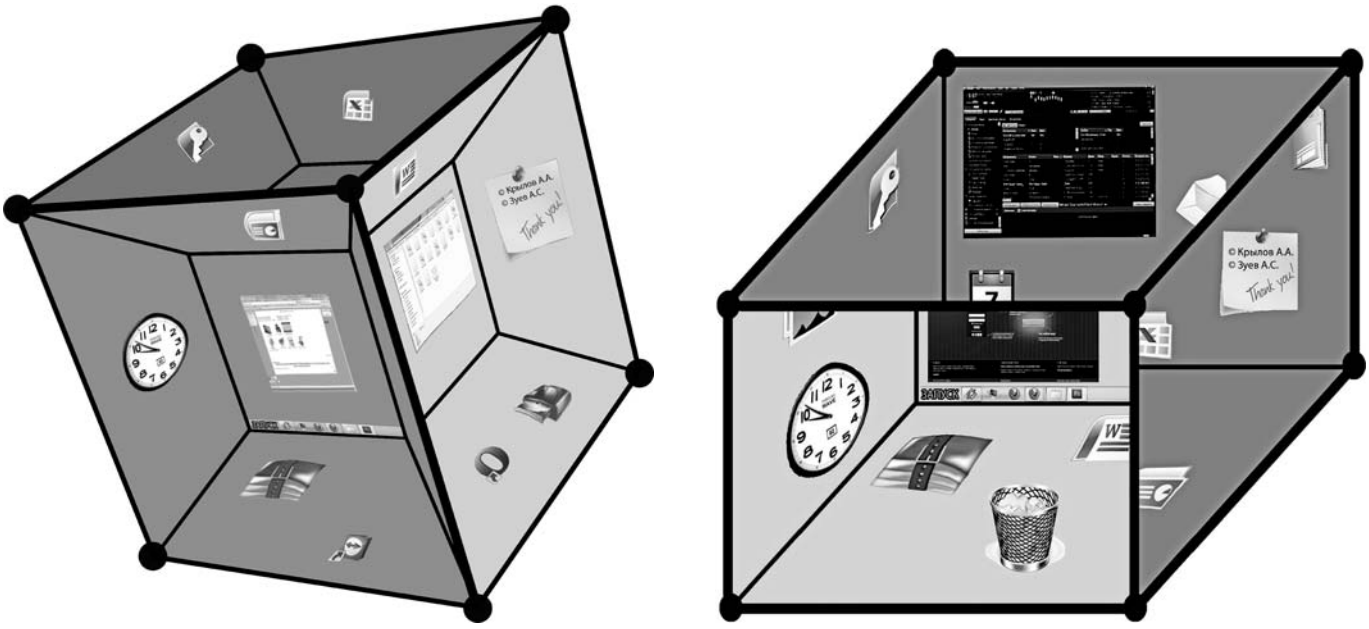


Рис. 4. Примеры трехмерного и четырехмерного информационных пространств

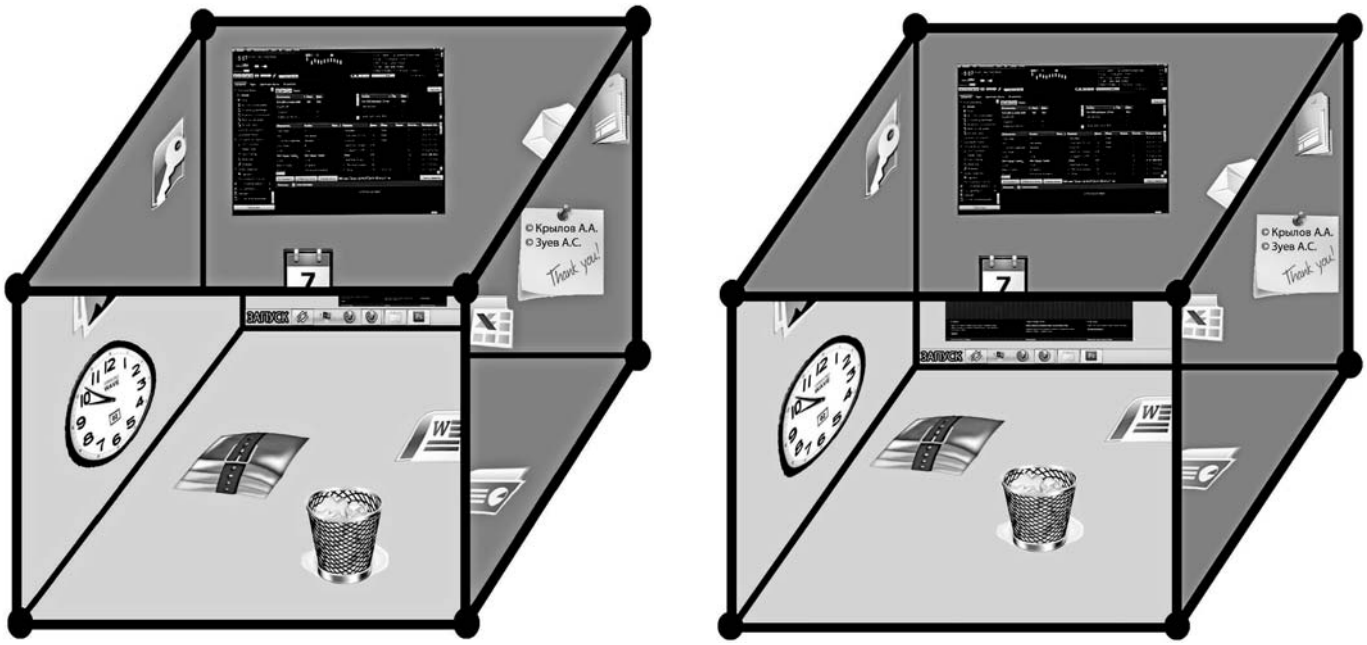


Рис. 5. Варианты отображения четырехмерного информационного пространства

На рис. 4 представлены макеты среды, отображающей виртуальные рабочие столы на гранях вращаемого куба, аналогично приложению Cube Desktop Switcher. При этом рабочие столы являются объемными, аналогично среде приложения BumpTop. Слева на рис. 4 представлен вариант реализации данной среды посредством визуального отображения проекции тессеракта на трехмерное пространство, а справа — посредством виртуального четырехмерного информационного пространства, где четвертым измерением является "непересекающийся" объем рабочих столов, соответствующих граням куба.

Допущенное на рис. 4 справа нарушение внутренних пропорций куба, выражающееся в смещении ребер в целях варьирования объемов рабочих столов, может быть выполнено в различных вариантах (например, рис. 5 слева), но может и отсутствовать (рис. 5 справа).

На рис. 6 представлены примеры симуляции четырехмерного информационного пространства без

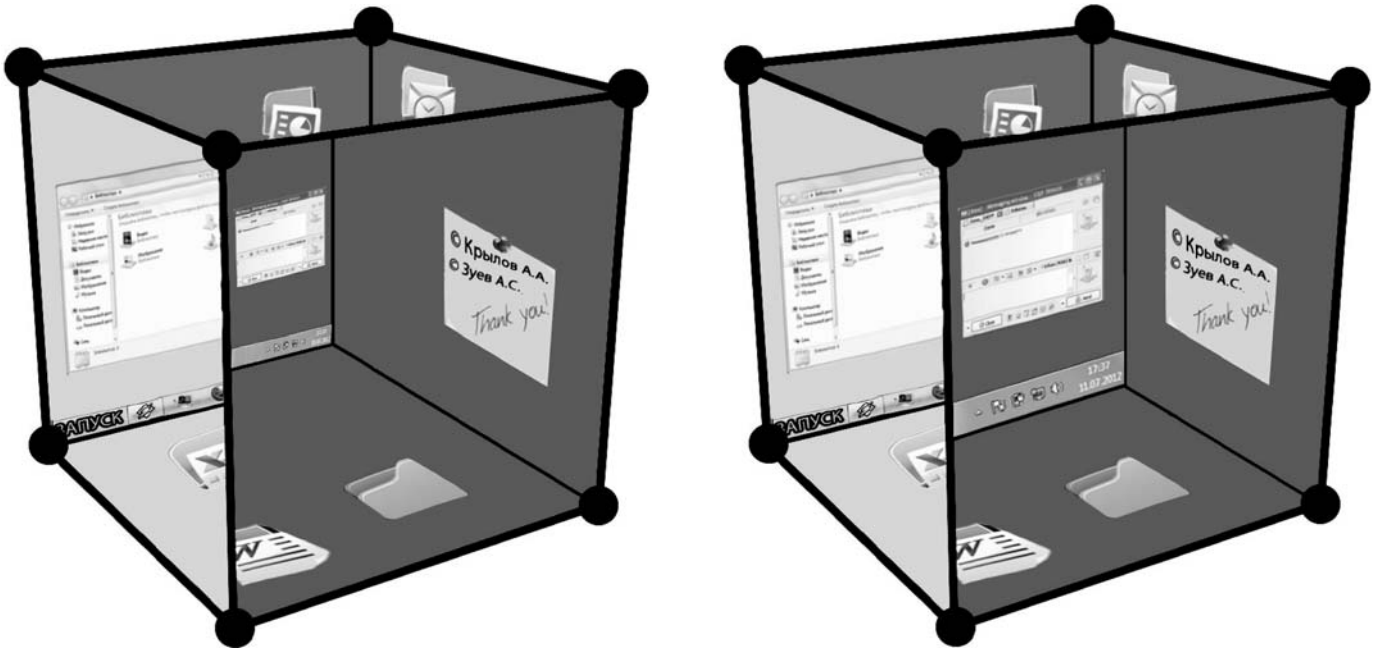


Рис. 6. Примеры симуляции четырехмерного информационного пространства

нарушения внешних пропорций куба. Заметим, что нарушение его внутренних пропорций, выражающееся в смещении ребер, может отсутствовать, как и в случае, представленном на рис. 5 справа.

В представленных выше примерах реализации четырехмерных графических интерфейсов на рис. 4 и 5 допущено нарушение внешних и внутренних пропорций куба, а на рис. 6 — нарушение только внутренних пропорций.

Возможная функциональность предложенных решений

Представленные на рис. 4—6 решения соответствуют концепции WIMP-интерфейсов, а работа с ними основана на принципах непосредственного манипулирования. Поэтому полученные результаты реализации четырехмерных графических интерфейсов являются оригинальным развитием ГПИ и принципов организации человеко-компьютерного взаимодействия.

В общем случае на поверхностях рабочих столов могут располагаться двухмерные и трехмерные объекты, например, окна приложений и стопки папок и файлов. В то же время объемными могут быть не все представленные пользователю рабочие столы (двухмерный и трехмерный режимы рабочего стола могут выбираться пользователем), а каждая грань объемного рабочего стола может содержать отдельное фоновое изображение.

На рис. 4—6 в вершинах кубов расположены элементы управления, обеспечивающие возможности:

- вращения куба и получения доступа к соответствующим его граням рабочим столам — в результате выполнения воздействия на один из элементов управления или в результате выбора одного из них и выполнения перемещения указательного координатного устройства;
- перемещения куба — в результате выбора двух элементов управления и выполнения перемещения указательного координатного устройства.

Решения по организации человеко-компьютерного взаимодействия, макеты которых представлены на рис. 4—6, позволяют объединить актуальные достижения в области создания сред виртуального рабочего стола и использовать множество существующих средств взаимодействия пользователя с программными и техническими средствами, включая Kinect-технологии, указательные и сенсорные устройства. Помимо этого внутренние поверхности, соответствующие объемным рабочим столам, могут быть использованы для отображения содержания папок, документов, web-страниц и т. п., что позволяет реализовать оригинальный способ представления данных и организации доступа к ним. Заметим также, что решения, представленные на рис. 4—6, могут рассматриваться в качестве макетов элементов ГПИ, реализуемых в программных приложениях или в операционных системах для организации работы пользователя и предоставления ему оригинальных средств взаимодействия с программным обеспечением и компьютерной техникой. При этом для отображения рабочих столов могут быть использованы и другие трехмерные объекты, например, прямоугольные параллелепипеды.

В настоящее время автору не известны аналогичные разработки и данное исследование будет являться одним из дальнейших направлений его работы в сфере вопросов организации человеко-компьютерного взаимодействия.

Заключение

Одним из направлений исследований в сфере HCI, развития пространственных интерфейсов и среды виртуального рабочего стола может являться использование специальных визуальных и анимационных эффектов для симуляции четырехмерных (а, возможно, и более) информационных пространств. При этом получаемые решения по организации человеко-компьютерного взаимодействия могут рассматриваться как расширения функциональных возможностей существующих сред виртуального рабочего стола, а также как модели оригинальных элементов графических пользовательских интерфейсов.

Список литературы

1. **Магазанчик В. Д.** Человеко-компьютерное взаимодействие: учеб. пособие. М.: Университетская книга; Логос, 2007. 256 с.
2. **Гультяев А. К., Машин В. А.** Проектирование и дизайн пользовательского интерфейса. СПб.: КОРОНА принт, 2004. 352 с.
3. **Торрес Р. Дж.** Практическое руководство по проектированию и разработке пользовательского интерфейса. М.: Вильямс, 2002. 400 с.
4. **Мандел Т.** Дизайн интерфейсов. М.: ДМК Пресс, 2005. 416 с.
5. **Baecker R. M., Gridin J., Buxton A. S., Greenberg S.** Designing to fit Human capabilities. Readings in Human-Computer Interaction: Toward the year 2000. San Francisco: Morgan Kaufmann Publishers. 1995.
6. **Раскин Д.** Интерфейс: новые направления в проектировании компьютерных систем: Пер. с англ. СПб.: Символ-Плюс, 2005. 272 с.

УДК 004:378 + 004.923

К. Д. Яшин[1], канд. техн. наук, доц., зав. каф.,
Г. В. Лосик[2], д-р психол. наук, ст. науч. сотр.,
В. В. Ткаченко[2], канд. техн. наук, зав. лаб.,
В. С. Осипович[1], канд. техн. наук, доц.,
e-mail: seth22@mail.ru,
О. А. Скаскевич[1], студент
[1] Белорусский государственный университет информатики и радиоэлектроники
[2] Объединенный институт проблем информатики Национальной академии наук Беларуси

Метод противопоставления систем искусственного интеллекта и виртуальной реальности в преподавании когнитивной графики в университете

Описывается применение трехмерной визуализации информации в рамках изучения спецкурса "Когнитивная графика". Рассмотрены дидактические возможности, строение и принцип действия некоторых современных разработок, которые используются в процессе преподавания студентам высших учебных заведений одного из разделов информатики — компьютерной графики и визуализации.

Ключевые слова: *трехмерная визуализация, стереомонитор, шлем виртуальной реальности, трекер глаз, стереоскопический видеоэкран, кластерный суперкомпьютер*

Введение

Новизна в исследованиях зачастую формируется на стыке нескольких научных направлений, поэтому их преподавание в университете требует междисциплинарного подхода, новых дидактических приемов. В настоящее время все шире входят в практику компьютерные системы трехмерной, стереоскопической визуализации образной информации. Для восприятия человеком текста, страницы сайта отсутствие глубины пространства в плоскостном дисплее или на экране мобильного телефона непринципиально. Однако оно принципиально для тренажеров, где важен эффект "присутствия" человека в виртуальном пространстве. Стереоскопическая компьютерная визуализация, обеспечивающая человеку эффект присутствия, дала толчок второй технической идее — передать управление перерисовкой воспринимаемой человеком образной стереосцены трекеру. Система компьютерной визуализации, получая информацию о направлении взора человека, управляет перерисовкой видеосцены. В итоге сегодня системы виртуальной реальности — это не только системы стереоскопической трехмерной визуализации, но и системы-тренажеры, интерактивные диагностические системы, технические параметры которых в разработке подчиняются психологическому параметру: увеличение эффекта присутствия и взаимодействия. Для студентов на практических занятиях и лекциях полезна демонстрация этих систем.

Процесс технической эволюции систем виртуальной реальности идет неоднозначным и сложным путем. Поэтому преподавание в университете требует научного анализа этой эволюции, вскрытия закономерностей, предсказания хода ее дальнейшего развития. Следовательно, та концепция, которая выбирается для преподавания систем виртуальной реальности, является проекцией в будущее развития информационных технологий (ИТ).

Состоялись первые выпуски инженеров по специальности "Инженерно-психологическое обеспечение информационных технологий (ИПОИТ)" и квалификацией инженер-системотехник в Белорусском государственном университете информатики и радиоэлектроники (БГУИР, Минск). Подготовка таких специалистов начата в рамках новых для Беларуси направления образования "Эргономика" и группы специальностей "Эргономика информационных систем". Такой инженер обеспечивает проектирование и эксплуатацию различных информационно-технических систем, созданных с использованием современных когнитивно-информационных технологий. Он является специалистом по анализу человеко-машинных информационных систем и их разработке на основе психологически обоснованных требований и параметров [1—4]. У авторов статьи, таким образом, есть опыт многолетних экспериментов по преподаванию студентам спецкурса "Когнитивная графика" и особенно такого крупного его раздела, как изучение систем виртуальной реальности и тренажеров. В основу нашей разработки данного спецкурса был положен опыт преподавания информатики на кафедре системного программирования Санкт-Петербургского государственного университета. В статье рассмотрены вопросы преподавания студентам одного из разделов информатики, а именно — компьютерной графики и визуализации, т. е. раздела GV [5].

Организация и методическое обеспечение преподавания курса "Когнитивная графика"

Подготовку инженеров-системотехников специальности ИПОИТ в БГУИР выполняет кафедра инженерной психологии и эргономики. Имеется филиал кафедры в Объединенном институте про-

блем информатики (ОИПИ) Национальной академии наук Беларуси, где реализуются практические занятия для студентов. Назовем технические средства, с которыми знакомятся будущие инженеры-системотехники в Академии наук при изучении систем трехмерной визуализации информации:

- стереомонитор StereoPixel;
- шлем виртуальной реальности i-glasses-Pro;
- стереомонитор Philips;
- трекер движения головы и трекер глаз;
- стереоскопический видеоэкран;
- кластерный суперкомпьютер СКИФ-Триада.

Стереомонитор StereoPixel имеет размер экрана по диагонали 17 дюймов (43 см). Принцип действия стереомонитора основан на совмещении двух ортогонально поляризованных изображений, полученных от пары жидкокристаллических дисплеев (рис. 1). Последующая сепарация левой/правой половины стереопары в нем осуществляется через пассивные поляризационные очки. Глубина воображаемого объемного изображения достигает 3...5 м, поэтому можно видеть комнату в объемном изображении, где совсем рядом с человеком-зрителем расположен журнальный столик с книгой, а около дальней стены комнаты — телевизор. Для проведения практических занятий со студентами используется стереодисплей модели IcReflex. Такие стереодисплеи предназначены для медицины, а также для систем автоматизированного проектирования трехмерных объектов. Полноэкранный стереорежим поддерживается видеокартой персонального компьютера на основе чипсейтов nVidia. В этом режиме на компьютере могут работать 3D-программы на основе стандартов видеокарт DirectX и OpenGL с визуализацией образной информации через StereoPixel.

Студенты усваивают как аппаратно-программные знания о системе, так и психолого-когнитивные знания о современных системах трехмерной визуализации информации.

Студенты изучают также *шлем виртуальной реальности i-glasses Pro*. Он закрепляется на голове пользователя, как очки, и поддерживает стереоэффект восприятия (рис. 2). Такой переносной видеошлем предоставляет (одновременно) для левого и правого глаза различную картинку, и за счет горизонтального смещения кадров для левого и правого глаза относительно друг друга формируется стереоэффект. Шлем состоит из двух миниатюрных экранов (дисплеев) для глаз и наушников. В каждом из окуляров шлема i-glassesPro человек видит перед собой виртуальный экран диагональю примерно 180 см. Он видит изображение на экране в объеме; глубина изображения достигает 3...5 м. Такие шлемы применяют как тренажеры в центрах подготовки спецперсонала, кто сталкивается с быстро меняющейся ситуацией и должен уметь мгновенно действовать в зависимости от возникающей проблемы. Указанный шлем подключается к VGA выходу видеокарты или видеовыходу DVD плеера.

Студенты изучают *трекер отслеживания поворотов головы*. Он представляет собой устройство, встраиваемое в шлем виртуальной реальности. Трекер предназначен для перерисовки объемной картинки в шлеме. Он дает возможность, например, увидеть салон автомобиля либо со стороны водителя, либо со стороны пассажира. Перемещение объемной картинки осуществляется трекером за счет движения головы человека, на которую надет шлем виртуальной реальности.

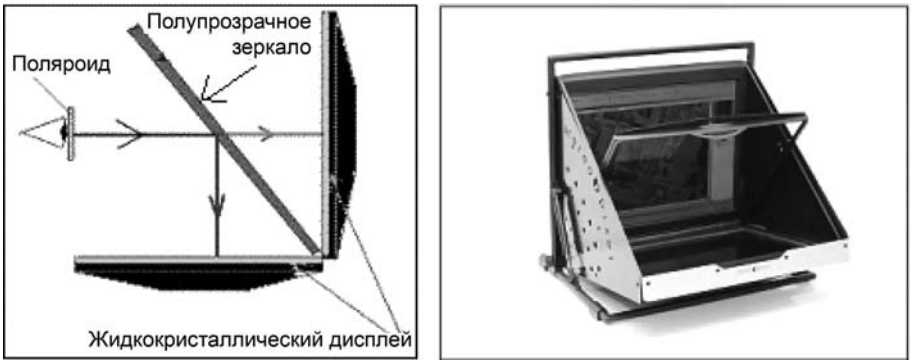


Рис. 1. Стереомонитор StereoPixel

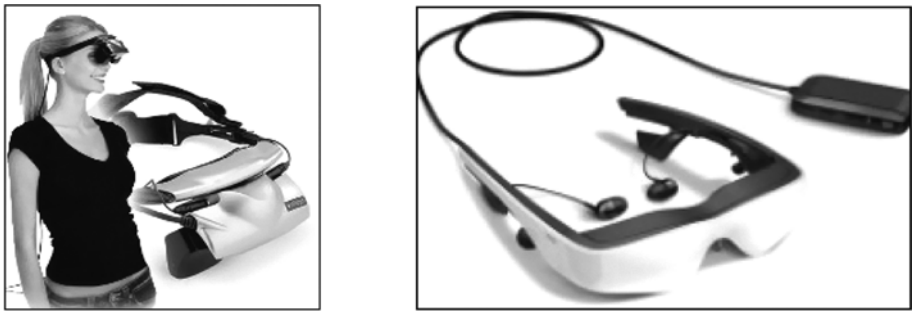


Рис. 2. Шлем виртуальной реальности

Студенты знакомятся со *стереомонитором трехмерного изображения Philips*. Его экран имеет размер 48 × 65 см и крепится на стене. Глубина воображаемой объемной картинки достигает 1...2 м. По всей площади монитора вертикально расположены 500...600 оптических полуцилиндров. На жидкокристаллический монитор подаются два изображения — для левого и правого глаза. Полуцилиндры с частотой 60 Гц подают изображение влево-вправо, и пространство между зрителем и экраном пронизано многими лучами. В итоге образуется 6...8 мест перед экраном

монитора, где человек может наблюдать объемное изображение, если находит для себя удобное место перед экраном. Важно, что просмотр трехмерного изображения в этом случае не требует применения специальных очков.

Система управления компьютером движением глаз перемещает курсор по экрану монитора. Глазное яблоко — электрический диполь и при движении глаз в районе глазниц регистрируются изменения потенциалов. Знак потенциала зависит от направления поворота глаз, а амплитуда — от угла поворота глаз. Разработчики системы используют четыре электрода, закрепляя их около глаз человека. Нижние электроды служат для оценки вертикальной составляющей смещений взгляда человека (взора). Боковые электроды служат для оценки горизонтальной составляющей взора. По картинке потенциалов можно описать траекторию взгляда человека (взора) и использовать глаза в режиме реального времени для управления компьютером, например, для перемещения экранного курсора. Разработка относится к так называемым мозг-компьютерным интерфейсам.

В распоряжении преподавателя имеется *стереоскопический экран*, который предназначен для создания комнат виртуальной реальности, где стенами являются 3...4 таких стереоэкрана (рис. 3, см. третью сторону обложки). Размер экрана 2 × 3 м. Ткань экрана способна отражать свет двух поляризаций. На экран проецируется изображение одновременно с двух видеопроекторов. Поляризационные линзы к проекторам позволяют получать изображения — с одного из них с горизонтальной поляризацией света, а с другого — с вертикальной поляризацией. Человек снабжается специальными очками и они восстанавливают объемную картину. При этом мозг человека воспринимает ощущение глубины пространств 2...3 м. Поляризационные линзы к видеопроекторам разработаны в ОИПИ НАН Беларуси.

Приборная панель автомобиля создана на основе дисплея с объемным изображением картинок участков автомобильных дорог. Два налагаемых друг на друга изображения воспроизводят 3D-эффект на дисплее. После поворота ключа зажигания на дисплее автомобиля появляется объемная модель города (рис. 4, см. третью сторону обложки). Встроенная навигационная система направляет автомобилиста к месту назначения. Две камеры расположены прямо перед водителем и направлены на него. В режиме реального времени камера фиксирует положение глаз водителя и расстояние между водителем и камерами. В результате в каждый момент времени изображения на 3D-дисплее приспосабливаются к зрительному аппарату водителя. Это гарантирует эффект объемного видения даже при быстром (беглом) взгляде водителя на панель и при любом положении водителя в кресле. Главное, для просмотра 3D-изображения не требуется специальных очков.

Студенты знакомятся с *суперкомпьютером серии СКИФ* разработки ОИПИ НАН Беларуси. Первая его версия — это большой суперкомпьютер с 60-ядерным процессором. Вторая версия — серия персональных кластеров СКИФ-Триада (рис. 5). Третья версия — серия "малых" компьютеров для персонального применения (еще меньших, чем персональные кластеры СКИФ-Триада). Четвертый этап: идет разработка еще большего, чем первый СКИФ, суперкомпьютера, полностью отвечающего современным международным стандартам.

Компьютеры СКИФ-Триада разработаны в рамках программы Союзного государства России и Беларуси. Они заполняют нишу, разделяющую обычные ПК и суперкомпьютеры, обеспечивая возможность использования суперкомпьютерных технологий в отдельно взятой организации. Компьютеры позволяют пользователям адаптироваться к переходу на вычислительную технику с параллельной архитектурой, которая существенно изменяет привычный стиль взаимодействия с компьютером. В семействе персональных кластеров СКИФ-Триада реализована возможность работы под управлением как операционной системы Linux, так и операционной системы WindowsComputeClusterServer 2003. Опытный образец кластера СКИФ-Триада применяется в качестве высокопроизводительного вычислительного устройства в аппаратно-программном комплексе слежения в реальном масштабе времени за

Рис. 5. Компьютер СКИФ-Триада

движущимися объектами. Второй образец используется в противотуберкулезном диспансере Минска в качестве высокопроизводительного вычислительного устройства в распределенной телемедицинской системе реального времени по цифровой флюорографии. К кластеру подключены городские поликлиники, объединенные в единую телекоммуникационную сеть. Третий образец используется в ОИПИ для проведения научных и инженерных расчетов и моделирования. Четвертый образец персонального кластера используется в ОИПИ в составе gLite сайта "BY-UIIP" грид-сегмента Baltic-grid общеевропейской грид-сети.

Противопоставление концепций искусственного интеллекта и систем виртуальной реальности

Можно констатировать, что курсы лекций: системы искусственного интеллекта и системы виртуальной реальности близки по тематике к частным научным вопросам, которые входят в их состав. Оба курса непосредственно касаются вопросов переработки информации мозгом и компьютером, вопросов обучаемости нейронных структур, алгоритмов обучения с преподавателем и самообучения, константности распознавания образов. В то же время указанные два научных направления различаются концепциями, по которым исследователи используют научные сведения о работе компьютера и мозга человека. Поэтому в педагогических целях преподаватель может противопоставлять эти концепции. При создании систем искусственного интеллекта ставится цель заменить ею мозг человека, в то время как при создании систем виртуальной реальности ставится цель состыковать с мозгом современный компьютер, придав ему функции своеобразного нового "полушария" мозга. Появление систем виртуальной реальности стало возможным только недавно и объясняется не теоретическими, а технологическими успехами науки и техники. Значительно возросла быстрота работы процессора, возросли разрешающая способность и цветность дисплеев, появились трекеры, мгновенно отслеживающие движения разных органов человека. Вместе с тем, системы виртуальной реальности рассчитаны на интерактивное взаимодействие компьютера с мозгом человека сугубо в сфере образной, а не вербальной переработки информации. В итоге в ходе обучения студентов по курсу "Когнитивная графика" мы отмечаем превосходство систем искусственного интеллекта над системами виртуальной реальности и тренажерами в области обработки вербальной, знаковой информации, анализа текстовой информации. Нами подчеркивается в дидактических целях перспективность систем виртуальной реальности как концепции сохранить за мозгом в тандеме "мозг—компьютер" функцию совершать мыслительный акт, генерировать новую информацию. Эта концепция отрицает возможность нейросетевой модели искусственного интеллекта генерировать новые знания в области образной информации.

В системе виртуальной реальности как подсистема обязательно присутствуют мозг человека, зрение, рефлекторные навыки. Поэтому мы отводим большое внимание в учебном процессе рассмотрению эффекта присутствия и эффекта взаимодействия. Эти два известных феномена мозга невозможно смоделировать в системах искусственного интеллекта.

В то же время в дидактическом плане в курсе "Когнитивная графика" рассматривается сходство архитектуры обучающих систем в области искусственного интеллекта и виртуальной реальности. Они условно противопоставляются и называются в первом случае — системами обучения текстовым знаниям, во втором случае — системами обучения образным знаниям. Обучающая система и в том и в другом случае предъявляет студенту порцию новой информации, регистрирует его реакцию как ответ на восприятие, измеряет ошибку и по запрограммированному сценарию корректирует ошибку неким сообщением в адрес студента. Аналогично архитектуре систем искусственного интеллекта (обучающим студента текстовым знаниям) мы в курсе "Когнитивная графика" рассматриваем архитектуру системы обучения образным знаниям (архитектуру тренажера). Она состоит из пяти модулей: модуль системы отображения информации для предъявления студенту стереоизображения, модуль регистрации реакции, модуль измерения ошибки, интеллектуальный модуль алгоритма обучения и модуль истории тренинга. Если в системе отсутствуют два последних модуля, она превращается в диагностическую, если три последних — в автоматизированное рабочее место.

Перспективы развития научно-технического и образовательного направления

Изучение современных систем трехмерной визуализации информации предполагает широкую самостоятельную работу студентов при выполнении курсовых и дипломных проектов. В курсе "Когнитивная графика" формулируются актуальные для медицины, промышленности и образования прикладные темы:

1. Разработка алгоритмов для тренажеров виртуальной реальности по формированию у человека сенсомоторных, инструментальных и навигационных навыков.

2. Разработка методов и средств сканирования, визуализации и интерактивного управления трехмерной машинной графикой.

3. Разработка тренажера для развития внимания.

4. Разработка тренажера на базе шлема виртуальной реальности для обучения зрительному таможенному осмотру.

5. Разработка тренажера для обучения хирургов.

6. Разработка систем диагностики когнитивных функций человека по трекингу его взора.

7. Разработка тренажера для пожарных.

8. Разработка тренажера для авиадиспетчеров.

9. Разработка тренажера для обучения работе на станках с числовым программным управлением.

Разработка тренажеров с 3D-изображением для подготовки медицинских специалистов обеспечивает: привыкание обучающихся к медицинским и хирургическим сценам; приобретение навыков идентификации и анатомического осмотра внутренних органов человека; приобретение навыков диагностики заболеваний по визуальным картинам органов человека и др.

Разработка компьютерного тренажера с 3D-изображением для подготовки специалистов в области авиационной техники обеспечивает: приобретение навыков распознания и идентификации узлов и агрегатов самолетов и вертолетов; приобретение навыков осмотра и оценки работоспособности узлов и агрегатов самолетов и вертолетов по визуальным картинкам этой аппаратуры и др.

Разработка компьютерного тренажера с 3D-изображением для подготовки специалистов по управлению беспилотными летательными аппаратами обеспечивает приобретение навыков восприятия операторами речевых сообщений, посылаемых аппаратурой летательного аппарата и др. Разработка манипулятора для передачи в руку пользователя персонального компьютера ощущения упругости и гибкости виртуального предмета, наблюдаемого на мониторе компьютера, в целях передачи пользователю не только визуального восприятия, но и тактильного восприятия этого предмета, может быть использована: в тренажере для хирургов; при разработке Интернет-магазина; при разработке информационной компьютерной системы для слепых людей и реализации возможности ощупывать с помощью компьютера трудно доступные для них предметы и др.

Выводы

Разработана концепция преподавания в университете вопросов дальнейшего развития (техногенеза) информационных технологий, в основе которого положен принцип дидактического противопоставления развития систем искусственного интеллекта и развития систем виртуальной реальности. Лекционные курсы "Системы искусственного интеллекта" читаются в БГУИР более 15 лет. Сюда входят спецкурсы "Нейросетевые модели", "Автоматическое распознавание образов", "Экспертные системы". Студентам хорошо известно это самостоятельное направление в информационных технологиях. Новое направление в информационных технологиях — системы виртуальной реальности — нельзя рассматривать как звено направления искусственного интеллекта. Поэтому с дидактической целью в нашем преподавании эти два направления сравниваются, анализируются и противопоставляются.

При подготовке статьи были использованы информационные ресурсы: www.stereo-pixel.ru, www.politex.by, www.really.ru, www.3dhd.ru, www.ohgizmo.com, www.irinagruzdeva.blogspot.com, www.nvworld.ru, www.reghardware.com, www.niievm.by, www.by.all.biz/g9945/, www.armsexpo.ru.

Список литературы

1. **Общеобразовательный** стандарт Республики Беларусь ОС РБ II-58 01 01—2007 "Высшее образование, первая ступень, специальность I-58-01 01 Инженерно-психологическое обеспечение информационных технологий, квалификация инженер-системотехник". Разработан Белорусским государственным университетом информатики и радиоэлектроники (Шупейко И. Г., Яшин К. Д.), утвержден и введен в действие постановлением Министерства образования Республики Беларусь от 2.05.2008, № 4, Минск.

2. **Борисенко В. Е., Олекс О. А., Прокопчик Т. К., Яшин К. Д.** Инженерно-психологическое обеспечение информационных технологий // Вышэйшая школа. 2005. № 4. С. 18—20.

3. **Борисенко В. Е., Осипов А. Н., Яшин К. Д.** Инженерная психология информационных технологий. Парк высоких технологий — путь в интеллектуальное мировое сообщество // Докл. IV Междунар. конгресса "Развитие информатизации и системы научно-технической информации в Республике Беларусь" (Минск, 2004 г.). С. 262—267.

4. **Шупейко И. Г.** Эргономика в Беларуси // Вышэйшая школа. 2007. № 5. С. 54—57.

5. **Рекомендации** по преподаванию информатики в университете: Пер. с англ. СПб.: Санкт-Петербургский университет, 2002. 372 с.

УДК 004.8

В. В. Грибова[1, 2], д-р техн. наук, проф., зав. лаб.
П. А. Заморова[2], студент,
[1] Институт автоматики и процессов управления ДВО РАН,
e-mail: gribova@iacp.dvo.ru
[2] Дальневосточный федеральный университет,

Использование методов искусственного интеллекта при разработке медицинских диагностических компьютерных тренажеров

Описана общая концепция и архитектура программного комплекса, совмещающего в себе инструментарий для создания, сопровождения, функционирования диагностических медицинских тренажеров и медицинский диагностический компьютерный тренажер на основе знаний. Определены классы пользователей тренажера и задачи, ими решаемые. Описаны информационные и программные компоненты, входящие в состав программного комплекса.

Ключевые слова: *компьютерный тренажер, онтология, база знаний, интеллектуальная система*

Введение

Развитие информационных технологий и сети Интернет требуют разработки новых подходов к организации учебного процесса. Одним из эффективных средств обучения студентов являются компьютерные тренажеры, причина возрастающей популярности которых заключается в их способности реализовать активные методики обучения, улучшить качество обучения, снизить его стоимость, дать возможность обучаемым "проиграть" различные практические ситуации до начала их работы в реальных условиях [1]. Особо остро эта проблема стоит в медицине, поскольку в странах Евросоюза запрещено обучение студентов на реальных больных, а в России требуется обязательное их согласие [2].

По характеру стоящих перед тренажерами задач можно выделить три типа тренажеров [3]: диагностические; предназначенные для отработки моторно-рефлекторных реакций и навыков; смешанный тип. Несмотря на важность и потребность всех типов тренажеров, особо востребованными в медицине являются диагностические тренажеры, предназначенные для отработки навыков постановки диагноза студентами на основе полученных значений наблюдений, отработке различных диагностических ситуаций, которые могут возникнуть в практической деятельности врача.

На сегодняшний день специализированные средства, снижающие трудоемкость при проектировании, генерации и отладке тренажеров отсутствуют. Их создают высококвалифицированные команды профессиональных программистов, имеющих соответствующую квалификацию и специализацию с использованием разнородных программных сред и компонентов (программных библиотек). Трудоемкость создания и последующего сопровождения медицинских компьютерных тренажеров является основной причиной их исключительно медленного развития и внедрения в практику отечественной медицины [1].

Таким образом, актуальной задачей является разработка специализированных инструментальных средств, снижающих трудоемкость создания и сопровождения медицинских диагностических тренажеров. Цель данной работы — описание функций и архитектуры медицинского диагностического компьютерного тренажера, концепции инструментария для его разработки на основе знаний, а также описание информационных и программных компонентов такого тренажера.

Постановка задачи

Любой диагностический тренажер, в том числе медицинский, предназначен для отработки у обучаемого навыков диагностики некоторого объекта, медицинский тренажер — для постановки диагноза больного на основе набора наблюдений (жалоб, результатов объективных методов исследований, в том числе лабораторных и инструментальных).

Задачу диагностики можно разбить на две подзадачи. Первая подзадача — формирование необходимого и достаточного множества наблюдений по первоначальному подмножеству. Вторая подзадача — непосредственно постановка правильного диагноза по сформированному множеству наблюдений. Первоначальным подмножеством таких наблюдений являются жалобы больного. По этому подмножеству наблюдений обучаемый должен понять, какие

еще наблюдения необходимо провести (дополнительный наружный осмотр, лабораторные, инструментальные методы исследования), чтобы поставить правильный диагноз. Набор наблюдений в совокупности с их значениями должен позволять однозначно определить, каким заболеванием (или заболеваниями) болен пациент. Тренажер является, прежде всего, обучающей системой, поэтому важной его функцией является объяснение результата диагностики (особенно в случае ошибочной постановки диагноза) с указанием, какие наблюдения из множества им сформированных входят либо не входят в клиническую картину заболевания. Из сказанного очевидно, что диагностический медицинский тренажер является интеллектуальной системой, в ядро которого входят знания о заболеваниях.

Любые знания постоянно изменяются и модифицируются, поэтому жизнеспособность диагностического тренажера может быть достигнута только в случае, если знания, входящие в тренажер, во-первых, будут постоянно обновляться и модифицироваться; во-вторых, они не будут ограничены жестко заданным разделом медицины либо группой заболеваний; в-третьих, будет обеспечена возможность формирования различных диагностических ситуаций.

Для любого диагностического медицинского компьютерного тренажера можно выделить четыре основные категории пользователей.

1. *Обучаемые* — основная группа пользователей (студенты, интерны, врачи), использующие тренажер для отработки навыков постановки диагноза в различных диагностических ситуациях. Выполнив упражнение, обучаемый должен иметь возможность узнать, правильный ли диагноз он поставил, а также ознакомиться с объяснением подтверждения или отклонения поставленного обучаемым диагноза.

2. *Преподаватели* — группа пользователей системы, отвечающих за формирование упражнений (диагностических ситуаций), предоставляемых обучаемым. Преподаватели должны иметь возможность распределять формируемые упражнения по темам. Множество тем также задается преподавателем. Генерация некоторых значений наблюдений, соответствующих выбранному им заболеванию, может быть поручена преподавателем соответствующей подсистеме тренажера.

3. *Администратор* — пользователь системы, наполняющий соответствующие компоненты системы информацией об обучаемых.

4. *Эксперты предметной области* — группа пользователей системы, являющихся специалистами в одном из разделов медицины. Эксперты наполняют знаниями соответствующие компоненты тренажера и обеспечивают их модификацию в процессе его жизненного цикла: формируют множество заболеваний, множество наблюдений, задают связи между тем или иным заболеванием и набором наблюдений, которые следует рассматривать при данном заболевании, указывают, какое значение (либо множество значений) имеет то или иное наблюдение при конкретном заболевании.

Инструментарий для создания медицинского диагностического компьютерного тренажера должен содержать специализированные средства для всех рассмотренных выше групп пользователей, с помощью которых пользователь той или иной группы будет осуществлять проектирование и модификацию соответствующих компонентов системы.

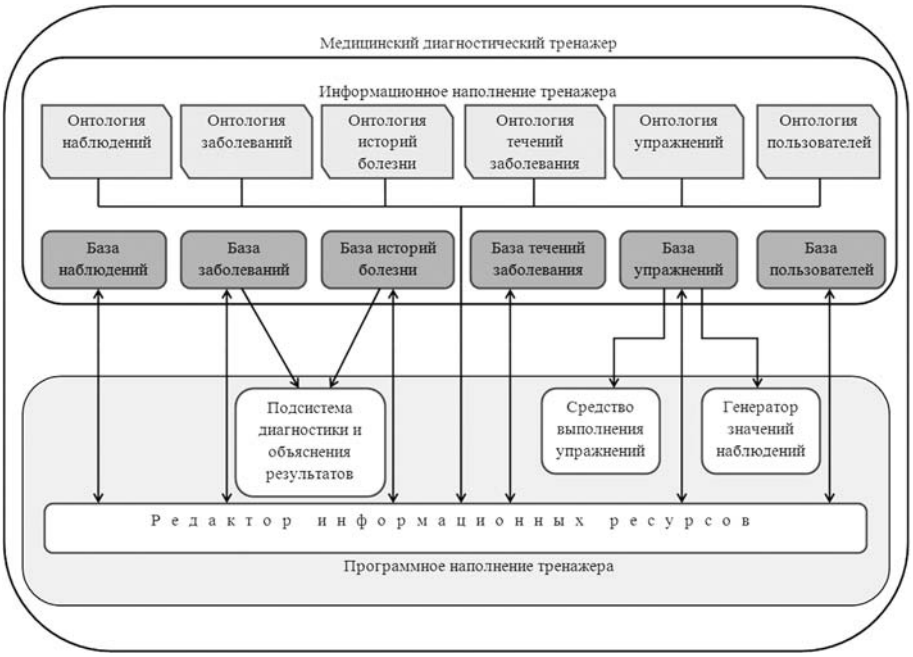
Поддержка знаний в актуальном состоянии может быть достигнута, если база знаний не передается конечным пользователям, а находится у разработчика (иначе невозможно обеспечить ее своевременную и перманентную модификацию). Для реализации данного требования тренажер должен быть разработан как Интернет-сервис (на основе технологии облачных вычислений), что дополнительно обеспечит расширение аудитории пользователей и его доступность. Включение экспертов на всех этапах жизненного цикла тренажера возможно, если знания, входящие в состав тренажера, им понятны. Известным способом формирования знаний, понятным специалистам, является онтология.

Таким образом, актуальной задачей является разработка интеллектуального Интернет-сервиса для создания, сопровождения и функционирования медицинских диагностических компьютерных тренажеров (МДКТ) в различных разделах медицины с предоставлением специализированных средств для всех групп пользователей, возможностью формирования множества диагностических ситуаций для каждого заболевания либо группы заболеваний, а также обеспечивающего обучаемых системой объяснений результатов проведенной им диагностики.

Концептуальная архитектура инструментария

Архитектура интеллектуального Интернет-сервиса для создания, сопровождения и функционирования диагностических компьютерных тренажеров (см. рисунок) состоит из информационных и программных компонентов.

Информационные компоненты включают онтологии, базы данных и знаний. Онтологии необходимы для того, чтобы пользователи системы — эксперты предметной области, администраторы и преподаватели в их терминах могли формировать и модифицировать соответствующие базы знаний и данных. Онтологиями, входящими в архитектуру МДКТ и предназначенными для экспертов, являются: онтология заболеваний, онтология наблюдений; онтология течения заболевания. Онтологиями, предназначенными для преподавателей, являются: онтология истории болезни и онтология упражнений. Онтологией для администратора является онтология пользователей системы.



Концептуальная архитектура медицинского диагностического тренажера

В соответствии с концепцией, предложенной в работе [3], все информационные компоненты представляются в едином унифицированном формате — в виде семантических сетей, что обеспечивает единые принципы для их формирования, доступа и модифицирования.

Программные компоненты МДКТ — это редакторы знаний и данных, управляемые онтологиями и ориентированные на интеллектуальную поддержку пользователей тренажера, средства по их обработке, а также административная подсистема. Программные компоненты, обеспечивающие обработку знаний и данных — это подсистема диагностики и объяснения, генератор значений наблюдений, а также средство выполнения упражнений.

Платформой для реализации МДКТ является Интернет-комплекс IACPaaS (Intelligence application, control and platform as a service) [4], предназначенный для создания и использования интеллектуальных сервисов и их компонентов, представленных семантическими сетями [5].

В основе платформы лежит технология, среда и программно-информационный комплекс, основанные на технологии облачных вычислений и обеспечивающие удаленный доступ конечным пользователям к интеллектуальным системам, а разработчикам и управляющим — к средствам создания интеллектуальных систем и управления ими. Основные архитектурные компоненты Интернет-комплекса IACPaaS следующие: административная система; виртуальная машина; фонд. Административная система предназначена для всех пользователей проекта. Через нее они могут просматривать доступное им содержимое фонда, подавать заявки на регистрацию в предметных областях фонда, регистрацию полномочий, модификацию фонда, а также реализовывать свои полномочия. Виртуальная машина представляет собой совокупность трех процессоров: процессора информационных ресурсов; процессора решателей задач; процессора пользовательского интерфейса, а также ряда вспомогательных средств. Каждый процессор представляет собой набор функций для поддержки соответствующих компонентов интеллектуального приложения. Фонд представляет собой совокупность единиц хранения — программных и информационных ресурсов различных типов; для удобства навигации он разделен на предметные области.

Информационные компоненты

Информационные компоненты МДКТ включают онтологии, знания и данные.

1. *Онтология наблюдений и база наблюдений.* В качестве онтологии наблюдений используется онтология наблюдений, описанная в работе [6]. Она включает названия групп наблюдений, а также конкретных признаков, особенностей, событий, названия характеристик, значения наблюдений. Онтология наблюдений задает совместность значений наблюдений и единицы измерения. Знания о наблюдении (помимо названия) включают область его возможных значений (множество значений).

База наблюдений задается экспертами в области медицины на начальном этапе формирования знаний, после чего ее содержимое используется при формировании баз знаний о заболеваниях и базы историй болезни пациентов, описанных ниже. Эта база может дополняться в процессе эксплуатации системы новыми терминами.

2. *Онтология заболеваний и база знаний о заболеваниях.* Онтология заболеваний описывает процессы, происходящие в организме пациента и окружающей его среде, которые существенны при решении задачи медицинской диагностики. В качестве данного компонента при разработке системы используется онтология заболеваний, описанная в работе [6]. Онтология заболеваний задает структуру описания заболеваний (значения признаков при заболеваниях задаются клиническими проявлениями и клиническими проявлениями, измененными воздействием событий), их причин (задаются этиологиями) и нормального состояния пациента (описываются нормальные реакции и реакции на воздействие со-

бытий). Знания о заболевании (помимо его названия) включают описание его периодов развития. Описание периодов развития заболеваний представляет собой последовательность описания периодов. Каждое описание периода содержит интервал возможных длительностей этого периода. Каждое заболевание, входящее в диагноз пациента, протекает в соответствии с описанием периодов его развития, т. е. длительность каждого периода его развития в действительности принадлежит соответствующему этому периоду интервалу его возможных длительностей.

База знаний, основанная на модели онтологии заболеваний, содержит описания заболеваний, их причин и нормального состояния пациента.

3. *Онтология истории болезни и база историй болезни*. Онтология истории болезни задает структуру описания конкретной истории болезни реального пациента. В качестве данного компонента при разработке системы используется онтология истории болезни, описанная в работе [7]. История болезни представляет собой описание состояния пациента в моменты проведения осмотров и до поступления в клинику: задаются значения событий, признаков, и анатомо-физиологических особенностей. События могут происходить с пациентом в различные моменты времени и иметь различные значения. Признаки также в разные моменты времени могут иметь различные значения, а анатомо-физиологические особенности пациента имеют постоянные во времени значения (время наблюдения для них не указывается).

4. *Онтология течения заболевания и база течений заболеваний*. Онтология течения заболевания описывает структуру терминов, необходимых для описания конкретного течения заболевания и связей между ними. Течение заболевания является описанием развития значений признаков во времени. Для каждого наблюдаемого признака течение заболевания включает указание длительностей периодов динамики соответствующего ему клинического проявления. Для каждого периода динамики указывается одно или несколько (в случае признаков с совместными значениями) значений признака.

5. *Онтология упражнений и база упражнений*. В рамках разрабатываемого медицинского диагностического компьютерного тренажера под упражнением подразумевается совокупность истории болезни пациента и течения заболевания, которым болен пациент. Онтология упражнений описывает термины, необходимые для описания конкретного упражнения: указатель на эталонную историю болезни и термины для описания результатов, полученных обучаемыми, которые выполнили это упражнение. К ним относятся имя пользователя, ссылка на историю болезни пользователя, результирующий диагноз. База упражнений содержит сформированные преподавателями упражнения.

6. *Онтология обучаемого и база обучаемых*. Онтология обучаемого описывает структуру терминов, необходимых для описания пользователя системы, и связей между ними. В качестве данного компонента при разработке системы использована система пользователей проекта IACPaaS, описанная в работе [8]. База пользователей хранит учетную информацию об обучаемых и результаты сеансов выполнения упражнений. Из базы обучаемых можно почерпнуть информацию о том, какие темы прослушаны конкретным обучаемым, какие темы ему рекомендованы, какие из множества прослушанных тем были проверены в ходе выполнения упражнений, а также информацию о результатах выполнения обучаемым каждого предложенного ему тренажером упражнения. База пользователей хранит результаты всех пройденных обучаемым сеансов выполнения упражнений. Структура базы пользователей основана на модели онтологии пользователя системы.

Программные компоненты

Диагностический компьютерный тренажер включает следующие программные компоненты.

1. *Редактор информационных ресурсов*. В качестве редактора информационных ресурсов используется редактор ИРУО, разработанный ранее [4]. Редактор предназначен для формирования и модификации информационных ресурсов различных уровней общности, представленных семантическими сетями. Инженеры по знаниям формируют с помощью данного редактора онтологии на языке ИРУО. Затем сформированные онтологии подаются на вход этому же редактору, автоматически генерируется интерфейс редактора, управляемый онтологией, далее эксперты предметной области и преподаватели формируют базы знаний и данных напрямую, без посредников. Все информационные ресурсы — онтологии, базы знаний и данных представляются в едином унифицированном формате — семантическими сетями.

2. *Средство выполнения упражнений*. Оно служит для отображения упражнений, предлагаемых обучаемому, предоставления обучаемому возможности выбора упражнения, ввода решения поставленной перед обучаемым задачи и сохранения результата выполнения им упражнения в соответствующем информационном хранилище.

3. *Генератор значений наблюдений*. Формирование обучающих заданий преподавателем заключается в выборе заболевания из базы заболеваний и формирование одной из возможных диагностических ситуаций, а именно выбор некоторого подмножества начальных наблюдений и установление их значений, соответствующих клинической картине заболевания. Эти значения может формировать либо непосредственно преподаватель, либо они автоматически генерируются по описанию клинической картины заболевания. Далее, в процессе работы над упраж-

нением, обучаемый может запросить значения дополнительных наблюдений, которые необходимы для постановки диагноза. Значения запрошенных наблюдений автоматически генерируются в соответствии с клинической картиной заболевания. Если некоторое наблюдение не входит в клиническую картину заболевания, то генерируется значение из области нормальных значений.

4. *Решатель задачи диагностики*. Согласно модели онтологии упражнений исходные данные любого упражнения представляют собой набор значений наблюдений (заранее заданных преподавателем при формировании упражнения и сгенерированных тренажером после запроса значений дополнительных наблюдений обучаемым). Описываемый программный компонент решает обратную задачу медицинской диагностики, состоящую в определении диагноза по состоянию пациента. Для этого на основе информации из базы знаний о заболеваниях строят всевозможные схемы развития причинно-следственных связей, каждая из которых протекает согласно одному из возможных вариантов ее развития. При трансляции результатов диагностики обучаемый видит не только список подтвержденных и опровергнутых гипотез о диагнозе, но и объяснения принятых решений: причины опровержения и принятия гипотез. В качестве решателя задачи диагностики с подсистемой объяснения используется решатель задач экспертной системы медицинской диагностики, разработанный ранее [9].

Заключение

Разработка компьютерных диагностических медицинских тренажеров является актуальной задачей, потребность в них очень высока, однако на сегодняшний день рынок не предлагает специализированных инструментальных средств, упрощающих их разработку и сопровождение. Вместе с тем диагностический компьютерный тренажер — это сложный программный комплекс, основанный на знаниях.

В работе описана архитектура инструментального комплекса (оболочки) для создания медицинских диагностических компьютерных тренажеров, основанного на онтологии медицинской диагностики, приближенной к реальной. С помощью данного инструментального комплекса могут быть разработаны медицинские диагностические компьютерные тренажеры для различных разделов медицины. Архитектура этого комплекса включает ряд информационных и программных компонентов, разработанных ранее и использовавшихся при решении других задач. Так, решатель задач с подсистемой объяснений используется в экспертной системе медицинской диагностики. Редактор ИРУО, входящий в комплекс IACPaaS, на протяжении многих лет используется для формирования онтологии, баз знаний и данных. Информационные ресурсы — онтология медицинской диагностики, онтология наблюдений и онтология истории болезни используются в компьютерном банке медицинских знаний [10]. На их основе разработаны базы наблюдений для большинства разделов медицины (терапия, офтальмология, урология, гинекология, иммунология, эндокринология, хирургия, невропатология и др.). С использованием онтологии медицинской диагностики также формально описан ряд заболеваний (острый и хронический панкреатит, конъюнктивит, аппендицит, тупая травма почек, холецистит, прободение язвы желудка и 12-перстной кишки и др.). Онтология истории болезни использована для ввода историй болезни реальных пациентов. В настоящее время база историй болезни содержит более 100 реальных историй болезни пациентов. Разработанные информационные ресурсы (базы знаний и базы данных) также могут быть использованы при формировании диагностических компьютерных тренажеров в различных разделах медицины.

Работа выполнена при финансовой поддержке ДВО РАН в рамках Программы № 15 Президиума РАН, проект № 12-I-П15-03, и финансовой поддержке РФФИ, проект № 11-07-00460-а.

Список литературы

1. **Грибова В. В., Федорищев Л. А.** Обучающие виртуальные системы и средства для их создания // Вестник компьютерных и информационных технологий. 2012. № 3. С. 48—51.

2. **Сравнительный** анализ и тенденции развития компьютерных тренажеров и контролирующих систем // URL: http://nasgordeeva.narod.ru/ref.htm

3. **Грибова В. В., Осипенков Г., Сова С.** Концепция разработки диагностических компьютерных тренажеров на основе знаний // International Book Series "Information Science and Computing". 2009. № 12. С. 27—33.

4. **Орлов В. А., Клещев А. С.** Компьютерные банки знаний. Многоцелевой банк знаний // Информационные технологии. 2006. № 2. С. 2—8.

5. **Орлов В. А., Клещев А. С.** Компьютерные банки знаний. Модель процесса редактирования информационного наполнения // Информационные технологии. 2006. № 7. С. 11—16.

6. **Клещев А. С., Москаленко Ф. М., Черняховская М. Ю.** Онтология и модель онтологии предметной области "Медицинская диагностика". Владивосток: ИАПУ ДВО РАН, 2005. 44 с.

7. **Москаленко Ф. М.** Методы решения задачи медицинской диагностики на основе математической модели предметной области. Владивосток: ИАПУ ДВО РАН, 2010. С. 101—124.

8. **Грибова В. В., Клещев А. С., Крылов Д. А., Москаленко Ф. М., Смагин С. В., Тимченко В. А., Тютюнник М. Б., Шалфеева Е. А.** Проект IACPaaS — развиваемый комплекс для разработки, управления и использования интеллектуальных систем // Искусственный интеллект и принятие решений. 2011. № 1. С. 27—35.

9. **Москаленко Ф. М.** Экспертная система медицинской диагностики, основанная на реальной онтологии медицины, для многопроцессорной ЭВМ // Труды II международной конференции "Параллельные вычисления и задачи управления" PACO—2004. М.: Институт проблем управления им. В. А. Трапезникова РАН. С. 999—1084.

10. **Москаленко Ф. М.** Компьютерный банк знаний по медицинской диагностике (концепция, реализация, исследование производительности). Владивосток: ИАПУ ДВО РАН, 2011. 56 с.

CONTENTS

the training time of known systems. The experimental research shows that the proposed algorithm is characterized by satisfactory accuracy and high computing efficiency.

Keywords: automatic speech recognition, remote control systems, syllable phonetic, words phonetic decoding method, minimum information discrimination principle

Consider the problem of estimating the probabilities of strings in a document. To solve the problem, the model of *n*-grams is used. The *n*-gram classes is proposed to solve the estimation problem the large number of model parameters. Three discount models: Good-Turing, Katz and absolute discounting are used to solve the problem of zero probability of strings. The proposed model is illustrated by computational experiments on real data.

Keywords: statistical model, discount model, *n*-gram class, Good-Turing model, Katz model, absolute discounting

The article discuses the construction of virtual science labs and virtual research environments. An author's classification of different approaches and solutions. Describes the formulation of the problem, design and implementation of a test version of a virtual research environment for collaborative work geographically distributed researchers.

Keywords: modern information technology for research, virtual research environments, virtual lab

The article is about algorithms of realization of the elementary mathematical functions with analyzing performance and accuracy. Result criterion is the compromise between maximum speed, accuracy of computations and ease of technical implementation using PLD.

Keywords: programmable logic device, algorithm, performance

Article presents the examples of implementation of special visual effects of four-dimensional information spaces simulation in graphical user interfaces and the virtual desktop environment.

Keywords: graphical interface, software ergonomics, human-computer interaction, desktop

Describes the use of three-dimensional visualization of the information in the study of the special course "Cognitive graphics". The possibilities, the structure and function of some modern developments, which are used in teaching university students from one section of computer science — computer graphics and visualization.

Keywords: three-dimensional visualization, stereomonitor, virtual reality helmet, the system controlling the computer with eye movement, stereoscopic video display wall, cluster supercomputer

The article presents the concept and the architecture of the software complex joining a tool for development, maintenance and functioning medical diagnostic simulators and a medical diagnostic simulator based on knowledge. Classes of users and solving tasks are defined. Information and program components of the complex are described.

Keywords: computer simulator, ontology, knowledge base, intelligent system

Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала **(499) 269-5510**

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *Е. В. Конова*.

Корректор *Е. В. Комиссарова*.

Сдано в набор 01.02.2013. Подписано в печать 20.03.2013. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ IT413. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Адвансед солюшнз". Отпечатано в ООО "Адвансед солюшнз".

105120, г. Москва, ул. Нижняя Сыромятническая, д. 5/7, стр. 2, офис 2.